IN PARTNERSHIP WITH:
CNRS

Ecole normale supérieure de Lyon

Université Claude Bernard (Lyon 1)

Activity Report 2016

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
1.1.2. - Hardware accelerators (GPGPU, FPGA, etc.)
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
6.2.3. - Probabilistic methods
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
7.2. - Discrete mathematics, combinatorics
7.3. - Optimization
7.9. - Graph theory
7.11. - Performance evaluation

Other Research Topics and Application Domains:

3.2. - Climate and meteorology
3.3. - Geosciences
4. - Energy
4.1. - Fossile energy production (oil, gas)
4.5.1. - 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, temporary member (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]
Louis-Claude Canon [Univ. Franche-Comté, Associate Professor]

Laure Gonnord [Univ. Lyon I, Associate Professor, temporary member (see Section 3.4)]
Yves Robert [ENS Lyon, Professor, HDR]

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

PhD Students
Aurélien Cavelan [Inria]
Changjiang Gou [China Scholarship Council, from Oct 2016]
Li Han [China Scholarship Council, from Sep 2016]
Julien Herrmann [ENS Lyon, until Aug 2016]
Oguz Kaya [Inria]
Aurélie Kong Win Chang [ENS Lyon, from Feb 2016]
Maroua Maalej [Univ. Lyon]
Gilles Moreau [Inria]
Loïc Pottier [ENS Lyon]
Bertrand Simon [ENS Lyon]
Issam Rais [Inria]

Post-Doctoral Fellow
Hongyang Sun [Inria, until Jul 2016]

Visiting Scientists
Jiafan Li [ECNU, until Jan 2016]
Sicheng Dai [Inria, Visiting PhD student, from Oct 2016]
Samuel Mccauley [Inria, Visting PhD student, until Feb 2016]
Vitor Mendes Paisante [ENS Lyon, Visiting Master student, from Feb 2016 until May 2016]

Administrative Assistants
Emeline Boyer [Inria, from Oct 2016]
Laetitia Gauthe [Inria, until Oct 2016]

Others
Julien Braine [ENS Lyon, Master student, from Feb 2016 until Jun 2016]
Patrick Amestoy [INP Toulouse, external collaborator, HDR]
Alfredo Buttari [CNRS, external collaborator]
Franck Cappello [Argonne National Laboratory – USA, external collaborator, HDR]
Valentin Le Fèvre [ENS Lyon, Master student, from Feb 2016 until Jun 2016]
Raluca Portase [Intern from Cluj Napoca, Romania, from June 2016 to Sept 2016]

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 [83]. 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 [65], [71], [72], [82], 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 $2 \leq \alpha \leq 3$ [63], [64], [69]; 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 [84] 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 [80]. 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 [84]; (ii) *replication* consists in executing the same task on several processors simultaneously, in order to meet the reliability constraints [62]; and (iii) *checkpointing* consists in "saving" the work done at some certain instants, hence reducing the amount of work lost when a failure occurs [79].

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" [76]. 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 [73], [74], [78]. 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 [67] 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 [66] 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 [68], [81], [77]. 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 [70]. 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, code optimization and high-level synthesis for FPGA

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 2016 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 while keeping energy consumption at a reasonnable level 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). 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 [75] 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

Energy consumption bounds the performance of supercomputers since the end of Dennard scaling. Hence, reducing the electrical energy spent in a computation is the major challenge raised by Exaflop computing. Novel hardware, software, compilers and operating systems must be designed to increase the energy efficiency (in flops/watt) of data manipulation and computation itself. In the last decade, many specialized hardware accelerators (Xeon Phi, GPGPU) has emerged to overcome the limitations of mainstream processors, by trading the genericity for energy efficiency. However, the best supercomputers can only reach 8 Gflops/watt [61], which is far less than the 50 Gflops/watt required by an Exaflop supercomputer. An extreme solution would be to trade all the genericity by using specialized circuits. However such circuits (application specific integrated circuits, ASIC) are usually too expensive for the HPC market and lacks of flexibility. Once printed, an ASIC cannot be modified. Any algorithm update (or bug fix) would be impossible, which clearly not realistic.

Recently, reconfigurable circuits (Field Programmable Gate Arrays, FPGA) has appeared as a credible alternative for Exaflop computing. Major companies (including Intel, Google, Facebook and Microsoft) show a growing interest to FPGA and promising results has been obtained. For instance, in 2015, Microsoft reaches 40 Gflop/watts on a data-center deep learning algorithm mapped on Intel/Altera Arria 10 FPGAs. We believe that FPGA will become the new building block for HPC and Big Data systems. 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 in order 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/cross-fertilize the algorithms developed in both HPC and high-level synthesis communities. The purpose of the XtremLogic start-up company, co-founded by Christophe Alias and Alexandru Plesco is to transfer the results of this research to an industrial level compiler.

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

- Anne Benoit was the program chair of HiPC 2016 and the Algorithm-track vice-chair for SC'16.

5.1.1. Awards

- Yves Robert was awarded the *2016 Outsanding Service Award* of the *IEEE Technical Committee on Parallel Processing* (TCPP)

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 is developed by INPT(ENSEEIHT)-IRIT, Inria, CERFACS, University of Bordeaux, CNRS and ENS Lyon. In 2014, a consortium of industrial users has been created (http://mumps-consortium.org).

- 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, François-Henry Rouet, Philippe Combes and Gilles Moreau
- Partners: CERFACS - CNRS - ENS Lyon - INPT - IRIT - LIP - Université de Bordeaux - Université de Lyon - Université de Toulouse
- Latest public release: MUMPS 5.0.2 (July 2016)
- Contact: Jean-Yves L'Excellent
- URL: http://mumps-solver.org/
- Next MUMPS User Days: we have started organizing the next MUMPS User days, which will be hosted by Inria on June 1 and 2, 2017 near Grenoble, France (see http://mumps.enseeiht.fr/ud_2017.php)

In the context of the MUMPS consortium (see Section 8.1 and http://mumps-consortium.org), we had scientific exchanges and collaborations with industrial members and released two versions in advance for the consortium (in July 2016 and November 2016), containing major improvements for large-scale problems and many other improvements. Much effort was also put on developing features and algorithms to improve the quality and performance of MUMPS, especially in the context of problems offering potential for low-rank compression. This work is done in close collaboration with the partners who co-develop MUMPS, in particular in Toulouse.

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 (XtremLogic SAS)
- Contact: Christophe Alias

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 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

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 the algorithm “Counter-example based generation of ranking functions”. 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 the algorithm "An encoding of array verification problems into array-free Horn clauses". 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
- Demo page : http://laure.gonnord.org/pro/demopage/vaphor/index.php

7. New Results

7.1. A backward/forward recovery approach for the preconditioned conjugate gradient method

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

Several recent papers have introduced a periodic verification mechanism to detect silent errors in iterative solvers. Chen [PPoPP'13, pp. 167-176] has shown how to combine such a verification mechanism (a stability test checking the orthogonality of two vectors and recomputing the residual) with checkpointing: the idea is to verify every d iterations, and to checkpoint every $c \times d$ iterations. When a silent error is detected by the verification mechanism, one can rollback to and re-execute from the last checkpoint. In this work, we also propose to combine checkpointing and verification, but we use algorithm-based fault tolerance (ABFT) rather than stability tests. ABFT can be used for error detection, but also for error detection and correction, allowing a forward recovery (and no rollback nor re-execution) when a single error is detected. We introduce an abstract performance model to compute the performance of all schemes, and we instantiate it using the preconditioned conjugate gradient algorithm. Finally, we validate our new approach through a set of simulations.

This work has been accepted for publication in the *Journal of Computational Science* [13].

7.2. High performance parallel algorithms for the tucker decomposition of sparse tensors

Participants: Oguz Kaya, Bora Uçar.

We investigate an efficient parallelization of a class of algorithms for the well-known Tucker decomposition of general N-dimensional sparse tensors. The targeted algorithms are iterative and use the alternating least squares method. At each iteration, for each dimension of an N-dimensional input tensor, the following operations are performed: (i) the tensor is multiplied with $N-1$ matrices (TTMc step); (ii) the product is then converted to a matrix; and (iii) a few leading left singular vectors of the resulting matrix are computed (TRSVD step) to update one of the matrices for the next TTMc step. We propose an efficient parallelization of these algorithms for the current parallel platforms with multicore nodes. We discuss a set of preprocessing steps which takes all computational decisions out of the main iteration of the algorithm and provides an intuitive shared-memory parallelism for the TTM and TRSVD steps. We propose a coarse and a fine-grain parallel algorithm in a distributed memory environment, investigate data dependencies, and identify efficient communication schemes. We demonstrate how the computation of singular vectors in the TRSVD step can be carried out efficiently following the TTMc step. Finally, we develop a hybrid MPI-OpenMP implementation of the overall algorithm and report scalability results on up to 4096 cores on 256 nodes of an IBM BlueGene/Q supercomputer.

This work has been published at *ICPP'16* [28].

7.3. Preconditioning techniques based on the Birkhoff–von Neumann decomposition

Participants: Michele Benzi [Emory University, Atlanta, USA], Bora Uçar.

We introduce a class of preconditioners for general sparse matrices based on the Birkhoff–von Neumann decomposition of doubly stochastic matrices. These preconditioners are aimed primarily at solving challenging linear systems with highly unstructured and indefinite coefficient matrices. We present some theoretical results and numerical experiments on linear systems from a variety of applications.

This work has been accepted for publication in the journal *Computational Methods in Applied Mathematics* [10].

7.4. Parallel CP decomposition of sparse tensors using dimension trees

Participants: Oguz Kaya, Bora Uçar.

Tensor factorization has been increasingly used to address various problems in many fields such as signal processing, data compression, computer vision, and computational data analysis. CANDECOMP/PARAFAC (CP) decomposition of sparse tensors has successfully been applied to many well-known problems in web search, graph analytics, recommender systems, health care data analytics, and many other domains. In these applications, computing the CP decomposition of sparse tensors efficiently is essential in order to be able to process and analyze data of massive scale. For this purpose, we investigate an efficient computation and parallelization of the CP decomposition for sparse tensors. We provide a novel computational scheme for reducing the cost of a core operation in computing the CP decomposition with the traditional alternating least squares (CP-ALS) based algorithm. We then effectively parallelize this computational scheme in the context of CP-ALS in shared and distributed memory environments, and propose data and task distribution models for better scalability. We implement parallel CP-ALS algorithms and compare our implementations with an efficient tensor factorization library, using tensors formed from real-world and synthetic datasets. With our algorithmic contributions and implementations, we report up to 3.95x, 3.47x, and 3.9x speedups in sequential, shared memory parallel, and distributed memory parallel executions over the state of the art, and up to 1466x overall speedup over the sequential execution using 4096 cores on an IBM BlueGene/Q supercomputer.

This work is described in a technical report [49].

7.5. Scheduling series-parallel task graphs to minimize peak memory

Participants: Enver Kayaaslan, Thomas Lambert, Loris Marchal, Bora Uçar.

We consider a variant of the well-known, NP-complete problem of minimum cut linear arrangement for directed acyclic graphs. In this variant, we are given a directed acyclic graph and asked to find a topological ordering such that the maximum number of cut edges at any point in this ordering is minimum. In our main variant the vertices and edges have weights, and the aim is to minimize the maximum weight of cut edges in addition to the weight of the last vertex before the cut. There is a known, polynomial time algorithm [Liu, SIAM J. Algebra. Discr., 1987] for the cases where the input graph is a rooted tree. We focus on the variant where the input graph is a directed series-parallel graph, and propose a polynomial time algorithm. Directed acyclic graphs are used to model scientific applications where the vertices correspond to the tasks of a given application and the edges represent the dependencies between the tasks. In such models, the problem we address reads as minimizing the peak memory requirement in an execution of the application. Our work, combined with Liu's work on rooted trees addresses this practical problem in two important classes of applications.

This work is described in a technical report [50].

7.6. Matrix symmetrization and sparse direct solvers

Participants: Raluca Portase [Cluj Napoca, Romania], Bora Uçar.

We investigate algorithms for finding column permutations of sparse matrices in order to have large diagonal entries and to have many entries symmetrically positioned around the diagonal. The aim is to improve the memory and running time requirements of a certain class of sparse direct solvers. We propose efficient algorithms for this purpose by combining two existing approaches and demonstrate the effect of our findings in practice using a direct solver. In particular, we show improvements in a number of components of the running time of a sparse direct solver with respect to the state of the art on a diverse set of matrices.

This work is described in a technical report [53].

7.7. Robust Memory-Aware Mapping for Parallel Multifrontal Factorizations

Participants: Emmanuel Agullo [HIEPACS project-team], Patrick Amestoy [INPT-IRIT], Alfredo Buttari [CNRS-IRIT], Abdou Guermouche [HIEPACS project-team], Jean-Yves L'Excellent, François-Henry Rouet [Lawrence Berkeley Laboratory, CA, USA].

In this work, we study the memory scalability of the parallel multifrontal factorization of sparse matrices. In particular, we are interested in controlling the active memory specific to the multifrontal factorization. We illustrate why commonly used mapping strategies (e.g., the proportional mapping) cannot provide a high memory efficiency, which means that they tend to let the memory usage of the factorization grow when the number of processes increases. We propose "memory-aware" algorithms that aim at maximizing the granularity of parallelism while respecting memory constraints. These algorithms provide accurate memory estimates prior to the factorization and can significantly enhance the robustness of a multifrontal code. We illustrate our approach with experiments performed on large matrices.

This work has been published in the *SIAM Journal on Scientific Computing* [1].

7.8. Fast 3D frequency-domain full waveform inversion with a parallel Block Low-Rank multifrontal direct solver: application to OBC data from the North Sea

Participants: Patrick Amestoy [INPT-IRIT], Romain Brossier [ISTerre], Alfredo Buttari [CNRS-IRIT], Jean-Yves L'Excellent, Théo Mary [UPS-IRIT], Ludovic Métivier [CNRS-ISTerre-LJK], Alain Miniussi [Geoazur], Stéphane Operto [Geoazur].

Wide-azimuth long-offset OBC/OBN surveys provide a suitable framework to perform computationally-efficient frequency-domain full waveform inversion (FWI) with a few discrete frequencies. Frequency-domain seismic modeling is performed efficiently with moderate computational resources for a large number of sources with a sparse multifrontal direct solver (Gauss-elimination techniques for sparse matrices). Approximate solutions of the time-harmonic wave equation are computed using a Block Low-Rank (BLR) approximation, leading to a significant reduction in the operation count and in the volume of communication during the LU factorization as well as offering a great potential for reduction in the memory demand. Moreover, the sparsity of the seismic source vectors is exploited to speed up the forward elimination step during the computation of the monochromatic wavefields. The relevance and the computational efficiency of the frequency-domain FWI performed in the visco-acoustic VTI approximation is shown with a real 3D OBC case study from the North Sea. The FWI subsurface models show a dramatic resolution improvement relative to the initial model built by reflection traveltime tomography. The amplitude errors introduced in the modeled wavefields by the BLR approximation for different low-rank thresholds have a negligible footprint in the FWI results. With respect to a standard multifrontal sparse direct factorization, and without compromise on the accuracy of the imaging, the BLR approximation can bring a reduction of the LU factor size by a factor up to three. This reduction is not yet exploited to reduce the effective memory usage (ongoing work). The flop reduction can be larger than a factor of 10 and can bring a factor of time reduction of around three. Moreover, this reduction factor tends to increase with frequency, namely with the matrix size. Frequency-domain visco-acoustic VTI FWI can be viewed as an efficient tool to build an initial model for elastic FWI of 4-C OBC data.

This work has been published in the journal *Geophysics* [2].

7.9. Matching-Based Allocation Strategies for Improving Data Locality of Map Tasks in MapReduce

Participant: Loris Marchal.

MapReduce is a well-know framework for distributing data-processing computations on parallel clusters. In MapReduce, a large computation is broken into small tasks that run in parallel on multiple machines, and scales easily to very large clusters of inexpensive commodity computers. Before the Map phase, the original dataset is first split into chunks, that are replicated (a constant number of times, usually 3) and distributed onto the computing nodes. During the Map phase, nodes request tasks and are allocated first tasks associated to local chunks (if any). Communications take place when requesting nodes do not hold any local chunk anymore. In this work, we provide the first complete theoretical data locality analysis of the Map phase of MapReduce, and more generally, for bag-of-tasks applications that behaves like MapReduce. We show that if tasks are homogeneous (in term of processing time), once the chunks have been replicated randomly on resources with a replication factor larger than 2, it is possible to find a priority mechanism for tasks that achieves a quasi-perfect number of communications using a sophisticated matching algorithm. In the more realistic case of heterogeneous processing times, we prove using an actual trace of a MapReduce server that this priority mechanism enables to complete the Map phase with significantly fewer communications, even on realistic distributions of task durations.

This work is described in a technical report [41].

7.10. Minimizing Rental Cost for Multiple Recipe Applications in the Cloud

Participant: Loris Marchal.

Clouds are more and more becoming a credible alternative to parallel dedicated resources. The pay-per-use pricing policy however highlights the real cost of computing applications. This new criterion, the cost, must then be assessed when scheduling an application in addition to more traditional ones as the completion time or the execution flow. In this work, we tackle the problem of optimizing the cost of renting computing instances to execute an application on the cloud while maintaining a desired performance (throughput). The target application is a stream application based on a DAG pattern, i.e., composed of several tasks with dependencies, and instances of the same execution task graph are continuously executed on the instances. We provide some theoretical results on the problem of optimizing the renting cost for a given throughput then propose some heuristics to solve the more complex parts of the problem, and we compare them to optimal solutions found by linear programming.

This work has been published in *IPDPS Workshops* [27].

7.11. Malleable task-graph scheduling with a practical speed-up model

Participants: Loris Marchal, Bertrand Simon, Oliver Sinnen [Univ. Auckland, New Zealand], Frédéric Vivien.

Scientific workloads are often described by Directed Acyclic task Graphs. Indeed, DAGs represent both a model frequently studied in theoretical literature and the structure employed by dynamic runtime schedulers to handle HPC applications. A natural problem is then to compute a makespan-minimizing schedule of a given graph. In this work, we are motivated by task graphs arising from multifrontal factorizations of sparsematrices and therefore work under the following practical model. We focus on malleable tasks (i.e., a single task can be allotted a time-varying number of processors) and specifically on a simple yet realistic speedup model: each task can be perfectly parallelized, but only up to a limited number of processors. We first prove that the associated decision problem of minimizing the makespan is NP-Complete. Then, we study a widely used algorithm, PropScheduling, under this practical model and propose a new strategy GreedyFilling. Even though both strategies are 2-approximations, experiments on real and synthetic data sets show that GreedyFilling achieves significantly lower makespans.

This work is described in a technical report [52].

7.12. Dynamic memory-aware task-tree scheduling

Participant: Loris Marchal.

Factorizing sparse matrices using direct multifrontal methods generates directed tree-shaped task graphs, where edges represent data dependency between tasks. This work revisits the execution of tree-shaped task graphs using multiple processors that share a bounded memory. A task can only be executed if all its input and output data can fit into the memory. The key difficulty is to manage the order of the task executions so that we can achieve high parallelism while staying below the memory bound. In particular, because input data of unprocessed tasks must be kept in memory, a bad scheduling strategy might compromise the termination of the algorithm. In the single processor case, solutions that are guaranteed to be below a memory bound are known. The multi-processor case (when one tries to minimize the total completion time) has been shown to be NP-complete. We designed in this work a novel heuristic solution that has a low complexity and is guaranteed to complete the tree within a given memory bound. We compared our algorithm to state of the art strategies, and observed that on both actual execution trees and synthetic trees, we always performed better than these solutions, with average speedups between 1.25 and 1.45 on actual assembly trees. Moreover, we showed that the overhead of our algorithm is negligible even on deep trees (10^5), and would allow its runtime execution.

This work is described in a technical report [39].

7.13. Optimal resilience patterns to cope with fail-stop and silent errors

Participants: Anne Benoit, Aurélien Cavelan, Yves Robert, Hongyang Sun.

This work focuses on resilience techniques at extreme scale. Many papers deal with fail-stop errors. Many others deal with silent errors (or silent data corruptions). But very few papers deal with fail-stop and silent errors simultaneously. However, HPC applications will obviously have to cope with both error sources. This work presents a unified framework and optimal algorithmic solutions to this double challenge. Silent errors are handled via verification mechanisms (either partially or fully accurate) and in-memory checkpoints. Fail-stop errors are processed via disk checkpoints. All verification and checkpoint types are combined into computational patterns. We provide a unified model, and a full characterization of the optimal pattern. Our results nicely extend several published solutions and demonstrate how to make use of different techniques to solve the double threat of fail-stop and silent errors. Extensive simulations based on real data confirm the accuracy of the model, and show that patterns that combine all resilience mechanisms are required to provide acceptable overheads.

This work was presented at the *IPDPS'2016* conference [20].

7.14. Two-level checkpointing and partial verifications for linear task graphs

Participants: Anne Benoit, Aurélien Cavelan, Yves Robert, Hongyang Sun.

Fail-stop and silent errors are unavoidable on large-scale platforms. Efficient resilience techniques must accommodate both error sources. A traditional checkpointing and rollback recovery approach can be used, with added verifications to detect silent errors. A fail-stop error leads to the loss of the whole memory content, hence the obligation to checkpoint on a stable storage (e.g., an external disk). On the contrary, it is possible to use in-memory checkpoints for silent errors, which provide a much smaller checkpoint and recovery overhead. Furthermore, recent detectors offer partial verification mechanisms, which are less costly than guaranteed verifications but do not detect all silent errors. In this work, we show how to combine all these techniques for HPC applications whose dependence graph is a chain of tasks, and provide a sophisticated dynamic programming algorithm returning the optimal solution in polynomial time. Simulations demonstrate that the combined use of multi-level checkpointing and partial verifications further improves performance.

This work was presented at the *17th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2016)* [21].

7.15. Resilient application co-scheduling with processor redistribution

Participants: Anne Benoit, Loïc Pottier, Yves Robert.

Recently, the benefits of co-scheduling several applications have been demonstrated in a fault-free context, both in terms of performance and energy savings. However, large-scale computer systems are confronted to frequent failures, and resilience techniques must be employed to ensure the completion of large applications. Indeed, failures may create severe imbalance between applications, and significantly degrade performance. In this work, we propose to redistribute the resources assigned to each application upon the striking of failures, in order to minimize the expected completion time of a set of co-scheduled applications. First we introduce a formal model and establish complexity results. When no redistribution is allowed, we can minimize the expected completion time in polynomial time, while the problem becomes NP-complete with redistributions, even in a fault-free context. Therefore, we design polynomial-time heuristics that perform redistributions and account for processor failures. A fault simulator is used to perform extensive simulations that demonstrate the usefulness of redistribution and the performance of the proposed heuristics.

This work was presented at the *ICCP'16* conference [22].

7.16. A different re-execution speed can help

Participants: Anne Benoit, Aurélien Cavelan, Valentin Le Fèvre, Yves Robert, Hongyang Sun.

We consider divisible load scientific applications executing on large-scale platforms subject to silent errors. While the goal is usually to complete the execution as fast as possible in expectation, another major concern is energy consumption. The use of dynamic voltage and frequency scaling (DVFS) can help save energy, but at the price of performance degradation. Consider the execution model where a set of K different speeds is given, and whenever a failure occurs, a different re-execution speed may be used. Can this help? We address the following bi-criteria problem: how to compute the optimal checkpointing period to minimize energy consumption while bounding the degradation in performance. We solve this bi-criteria problem by providing a closed-form solution for the checkpointing period, and demonstrate via a comprehensive set of simulations that a different re-execution speed can indeed help.

This work was presented at the *5th International Workshop on Power-aware Algorithms, Systems, and Architectures* [19].

7.17. Coping with recall and precision of soft error detectors

Participants: Anne Benoit, Aurélien Cavelan, Yves Robert, Hongyang Sun.

Many methods are available to detect silent errors in high-performance computing (HPC) applications. Each method comes with a cost, a recall (fraction of all errors that are actually detected, i.e., false negatives), and a precision (fraction of true errors amongst all detected errors, i.e., false positives). The main contribution of this work is to characterize the optimal computing 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 first prove that detectors with imperfect precisions offer limited usefulness. Then we focus on detectors with perfect precision, and 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. Extensive simulations illustrate the usefulness of detectors with false negatives, which are available at a lower cost than the guaranteed detectors.

This work was accepted for publication in the *Journal of Parallel and Distributed Computing* [7].

7.18. Checkpointing strategies for scheduling computational workflows

Participants: Anne Benoit, 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 deciding for each task whether to checkpoint it or not after it completes. We give a polynomial-time optimal algorithm for fork DAGs (Directed Acyclic Graphs) and show that the problem is NP-complete with join DAGs. We also investigate the complexity of the simple case in which no task is checkpointed. Our main result is a polynomial-time algorithm to compute the expected execution time of a workflow, with a given task execution order and specified to-be-checkpointed tasks. Using this algorithm as a basis, we propose several heuristics for solving the scheduling problem. We evaluate these heuristics for representative workflow configurations.

This work was published in the *International Journal of Networking and Computing* [4].

7.19. Assessing General-Purpose Algorithms to Cope with Fail-Stop and Silent Errors

Participants: Anne Benoit, Aurélien Cavelan, Yves Robert, Hongyang Sun.

We combine the traditional checkpointing and rollback recovery strategies with verification mechanisms to cope with both fail-stop and silent errors. The objective is to minimize makespan and/or energy consumption. For divisible load applications, we use first-order approximations to find the optimal checkpointing period to minimize execution time, with an additional verification mechanism to detect silent errors before each checkpoint, hence extending the classical formula by Young and Daly for fail-stop errors only. We further extend the approach to include intermediate verifications, and to consider a bi-criteria problem involving both time and energy (linear combination of execution time and energy consumption). Then, we focus on application workflows whose dependence graph is a linear chain of tasks. Here, we determine the optimal checkpointing and verification locations, with or without intermediate verifications, for the bi-criteria problem. Rather than using a single speed during the whole execution, we further introduce a new execution scenario, which allows for changing the execution speed via dynamic voltage and frequency scaling (DVFS). In this latter scenario, we determine the optimal checkpointing and verification locations, as well as the optimal speed pairs for each task segment between any two consecutive checkpoints. Finally, we conduct an extensive set of simulations to support the theoretical study, and to assess the performance of each algorithm, showing that the best overall performance is achieved under the most flexible scenario using intermediate verifications and different speeds.

This work was accepted for publication in the journal *ACM Transactions on Parallel Computing* [8].

7.20. A failure detector for HPC platforms

Participant: Yves Robert.

Building an infrastructure for Exascale applications requires, in addition to many other key components, a stable and efficient failure detector. This work describes the design and evaluation of a robust failure detector, able to maintain and distribute the correct list of alive resources within proven and scalable bounds. The detection and distribution of the fault information follow different overlay topologies that together guarantee minimal disturbance to the applications. A virtual observation ring minimizes the overhead by allowing each node to be observed by another single node, providing an unobtrusive behavior. The propagation stage is using a non-uniform variant of a reliable broadcast over a circulant graph overlay network, and guarantees a logarithmic fault propagation. Extensive simulations, together with experiments on the Titan ORNL supercomputer, show that the algorithm performs extremely well, and exhibits all the desired properties of an Exascale-ready algorithm.

This work was presented at the *SC'16* conference [24].

7.21. Optimal multistage algorithm for adjoint computatio

Participant: Yves Robert.

We reexamine the work of Stumm and Walther on multistage algorithms for adjoint computation. We provide an optimal algorithm for this problem when there are two levels of checkpoints, in memory and on disk. Previously, optimal algorithms for adjoint computations were known only for a single level of checkpoints with no writing and reading costs; a well-known example is the binomial checkpointing algorithm of Griewank and Walther. Stumm and Walther extended that binomial checkpointing algorithm to the case of two levels of checkpoints, but they did not provide any optimality results. We bridge the gap by designing the first optimal algorithm in this context. We experimentally compare our optimal algorithm with that of Stumm and Walther to assess the difference in performance.

This work was accepted for publication in the *SIAM Journal on Scientific Computing* [5].

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

Participant: Yves Robert.

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 [14].

7.23. When Amdahl Meets Young/Daly

Participants: Aurélien Cavelan, Yves Robert.

This work investigates the optimal number of processors to execute a parallel job, whose speedup profile obeys Amdahl's law, on a large-scale platform subject to fail-stop and silent errors. We combine the traditional checkpointing and rollback recovery strategies with verification mechanisms to cope with both error sources. We provide an exact formula to express the execution overhead incurred by a periodic checkpointing pattern of length T and with P processors, and we give first-order approximations for the optimal values T* and P* as a function of the individual processor MTBF. A striking result is that P* is of the order of the fourth root of the individual MTBF if the checkpointing cost grows linearly with the number of processors, and of the order of its third root if the checkpointing cost stays bounded for any P. We conduct an extensive set of simulations to support the theoretical study. The results confirm the accuracy of first-order approximation under a wide range of parameter settings.

This work was presented at the *Cluster'16* conference [26].

7.24. Computing the expected makespan of task graphs in the presence of silent errors

Participants: Julien Herrmann, Yves Robert.

Applications structured as Directed Acyclic Graphs (DAGs) of tasks correspond to a general model of parallel computation that occurs in many domains, including popular scientific workflows. DAG scheduling has received an enormous amount of attention, and several list-scheduling heuristics have been proposed and shown to be effective in practice. Many of these heuristics make scheduling decisions based on path lengths in the DAG. At large scale, however, compute platforms and thus tasks are subject to various types of failures with no longer negligible probabilities of occurrence. Failures that have recently received increasing attention are silent errors, which cause a task to produce incorrect results even though it ran to completion. Tolerating silent errors is done by checking the validity of the results and re-executing the task from scratch in case of an invalid result. The execution time of a task then becomes a random variable, and so are path lengths. Unfortunately, computing the expected makespan of a DAG (and equivalently computing expected path lengths in a DAG) is a computationally difficult problem. Consequently, designing effective scheduling heuristics is preconditioned on computing accurate approximations of the expected makespan. In this work we propose an algorithm that computes a first order approximation of the expected makespan of a DAG when tasks are subject to silent errors. We compare our proposed approximation to previously proposed such approximations for three classes of application graphs from the field of numerical linear algebra. Our evaluations quantify approximation error with respect to a ground truth computed via a brute-force Monte Carlo method. We find that our proposed approximation outperforms previously proposed approaches, leading to large reductions in approximation error for low (and realistic) failure rates, while executing much faster.

This work was presented at the *Ninth Int. Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2)* [25].

7.25. Toward an Optimal Online Checkpoint Solution under a Two-Level HPC Checkpoint Model

Participants: Yves Robert, Frédéric Vivien.

The traditional single-level checkpointing method suffers from significant overhead on large-scale platforms. Hence, multilevel checkpointing protocols have been studied extensively in recent years. The multilevel checkpoint approach allows different levels of checkpoints to be set (each with different checkpoint overheads and recovery abilities), in order to further improve the fault tolerance performance of extreme-scale HPC applications. How to optimize the checkpoint intervals for each level, however, is an extremely difficult problem. In this work, we construct an easy-to-use two-level checkpoint model. Checkpoint level 1 deals with errors with low checkpoint/recovery overheads such as transient memory errors, while checkpoint level 2 deals with hardware crashes such as node failures. Compared with previous optimization work, our new optimal checkpoint solution offers two improvements: (1) it is an online solution without requiring knowledge of the job length in advance, and (2) it shows that periodic patterns are optimal and determines the best pattern. We evaluate the proposed solution and compare it with the most up-to-date related approaches on an extreme-scale simulation testbed constructed based on a real HPC application execution. Simulation results show that our proposed solution outperforms other optimized solutions and can improve the performance significantly in some cases. Specifically, with the new solution the wall-clock time can be reduced by up to 25.3% over that of other state-of-the-art approaches. Finally, a brute-force comparison with all possible patterns shows that our solution is always within 1% of the best pattern in the experiments.

This work has been published in IEEE Transactions on Parallel and Distributed Systems [11].

7.26. Cell morphing: from array programs to array-free Horn clauses

Participants: Laure Gonnord, David Monniaux [(CNRS/Verimag)], Julien Braine [(M2 Student)].

Automatically verifying safety properties of programs is hard. Many approaches exist for verifying programs operating on 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. Our work addresses that issue with a powerful and flexible abstraction that morphes concrete array cells into a finite set of abstract ones. This abstraction is parametric both in precision and in the back-end analysis used. From

our programs with arrays, we generate nonlinear Horn clauses over scalar variables only, in a common format with clear and unambiguous logical semantics, for which there exist several solvers. We thus avoid the use of solvers operating over arrays, which are still very immature. Experiments with our prototype VAPHOR show that this approach can prove automatically and without user annotations the functional correctness of several classical examples, including *selection sort*, *bubble sort*, *insertion sort*, as well as examples from literature on array analysis.

This work has been published in Static Analysis Symposium [30] for the array part. We are currently deseigning an extension to programs with inductive data structures.

7.27. Symbolic Analyses of pointers

Participants: Laure Gonnord, Maroua Maalej, Fernando Pereira [(UFMG, Brasil)], Leonardo Barbosa [(UFMG, Brasil)], Vitor Paisante [(UFMG, Brasil)], Pedro Ramos [(UFMG, 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.

A first work presents a new range-based 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.

A second work starts from an obvious, yet unexplored, observation: if a pointer is strictly less than another, they cannot alias. Motivated by this remark, we use the abstract interpretation framework to build strict less-than relations between pointers. To this end, we construct a program representation that bestows the Static Single Information (SSI) property onto our dataflow analysis. SSI gives us an efficient sparse algorithm, which, once seen as a form of abstract interpretation, is correct by construction. We have implemented our static analysis in LLVM. It runs in time linear on the number of program variables, and, depending on the benchmark, it can be as much as six times more precise than the pointer disambiguation techniques already in place in that compiler.

This work has been published in the *International Symposium of Code Generation and Optmization* [31] and at CGO'17 [29].

7.28. High-Level Synthesis of Pipelined FSM from Loop Nests

Participants: Christophe Alias, Fabrice Rastello [(Inria/CORSE)], Alexandru Plesco [(XtremLogic SAS, France)].

Embedded systems raise many challenges in power, space and speed efficiency. The current trend is to build heterogeneous systems on a chip with specialized processors and hardware accelerators. Generating an hardware accelerator from a computational kernel requires a deep reorganization of the code and the data. Typically, parallelism and memory bandwidth are met thanks to fine-grain loop transformations. Unfortunately, the resulting control automaton is often very complex and eventually bound the circuit frequency, which limits the benefits of the optimization. This is a major lock, which strongly limits the power of the code optimizations appliable by high-level synthesis tools.

In this work, we propose an architecture of control automaton and an algorithm of high-level synthesis which translates efficiently the control required by fine-grain loop optimizations. Unlike the previous approaches, our control automaton can be pipelined *at will, without any restriction*. Hence, the frequency of the automaton can be as high as possible. Experimental results on FPGA confirms that our control circuit can reach a high frequency with a reasonable resource consumption.

This work is described in a technical report [36].

7.29. Estimation of Parallel Complexity with Rewriting Techniques

Participants: Christophe Alias, Laure Gonnord, Carsten Fuhs [(Birbeck, UK)].

We show how monotone interpretations - a termination analysis technique for term rewriting systems - can be used to assess the inherent parallelism of recursive programs manipulating inductive data structures. As a side effect, we show how monotone interpretations specify a parallel execution order, and how our approach extends naturally affine scheduling - a powerful analysis used in parallelising compilers - to recursive programs. This preliminary work opens new perspectives in automatic parallelisation.

This work has been published in the *Workshop on Termination,* [15].

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

8.1.1. Mumps Consortium (2014-2019)

In 2016, in the context of the MUMPS consortium (http://mumps-consortium.org):

- We have signed two new membership agreements, with Free Field Technologies and Safran in 2016, on top of the on-going agreements signed in 2014 and 2015 with Altair, EDF, ESI-Group, LSTC, Michelin, Siemens SISW (Belgium) and TOTAL.
- 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 second consortium committee meeting, at Michelin (Clermont-Ferrand).
- Two engineers have been funded by the membership fees, for software engineering and software development, performance study and comparisons, business development and management of the consortium.
- 0.5 year of a PhD student were funded by the membership fees (see Section 9.1).

8.2. Technological Transfer: XtremLogic Start-Up

The XTREMLOGIC start-up (former Zettice project) was initiated 5 years ago by Alexandru Plesco and Christophe Alias.

The goal of XTREMLOGIC is to provide energy-efficient circuit blocks for FPGA reconfigurable circuits. These circuits are produced automatically through an high-level synthesis (HLS) tool based on state-of-the-art automatic parallelization technologies, notably from the polyhedral community. The compiler technology transfered to XTREMLOGIC is the result of a tight collaboration between Christophe Alias and Alexandru Plesco. In a way, XTREMLOGIC can be viewed as "applied research" targetting a direct industrial application.

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.

9.2. National Initiatives

9.2.1. ANR

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.

9.3. 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: Kentro Erevnas Technologias Kai Anaptyxix Thessalias

Partners:

Ethniko Kentro Erevnas Kai Technologikis Anaptyxis (Greece)

Ecole Polytechnique Federale de Lausanne (Switzerland)

The Queen's University of Belfast (United Kingdom)

Rheinisch-Westfaelische Technische Hochschule Aachen (Germany)

Interuniversitair Micro-Electronica Centrum Vzw (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

9.4.1.1. JLESC — Joint Laboratory on Extreme Scale Computing

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

9.4.2.1. Keystone

Title: Scheduling algorithms for sparse linear algebra at extreme scale

International Partner (Institution - Laboratory - Researcher):

Vanderbilt University (United States) - Padma Raghavan

Start year: 2016

See also: http://graal.ens-lyon.fr/~abenoit/Keystone

The Keystone project aims at investigating sparse matrix and graph problems on NUMA multicores and/or CPU-GPU hybrid models. The goal is to improve the performance of the algorithms, while accounting for failures and trying to minimize the energy consumption. The long-term objective is to design robust sparse-linear kernels for computing at extreme scale. In order to optimize the performance of these kernels, we plan to take particular care of locality and data reuse. Finally, there are several real-life applications relying on these kernels, and the Keystone project will assess the performance and robustness of the scheduling algorithms in applicative contexts. We believe that the complementary expertise of the two teams in the area of scheduling HPC applications at scale (ROMA — models and complexity; and SSCL — architecture and applications) is the key to the success of this associate team. We have already successfully collaborated in the past and expect the collaboration to reach another level thanks to Keystone.

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.
- 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.

9.4.4. 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.

In the context of this collaboration two students from ECNU, Li Han and Changjiang Gou, have joined Roma for their PhD.

9.5. International Research Visitors

9.5.1. Visits of International Scientists

- 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

- Laure Gonnord supervised two Master Students in Spring 2016, Vitor Paisante (static analyses for pointers) and Julien Braine (static analyses for data structures).
- Bora Uçar supervised an Raluca Portase, an Erasmus student, for three months (June–September 2016).

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.

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).

Bora Uçar was the workshops chair at IPDPS 2016, local chair of the a topic of Euro-Par 2016, co-chair of PCO 2016 (a workshop of IPDPS 2016), and has organized mini-symposia at SIAM PP16 and PMAA16.

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.

Bora Uçar serves in the steering committee of CSC.

10.1.2.2. Chair of Conference Program Committees

Anne Benoit was the program chair of HiPC 2016, and the program area chair for Algorithms of SC 2016.

Loris Marchal was the program chair of HeteroPar 2016.

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 IPDPS, SC, HCW, and Ena-HPC.

Laure Gonnord was a member of the program committee of VMCAI'17.

Jean-Yves L'Excellent was a member of the program committee of VECPAR'16.

Loris Marchal was a member of the program committee of IPDPS 2016.

Yves Robert was a member of the program committee of FTS, FTXS, ICCS, ISCIS, and SC

Bora Uçar was a member of the program committee of the following conferences and workshops: HiPC 2016, IPDPS 2016, ICCS 2016, HPC4BD 2016, P^{3}MA, CSE 2016, MPP2016.

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

10.1.2.4. Reviewer

Jean-Yves L'Excellent reviewed papers for VECPAR'16, ADVCOMP'16.

Christophe Alias reviewed papers for IMPACT'16.

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), of JPDC (Elsevier Journal of Parallel and Distributed Computing) and SUSCOM (Elsevier Journal of Sustainable Computing).

Loris Marchal is an invited associated Editor of Parallel Computing (Elsevier) for a special issue following HeteroPar 2016.

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

Bora Uçar is Associate Editor of Parallel Computing (Elsevier).

Frédéric Vivien is Associate Editor of Parallel Computing (Elsevier) and of JPDC (Elsevier Journal of Parallel and Distributed Computing).

10.1.3.2. Reviewer - Reviewing Activities

Christophe Alias reviewed papers for TVLSI (IEEE Transactions on Very Large Scale Integration Systems), PARCO (Parallel Computing), MICPRO (Microprocessors and Microsystems).

10.1.4. Invited Talks

Christophe Alias gave a talk at "Journée Calcul" in ENS-Lyon on May 2016 and a talk at "Journée Langage, Compilation et Sémantique" in ENS-Lyon on November 2016.

10.1.5. Tutorials

Yves Robert gave a tutorial on *Fault-tolerance techniques for HPC platforms* at PPoPP'16 (with Thomas Hérault), SC'16 (with Aurélien Bouteiller, George Bosilca, and Thomas Hérault) and Euro-Par'16.

10.1.6. Leadership within the Scientific Community

Christophe Alias, together with Cédric Bastoul (CAMUS), co-founded the Impact workshop (International Workshop on Polyhedral Compilation Techniques) on 2011, which is now the reference international event of the polyhedral compilation community http://impact.gforge.inria.fr/. Since then, Christophe Alias is involved in IMPACT committees.

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).

10.1.7. Research Administration

Anne Benoit is a member of the executive committee of the Labex MI-LYON. She is the head of the Master of fundamental computer science at ENS Lyon.

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 is a member of the direction board of the LIP laboratory.

Yves Robert was a member of the Senior Member election of Institut Universitaire de France. He was a committee member of the IEEE Fellows selection. he is a member of the scientific council of the Maison de la Simulation.

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.
- Maroua Maalej : Algorithmique et Programmation Avancée (TD=18, TP=24h), L2, Université Lyon 1 Claude Bernard : Autumn 2016.
- Maroua Maalej : Architecture et système (TP=24h), L2, Université Lyon 1 Claude Bernard : Autumn 2016.
- Maroua Maalej : Gestion de Projet et Génie Logiciel (TD/TP=10h), M1, Université Lyon 1 Claude Bernard : Autumn 2016.
- Christophe Alias : Compilation et outils de développement (CM+TD=18h), L3, INSA Centre-Val-de-Loire : Spring 2016.

- Christophe Alias : Concours E3A – épreuve informatique MPSI (correcteur) : Spring 2016.
- 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 : Compilation (CM+TD 76h), M1, Université Claude Bernard Lyon 1, et M1 Ecole Normale Supérieure de Lyon.
- Laure Gonnord : Préparation à l'écrit et à l'oral d'informatique du capès d'informatique, 10h, M1 MEEF Université Claude Bernard Lyon1.
- Laure Gonnord : Program Analysis : (CM+TP 10h, avec D.Monniaux), M2 Ecole Normale Supérieure de Lyon.
- Christophe Alias : Optimisation d'applications embarquées (CM+TD=24h), M1, INSA Centre-Val-de-Loire.
- Christophe Alias: Advanced Compilers: Automatic Parallelization and High-level Synthesis (CM 24h, avec F. Rastello), M2, Ecole Normale Supérieure de Lyon, France.
- 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: Changjiang Gou, "Resilient and energy-aware scheduling algorithms for large-scale distributed systems", started in September 2016, funding: China Scholarship Council, advisors: Anne Benoit and Loris Marchal.

PhD in progress: Li Han, "Algorithms for detecting and correcting silent and non-functional errors in scientific workflows", started in September 2016, funding: China Scholarship Council, advisors: Yves Robert and Frédéric Vivien

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: Aurélie Kong Win Chang, "Techniques de résilience pour l'ordonnancement de workflows sur plates-formes décentralisées (cloud computing) avec contraintes de sécurité", started in October 2016, funding: ENS Lyon, advisors: Yves Robert, Yves Caniou and Eddy Caron.

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 July 1st: 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).

10.2.3. Juries

- Christophe Alias participated to the PhD jury of Guillaume Iooss (Colorado State University), in July 2016.
- Laure Gonnord participated to the Inria recruting jury for junior research positions (CR2), in Rennes, in Spring 2016.
- Laure Gonnord participated to the PhD jury of Nasrine Damouche (Univ. Perpignan) and Sajith Kalathingal (Univ. Rennes), in December 2016.
- Laure Gonnord is member of the "Comité de Suivi de thèse" de Maurica Fonenantsoa (Univ. Réunion) since 2015.
- Loris Marchal participated to the selection committee recruting an assistant professor (MCF) at University of Bordeaux 1, in Spring 2016.

10.3. Popularization

Frédéric Vivien gave two lectures about "Approximation algorithms" 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

Articles in International Peer-Reviewed Journals

[1] E. AGULLO, P. R. AMESTOY, A. BUTTARI, A. GUERMOUCHE, J.-Y. L'EXCELLENT, F.-H. ROUET. *Robust memory-aware mappings for parallel multifrontal factorizations*, in "SIAM Journal on Scientific Computing", July 2016, vol. 38, n° 3, 23 p. , https://hal.inria.fr/hal-01334113

[2] P. R. AMESTOY, R. BROSSIER, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY, L. MÉTIVIER, A. MINIUSSI, S. OPERTO. *Fast 3D frequency-domain full waveform inversion with a parallel Block Low-Rank multifrontal direct solver: application to OBC data from the North Sea*, in "Geophysics", 2016, vol. 81, n° 6, pp. R363-R383, https://hal.inria.fr/hal-01349119

[3] G. AUPY, A. BENOIT. *Approximation Algorithms for Energy, Reliability, and Makespan Optimization Problems*, in "Parallel Processing Letters", 2016, vol. 26, n° 01, 23 p. , https://hal.inria.fr/hal-01252333

[4] G. AUPY, A. BENOIT, H. CASANOVA, Y. ROBERT. *Checkpointing Strategies for Scheduling Computational Workflows*, in "International Journal of Networking and Computing", 2016, vol. 6, n° 1, pp. 2-26 [*DOI :* 10.15803/IJNC.6.1_2], https://hal.inria.fr/hal-01354874

[5] G. AUPY, J. HERRMANN, P. HOVLAND, Y. ROBERT. *Optimal Multistage Algorithm for Adjoint Computation*, in "SIAM Journal on Scientific Computing", 2016, vol. 38, n° 3, pp. C232–C255 [*DOI :* 10.1137/15M1019222], https://hal.inria.fr/hal-01354902

[6] G. AUPY, M. SHANTHARAM, A. BENOIT, Y. ROBERT, P. RAGHAVAN. *Co-scheduling algorithms for high-throughput workload execution*, in "Journal of Scheduling", 2016, 14 p. [*DOI :* 10.1007/S10951-015-0445-X], https://hal.inria.fr/hal-01252366

[7] L. BAUTISTA-GOMEZ, A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Coping with recall and precision of soft error detectors*, in "Journal of Parallel and Distributed Computing", 2016, vol. 98, pp. 8–24 [*DOI :* 10.1016/J.JPDC.2016.07.007], https://hal.inria.fr/hal-01354888

[8] A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Assessing general-purpose algorithms to cope with fail-stop and silent errors*, in "ACM Transactions on Parallel Computing", 2016, https://hal.inria.fr/hal-01358146

[9] A. BENOIT, S. K. RAINA, Y. ROBERT. *Efficient checkpoint/verification patterns*, in "International Journal of High Performance Computing Applications", 2016 [*DOI :* 10.1177/1094342015594531], https://hal-ens-lyon.archives-ouvertes.fr/ensl-01252342

[10] M. BENZI, B. UÇAR. *Preconditioning Techniques Based on the Birkhoff–von Neumann Decomposition*, in "Computational Methods in Applied Mathematics", January 2016 [*DOI :* 10.1515/CMAM-2016-0040], https://hal.inria.fr/hal-01318486

[11] S. DI, Y. ROBERT, F. VIVIEN, F. CAPPELLO. *Toward an Optimal Online Checkpoint Solution under a Two-Level HPC Checkpoint Model*, in "IEEE Transactions on Parallel and Distributed Systems", January 2017, vol. 28, n^{o} 1, 16 p. [*DOI :* 10.1109/TPDS.2016.2546248], https://hal.inria.fr/hal-01353871

[12] F. DUFOSSÉ, B. UÇAR. *Notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices*, in "Linear Algebra and its Applications", February 2016, vol. 497, pp. 108–115 [*DOI :* 10.1016/J.LAA.2016.02.023], https://hal.inria.fr/hal-01270331

[13] M. FASI, J. LANGOU, Y. ROBERT, B. UÇAR. *A backward/forward recovery approach for the preconditioned conjugate gradient method*, in "Journal of Computational Science", 2016 [*DOI :* 10.1016/J.JOCS.2016.04.008], https://hal.inria.fr/hal-01354682

[14] 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

International Conferences with Proceedings

[15] C. ALIAS, C. FUHS, L. GONNORD. *Estimation of Parallel Complexity with Rewriting Techniques*, in "Workshop on Termination", Obergurgl, Austria, Workshop on Termination, September 2016, https://hal.archives-ouvertes.fr/hal-01345914

[16] M. BENDER, J. BERRY, R. JOHNSON, T. KROEGER, S. MCCAULEY, C. PHILLIPS, B. SIMON, S. SINGH, D. ZAGE. *Anti-Persistence on Persistent Storage: History-Independent Sparse Tables and Dictionaries*, in "Principle of Database Systems (PODS 2016)", San Francisco, United States, 2016 [*DOI :* 10.1145/2902251.2902276], https://hal.inria.fr/hal-01326312

[17] M. BENDER, R. CHOWDHURY, A. CONWAY, M. FARACH-COLTON, P. GANAPATHI, R. JOHNSON, S. MCCAULEY, B. SIMON, S. SINGH. *The I/O Complexity of Computing Prime Tables*, in "Latin American Theoretical Informatics Symposium", Ensenada, Mexico, LNCS, 2016, vol. 9644, pp. 192-206 [*DOI :* 10.1007/978-3-662-49529-2_15], https://hal.inria.fr/hal-01326317

[18] M. A. BENDER, S. MCCAULEY, B. SIMON, S. SINGH, F. VIVIEN. *Resource Optimization for Program Committee Members: A Subreview Article*, in "8th International Conference on Fun with Algorithms",

La Maddalena, Italy, Leibniz International Proceedings in Informatics (LIPIcs), 2016, vol. 49, n^{o} 8th International Conference on Fun with Algorithms (FUN 2016), 20 p. [*DOI :* 10.4230/LIPICS.FUN.2016.7], https://hal.inria.fr/hal-01326277

[19] A. BENOIT, A. CAVELAN, V. LE FÈVRE, Y. ROBERT, H. SUN. *A different re-execution speed can help*, in "5th International Workshop on Power-aware Algorithms, Systems, and Architectures (PASA'16), held in conjunction with ICPP 2016, the 45th International Conference on Parallel Processing", Philadelphia, United States, Proceedings of ICPP'2016 workshops (ICPPW'16), August 2016, https://hal.inria.fr/hal-01354887

[20] A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Optimal Resilience Patterns to Cope with Fail-Stop and Silent Errors*, in "IPDPS'2016, the 30th IEEE International Parallel and Distributed Processing Symposium", Chicago, United States, Proceedings of IPDPS'2016, the 30th IEEE International Parallel and Distributed Processing Symposium, IEEE Computer Society Press, May 2016 [*DOI :* 10.1109/IPDPS.2016.39], https://hal.inria.fr/hal-01354886

[21] A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Two-Level Checkpointing and Verifications for Linear Task Graphs*, in "The 17th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2016)", Chicago, United States, 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), IEEE, May 2016, 10 p. [*DOI :* 10.1109/IPDPSW.2016.106], https://hal.inria.fr/hal-01354625

[22] A. BENOIT, L. POTTIER, Y. ROBERT. *Resilient application co-scheduling with processor redistribution*, in "International Conference on Parallel Processing (ICPP)", Philadelphia, United States, The 45th International Conference on Parallel Processing, August 2016, https://hal.inria.fr/hal-01354863

[23] G. BOSILCA, A. BOUTEILLER, A. GUERMOUCHE, T. HERAULT, Y. ROBERT, P. SENS, J. DONGARRA. *Failure Detection and Propagation in HPC systems*, in "SC'2016 (SuperComputing)", Salt Lake City, United States, ACM, November 2016, https://hal.inria.fr/hal-01419279

[24] G. BOSILCA, A. BOUTEILLER, A. GUERMOUCHE, T. HÉRAULT, Y. ROBERT, P. SENS, J. DONGARRA. *Failure Detection and Propagation in HPC systems*, in "SC 2016 - The International Conference for High Performance Computing, Networking, Storage and Analysis", Salt Lake City, United States, November 2016, https://hal.inria.fr/hal-01352109

[25] H. CASANOVA, J. HERRMANN, Y. ROBERT. *Computing the expected makespan of task graphs in the presence of silent errors*, in "Ninth International Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2), 2016", Philadelphia, United States, Ninth International Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2), 2016, August 2016, https://hal.inria.fr/hal-01354711

[26] A. CAVELAN, J. LI, Y. ROBERT, H. SUN. *When Amdahl Meets Young/Daly*, in "Cluster'2016", Taipei, Taiwan, France, Cluster'2016, IEEE Computer Society, September 2016, https://hal.inria.fr/hal-01355963

[27] F. HANNA, L. MARCHAL, J.-M. NICOD, L. PHILIPPE, V. REHN-SONIGO, H. SABBAH. *Minimizing Rental Cost for Multiple Recipe Applications in the Cloud*, in "IPDPS Workshops", Chicago, United States, 2016 IEEE International Parallel and Distributed Processing Symposium Workshops, 2016, pp. 28–37 [*DOI :* 10.1109/IPDPSW.2016.71], https://hal.inria.fr/hal-01356152

[28] O. KAYA, B. UÇAR. *High Performance Parallel Algorithms for the Tucker Decomposition of Sparse Tensors*, in "International Conference on Parallel Processing (ICPP)", 2016-08-19, United States, August 2016, https://hal.inria.fr/hal-01354894

[29] M. MAALEJ, V. PAISANTE, R. PEDRO, L. GONNORD, F. M. QUINTÃO PEREIRA. *Pointer Disambiguation via Strict Inequalities*, in "Code Generation and Optimisation", Austin, United States, February 2017, https://hal.archives-ouvertes.fr/hal-01387031

[30] D. MONNIAUX, L. GONNORD. *Cell morphing: from array programs to array-free Horn clauses*, in "23rd Static Analysis Symposium (SAS 2016)", Edimbourg, United Kingdom, X. RIVAL (editor), Static Analysis Symposium, September 2016, https://hal.archives-ouvertes.fr/hal-01206882

[31] V. PAISANTE, M. MAALEJ, L. BARBOSA, L. GONNORD, F. M. QUINTÃO 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

[32] I. RAIS, L. LEFÈVRE, A. BENOIT, A.-C. ORGERIE. *An Analysis of the Feasibility of Energy Harvesting with Thermoelectric Generators on Petascale and Exascale Systems*, in "Workshop Optimization of Energy Efficient HPC & Distributed Systems (OPTIM 2016) - The 2016 International Conference on High Performance Computing & Simulation (HPCS 2016)", Innsbruck, Austria, Proceedings of the 2016 International Conference on High Performance Computing & Simulation (HPCS 2016), July 2016, https://hal.inria.fr/hal-01348554

Conferences without Proceedings

[33] I. RAIS, A. BENOIT, L. LEFÈVRE, A.-C. ORGERIE. *An Analysis of the Feasibility of Energy Harvesting with Thermoelectric Generators on Petascale and Exascale Systems*, in "Conférence d'informatique en Parallélisme, Architecture et Système (COMPAS 2016)", Lorient, France, Actes de COMPAS, la Conférence d'informatique en Parallélisme, Architecture et Système, July 2016, https://hal.inria.fr/hal-01348555

Scientific Books (or Scientific Book chapters)

[34] G. AUPY, A. BENOIT, A. CAVELAN, M. FASI, Y. ROBERT, H. SUN, B. UÇAR. *Coping with silent errors in HPC applications*, in "Emergent Computation", A. ADAMATZKY (editor), Springer Verlag, 2016, https://hal.inria.fr/hal-01354892

Books or Proceedings Editing

[35] A. H. GEBREMEDHIN, E. G. BOMAN, B. UÇAR (editors). *2016 Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing*, 2016 [*DOI :* 10.1137/1.9781611974690], https://hal.inria.fr/hal-01415503

Research Reports

[36] C. ALIAS, F. RASTELLO, A. PLESCO. *High-Level Synthesis of Pipelined FSM from Loop Nests*, Inria, April 2016, n^{o} 8900, 18 p. , https://hal.inria.fr/hal-01301334

[37] P. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY. *On the Complexity of the Block Low-Rank Multifrontal Factorization*, INPT-IRIT ; CNRS-IRIT ; Inria-LIP ; UPS-IRIT, May 2016, n^{o} IRIT/RT–2016–03–FR, 34 p. , https://hal.archives-ouvertes.fr/hal-01322230

[38] G. AUPY, A. BENOIT, L. POTTIER, P. RAGHAVAN, Y. ROBERT, M. SHANTHARAM. *Co-scheduling algorithms for cache-partitioned systems*, Inria Grenoble - Rhone-Alpes ; ENS de Lyon, November 2016, n^{o} RR-8965, 28 p. , https://hal.inria.fr/hal-01393989

[39] G. AUPY, C. BRASSEUR, L. MARCHAL. *Dynamic memory-aware task-tree scheduling*, Inria Grenoble - Rhone-Alpes, October 2016, n^{o} RR-8966, https://hal.inria.fr/hal-01390107

[40] G. AUPY, Y. ROBERT. *Scheduling for fault-tolerance: an introduction*, Inria, November 2016, n^{o} RR-8971, https://hal.inria.fr/hal-01393192

[41] O. BEAUMONT, T. LAMBERT, L. MARCHAL, B. THOMAS. *Matching-Based Allocation Strategies for Improving Data Locality of Map Tasks in MapReduce*, Inria - Research Centre Grenoble – Rhône-Alpes ; Inria Bordeaux Sud-Ouest, November 2016, n^{o} RR-8968, https://hal.inria.fr/hal-01386539

[42] A. BENOIT, A. CAVELAN, V. LE FÈVRE, Y. ROBERT, H. SUN. *A different re-execution speed can help*, Inria Grenoble - Rhone-Alpes, March 2016, n^{o} RR-8888, https://hal.inria.fr/hal-01297125

[43] A. BENOIT, A. CAVELAN, V. LE FÈVRE, Y. ROBERT, H. SUN. *Towards Optimal Multi-Level Checkpointing*, Inria Grenoble - Rhone-Alpes, June 2016, n^{o} RR-8930, https://hal.inria.fr/hal-01339788

[44] A. BENOIT, A. CAVELAN, Y. ROBERT, H. SUN. *Multi-level checkpointing and silent error detection for linear workflows*, Inria, September 2016, n^{o} RR-8952, https://hal.inria.fr/hal-01363581

[45] J. BRAINE, L. GONNORD, D. MONNIAUX. *Verifying Programs with Arrays and Lists*, ENS Lyon, June 2016, https://hal.archives-ouvertes.fr/hal-01337140

[46] A. CAVELAN, J. LI, Y. ROBERT, H. SUN. *When Amdahl Meets Young/Daly*, ENS Lyon, CNRS & Inria, February 2016, n^{o} RR-8871, https://hal.inria.fr/hal-01280004

[47] S. DI, Y. ROBERT, F. VIVIEN, F. CAPPELLO. *Toward an Optimal Online Checkpoint Solution under a Two-Level HPC Checkpoint Model*, Inria Grenoble - Rhone-Alpes, January 2016, n^{o} RR-8851, https://hal.inria.fr/hal-01263879

[48] M. FAVERGE, J. LANGOU, Y. ROBERT, J. DONGARRA. *Bidiagonalization with Parallel Tiled Algorithms*, Inria, October 2016, n^{o} RR-8969, https://hal.inria.fr/hal-01389232

[49] O. KAYA, B. UÇAR. *Parallel CP decomposition of sparse tensors using dimension trees*, Inria - Research Centre Grenoble – Rhône-Alpes, November 2016, n^{o} RR-8976, https://hal.inria.fr/hal-01397464

[50] E. KAYAASLAN, T. LAMBERT, L. MARCHAL, B. UÇAR. *Scheduling Series-Parallel Task Graphs to Minimize Peak Memory*, Inria Grenoble Rhône-Alpes, Université de Grenoble, November 2016, n^{o} RR-8975, https://hal.inria.fr/hal-01397299

[51] M. MAALEJ, V. PAISANTE, F. M. QUINTÃO PEREIRA, L. GONNORD. *Combining Range and Inequality Information for Pointer Disambiguation*, ENS Lyon, CNRS & Inria, December 2016, n^{o} RR-9009, https://hal.inria.fr/hal-01429777

[52] L. MARCHAL, B. SIMON, O. SINNEN, F. VIVIEN. *Malleable task-graph scheduling with a practical speed-up model*, ENS de Lyon, February 2016, n° RR-8856, https://hal.inria.fr/hal-01274099

[53] R. PORTASE, B. UÇAR. *On matrix symmetrization and sparse direct solvers*, Inria - Research Centre Grenoble – Rhône-Alpes, November 2016, n° RR-8977, https://hal.inria.fr/hal-01398951

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, https://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, https://www.top500.org/green500/lists/2015/11/

[62] 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

[63] 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

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

[65] 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° 2, pp. 202-217

[66] 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

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

[68] 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

[69] 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

[70] 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

[71] 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

[72] 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

[73] 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

[74] 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

[75] P. FEAUTRIER, C. LENGAUER. *The Polyhedron Model*, in "Encyclopedia of Parallel Programming", 2011

[76] 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

[77] 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

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

[79] 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

[80] 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

[81] 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

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

[83] 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

[84] 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