IN PARTNERSHIP WITH:
CNRS

Ecole normale supérieure de Lyon

Université Claude Bernard (Lyon 1)

Activity Report 2017

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:

A1.1.1. - Multicore, Manycore
A1.1.2. - Hardware accelerators (GPGPU, FPGA, etc.)
A1.1.3. - Memory models
A1.1.4. - High performance computing
A1.1.5. - Exascale
A1.1.9. - Fault tolerant systems
A1.6. - Green Computing
A6.1. - Mathematical Modeling
A6.2.3. - Probabilistic methods
A6.2.5. - Numerical Linear Algebra
A6.2.6. - Optimization
A6.2.7. - High performance computing
A6.3. - Computation-data interaction
A7.1. - Algorithms
A8.1. - Discrete mathematics, combinatorics
A8.2. - Optimization
A8.7. - Graph theory
A8.9. - Performance evaluation

Other Research Topics and Application Domains:

B3.2. - Climate and meteorology
B3.3. - Geosciences
B4. - Energy
B4.1. - Fossile energy production (oil, gas)
B4.5.1. - Green computing
B5.2.3. - Aviation
B5.5. - Materials

1. Personnel

Research Scientists

Frédéric Vivien [Team leader, Inria, Senior Researcher, HDR]
Christophe Alias [Inria, Researcher]
Jean-Yves L'Excellent [Inria, Researcher, HDR]
Loris Marchal [CNRS, Researcher]
Bora Uçar [CNRS, Researcher]

Faculty Members

Anne Benoit [Ecole Normale Supérieure Lyon, Associate Professor, HDR]
Louis-Claude Canon [Univ de Franche-Comté, Associate Professor]

Laure Gonnord [Univ de Claude Bernard, Associate Professor, HDR]
Matthieu Moy [Univ de Claude Bernard, Associate Professor, from Sep 2017, HDR]
Yves Robert [Ecole Normale Supérieure Lyon, Professor, HDR]

Post-Doctoral Fellow
Adrien Remy [Inria, from Apr 2017]

PhD Students
Aurélien Cavelan [Inria, until Aug 2017]
Changjiang Gou [ECNU/ENS Lyon]
Li Han [ECNU/ENS Lyon]
Oguz Kaya [Inria, until Sep 2017]
Aurélie Kong Win Chang [Ecole Normale Supérieure Lyon]
Valentin Le Fèvre [Ecole Normale Supérieure Lyon, from Sep 2017]
Maroua Maalej [Univ de Lyon, until Sep 2017]
Gilles Moreau [Inria]
Ioannis Panagiotas [Inria, from Oct 2017]
Loïc Pottier [Ecole Normale Supérieure Lyon]
Bertrand Simon [Ecole Normale Supérieure Lyon]
Issam Rais [Inria]

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

Interns
Szabolcs-Marton Bagoly [Ecole Normale Supérieure Lyon, from Jul 2017 until Sep 2017]
Julien Braine [Ecole Normale Supérieure Lyon, until Feb 2017]
Dorel Butaciu [Inria, from Jul 2017 until Sep 2017]
Hanna Nagy [Inria, from Jul 2017 until Sep 2017]

Administrative Assistants
Emeline Boyer [Inria, until April 2017]
Laetitia Gauthe [Inria, from May 2017]

External Collaborators
Patrick Amestoy [INP Toulouse, HDR]
Alfredo Buttari [CNRS]
Franck Cappello [Argonne National Laboratory – USA, HDR]

2. Overall Objectives

2.1. Overall Objectives

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

Scientists now have access to tremendous computing power. For instance, the four most powerful computing platforms in the TOP 500 list [65] each includes more than 500,000 cores and deliver a sustained performance of more than 10 Peta FLOPS. The volunteer computing platform BOINC [59] 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 [64], had an average of 2.33 faults per day during the period from August 2008 to February 2010 [90]. 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" [61] 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 [57] 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 [60]).

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 [71], [77], [78], [89], 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 $2 \leq \alpha \leq 3$ [69], [70], [75]; 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 [91] 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 [87]. 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 [91]; (ii) *replication* consists in executing the same task on several processors simultaneously, in order to meet the reliability constraints [68]; and (iii) *checkpointing* consists in "saving" the work done at some certain instants, hence reducing the amount of work lost when a failure occurs [86].

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 [58]. 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" [83]. 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, which 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 [79], [80], [85]. 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 [73] 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 [72] 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 [74], [88], [84]. 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 [76]. 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 software and hardware

Participants: Christophe Alias, Laure Gonnord, Matthieu Moy, Maroua Maalej [2014-2017].

Christophe Alias and Laure Gonnord asked to join the ROMA team temporarily, starting from September 2015. Matthieu Moy (formerly Grenoble INP) joined them in September 2017 to create a new team called CASH, for "Compilation and Analysis, Software and Hardware" (see https://matthieu-moy.fr/spip/?CASH-team-proposal). The proposal was accepted by the LIP laboratory and by Inria's "comité des équipes projet", and is waiting for final approval from Inria to officially become an "équipe centre". The text below describes their research domain. The results that they have achieved in 2017 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 reasonable 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. Dataflow models for HPC applications

In the last decades, several frameworks have emerged to design efficient compiler algorithms. The efficiency of all the optimizations performed in compilers strongly relies on performant *static analyses* and *intermediate representations*.

The contemporary computer, is constantly evolving. New architectures such as multi-core processors, Graphics Processing Units (GPUs) or many-core coprocessors are introduced, resulting into complex heterogeneous platforms.

A consequence of this diversity an heterogeneity is that a given computation can be implemented in many different ways, with different performance characteristics. As an obvious example, changing the degree of parallelism can trade execution time for number of cores. However, many choices are less obvious: for example, augmenting the degree of parallelism of a memory-bounded application will not improve performance. Most architectures involve a complex memory hierarchy, hence memory access patterns have a considerable impact on performance. The design-space to explore to find the best performance is much wider than it used to be with older architectures, and new tools are needed to help the programmer explore it. We believe that the dataflow formalism is a good basis to build such tools as it allows expressing different forms of parallelism.

The transverse theme of the CASH proposal is the study of the dataflow model for parallel programs: the dataflow formalism expresses a computation on an infinite number of values, that can be viewed as successive values of a variable during time. A dataflow program is structured as a set of *communicating processes* that communicate values through *communicating buffers*.

Examples of dataflow languages include the synchronous languages Lustre and Signal, as well as SigmaC; the DPN representation [67] (data-aware process network) is an example of a dataflow intermediate representation for a parallelizing compiler.

The dataflow model, which expresses at the same time data parallelism and task parallelism, is in our opinion one of the best models for analysis, verification and synthesis of parallel systems. This model will be our favorite representation for our programs. Indeed, it shares the "equational" description of computation and data with the polyhedral model, and the static single assignment representation inside compilers. The dataflow formalism can be used both as a programming language and as an intermediate representation within compilers.

This topic is transverse to the proposal. While we will not a priori restrict ourselves to dataflow applications (we also consider approaches to optimize CUDA and OpenCL code for example), it will be a good starting point and a convergence point to all the members of the team.

3.4.2. 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 [81] 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.

3.4.3. 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 [66], 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.

3.4.4. Simulation of Systems on a Chip

One of the bottlenecks in complex Systems on a Chip (SoCs) design flow is the simulation speed: it is necessary to be able to simulate the behavior of a complete system, including software, before the actual chip is available. Raising the level of abstraction from Register Transfer Level to more abstract simulations like Transaction Level Modeling (TLM) [63] in SystemC [62] allowed gaining several orders of magnitude of speed. We are particularly interested in the loosely timed coding style where the timing of the platform is not modeled precisely, and which allows the fastest simulations. Still, SystemC implementations used in production are still sequential, and one more order of magnitude in simulation speed could be obtained with proper parallelization techniques.

Matthieu Moy is a renown expert in the domain. He has worked on SystemC/TLM simulation with STMicroelectronics for 15 years. He presented part of his work on the subject to Collège de France with Laurent Maillet-Contoz from STMicroelectronics [1].

Matthieu Moy is the co-advisor (with Tanguy Sassolas) of the Phd of Gabriel Busnot, started in November 2017 in collaboration with CEA-LIST. The research will be on parallelizing SystemC simulation, and the first ideas include compilation techniques to discover process dependencies.

Work on SystemC/TLM parallel execution is both an application of other work on parallelism in the team and a tool complementary to HLS presented above. Indeed, some of the parallelization techniques we develop in CASH could apply to SystemC/TLM programs. Conversely, a complete design-flow based on HLS often needs fast system-level simulation: the full-system usually contains both parts designed using HLS, handwritten hardware components and software.

4. Application Domains

4.1. Applications of sparse direct solvers

Sparse direct (e.g., multifrontal solvers that we develop) 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

[1] http://www.college-de-france.fr/site/gerard-berry/seminar-2014-01-29-17h30.htm

often ends up solving a system of linear 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 are: structural mechanics, seismic modeling, 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 co-chair of ICPP'17 and of SC'17 (technical papers).
- Altair, EDF, Michelin, LSTC, and Total have renewed for three years their memberships in the MUMPS consortium.

5.1.1. Awards

- Aurélien Cavelan was awarded an accessit award for the Gilles Kahn 2017 PhD thesis award.

6. New Software and Platforms

6.1. 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: Alexandru Plesco and Christophe Alias
- Contact: Christophe Alias

6.2. 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).

RELEASE FUNCTIONAL DESCRIPTION: MUMPS versions 5.1.0, 5.1.1 and 5.1.2, all released in 2017 include many new features and improvements. The two main new features are Block Low-Rank compression, decreasing the complexity of sparse direct solvers for various types of applications, and selective 64-bit integers, allowing to process matrices with more than 2 billion entries.

- Participants: Gilles Moreau, Abdou Guermouche, Alfredo Buttari, Aurélia Fevre, Bora Uçar, Chiara Puglisi, Clément Weisbecker, Emmanuel Agullo, François-Henry Rouet, Guillaume Joslin, Jacko Koster, Jean-Yves L'excellent, Marie Durand, Maurice Bremond, Mohamed Sid-Lakhdar, Patrick Amestoy, Philippe Combes, Stéphane Pralet, Theo Mary and Tzvetomila Slavova
- Partners: Université de Bordeaux - CNRS - CERFACS - ENS Lyon - INPT - IRIT - Université de Lyon - Université de Toulouse - LIP
- Contact: Jean-Yves L'excellent
- URL: http://mumps-solver.org/

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

7. New Results

7.1. Acyclic partitioning of large directed acyclic graphs

Participant: Bora Uçar.

Finding a good partition of a computational directed acyclic graph associated with an algorithm can help find an execution pattern improving data locality, conduct an analysis of data movement, and expose parallel steps. The partition is required to be acyclic, i.e., the inter-part edges between the vertices from different parts should preserve an acyclic dependency structure among the parts. In this work [26], we adopt the multilevel approach with coarsening, initial partitioning, and refinement phases for acyclic partitioning of directed acyclic graphs and develop a direct k-way partitioning scheme. To the best of our knowledge, no such scheme exists in the literature. To ensure the acyclicity of the partition at all times, we propose novel and efficient coarsening and refinement heuristics. The quality of the computed acyclic partitions is assessed by computing the edge cut, the total volume of communication between the parts, and the critical path latencies. We use the solution returned by well-known undirected graph partitioners as a baseline to evaluate our acyclic partitioner, knowing that the space of solution is more restricted in our problem. The experiments are run on large graphs arising from linear algebra applications.

7.2. Further notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices

Participants: Ioannis Panagiotas, Bora Uçar.

The well-known Birkhoff-von Neumann (BvN) decomposition expresses a doubly stochastic matrix as a convex combination of a number of permutation matrices. For a given doubly stochastic matrix, there are many BvN decompositions, and finding the one with the minimum number of permutation matrices is NP-hard. There are heuristics to obtain BvN decompositions for a given doubly stochastic matrix. A family of heuristics are based on the original proof of Birkhoff and proceed step by step by subtracting a scalar multiple of a permutation matrix at each step from the current matrix, starting from the given matrix. At every step, the subtracted matrix contains nonzeros at the positions of some nonzero entries of the current matrix and annihilates at least one entry, while keeping the current matrix nonnegative. Our first result shows that this family of heuristics can miss optimal decompositions. We also investigate the performance of two heuristics from this family theoretically [46].

7.3. Low-Cost Approximation Algorithms for Scheduling Independent Tasks on Hybrid Platforms

Participants: Louis-Claude Canon, Loris Marchal, Frédéric Vivien.

Hybrid platforms embedding accelerators such as GPUs or Xeon Phis are increasingly used in computing. When scheduling tasks on such platforms, one has to take into account that a task execution time depends on the type of core used to execute it. We focus on the problem of minimizing the total completion time (or makespan) when scheduling independent tasks on two processor types, also known as the $(Pm, Pk)||C_{\max}$ problem. We propose BalanceEstimate and BalanceMakespan, two novel 2-approximation algorithms with low complexity. Their approximation ratio is both on par with the best approximation algorithms using dual approximation techniques (which are, thus, of high complexity) and significantly smaller than the approximation ratio of existing low-cost approximation algorithms. We compared both algorithms by simulations to existing strategies in different scenarios. These simulations showed that their performance is among the best ones in all cases.

This work has been presented at the EuroPar 2017 conference [22].

7.4. Memory-aware tree partitioning on homogeneous platforms

Participants: Anne Benoit, Changjiang Gou, Loris Marchal.

Scientific applications are commonly modeled as the processing of directed acyclic graphs of tasks, and for some of them, the graph takes the special form of a rooted tree. This tree expresses both the computational dependencies between tasks and their storage requirements. The problem of scheduling/traversing such a tree on a single processor to minimize its memory footprint has already been widely studied. Hence, we move to parallel processing and study how to partition the tree for a homogeneous multiprocessor platform, where each processor is equipped with its own memory. We formally state the problem of partitioning the tree into subtrees such that each subtree can be processed on a single processor and the total resulting processing time is minimized. We prove that the problem is NP-complete, and we design polynomial-time heuristics to address it. An extensive set of simulations demonstrates the usefulness of these heuristics.

This work has been accepted as a short paper in the PDP 2018 conference [50].

7.5. Parallel scheduling of DAGs under memory constraints.

Participants: Loris Marchal, Bertrand Simon, Frédéric Vivien.

Scientific workflows are frequently modeled as Directed Acyclic Graphs (DAG) of tasks, which represent computational modules and their dependencies, in the form of data produced by a task and used by another one. This formulation allows the use of runtime sys- tems which dynamically allocate tasks onto the resources of increasingly complex and hetero- geneous computing platforms. However, for some workflows, such a dynamic schedule may run out of memory by exposing too much parallelism. This work focuses on the problem of transforming such a DAG to prevent memory shortage, and concentrates on shared memory platforms. We first propose a simple model of DAG which is expressive enough to emulate com- plex memory behaviors. We then exhibit a polynomial-time algorithm that computes the max- imum peak memory of a DAG, that is, the maximum memory needed by any parallel schedule. We consider the problem of reducing this maximum peak memory to make it smaller than a given bound by adding new fictitious edges, while trying to minimize the critical path of the graph. After proving this problem NP-complete, we provide an ILP solution as well as several heuristic strategies that are thoroughly compared by simulation on synthetic DAGs modeling actual computational workflows. We show that on most instances, we are able to decrease the maximum peak memory at the cost of a small increase in the critical path, thus with little im- pact on quality of the final parallel schedule.

This work has been accepted for presentation at the IPDPS 2018 conference [56].

7.6. On the Complexity of the Block Low-Rank Multifrontal Factorization

Participants: Patrick Amestoy [INP-IRIT], Alfredo Buttari [CNRS-IRIT], Jean-Yves L'Excellent, Théo Mary [UPS-IRIT].

Matrices coming from elliptic Partial Differential Equations have been shown to have a low-rank property: well defined off-diagonal blocks of their Schur complements can be approximated by low-rank products and this property can be efficiently exploited in multifrontal solvers to provide a substantial reduction of their complexity. Among the possible low-rank formats, the Block Low-Rank format (BLR) is easy to use in a general purpose multifrontal solver and has been shown to provide significant gains compared to full-rank on practical applications. However, unlike hierarchical formats, such as $\mathcal{H}$ and HSS, its theoretical complexity was unknown. We extended the theoretical work done on hierarchical matrices in order to compute the theoretical complexity of the BLR multifrontal factorization. We then studied several variants of the BLR multifrontal factorization, depending on the strategies used to perform the updates in the frontal matrices and on the constraints on how numerical pivoting is handled. We showed that these variants can further reduce the complexity of the factorization. In the best case (3D, constant ranks), we obtain a complexity of the order of $O(n^{4/3})$. We provide an experimental study with numerical results to support our complexity bounds.

This work has been published in the SIAM Journal on Scientific Computing [6].

7.7. Large-scale 3-D EM modelling with a Block Low-Rank multifrontal direct solver

Participants: Daniil Shantsev [EMGS-Univ. Oslo], Piyoosh Jaysaval [Univ. Oslo], Sébastien de La Kethulle de Ryhove [EMGS], Patrick Amestoy [INP-IRIT], Alfredo Buttari [CNRS-IRIT], Jean-Yves L'Excellent, Théo Mary [UPS-IRIT].

We put forward the idea of using a Block Low-Rank (BLR) multifrontal direct solver to efficiently solve the linear systems of equations arising from a finite-difference discretization of the frequency-domain Maxwell equations for 3-D electromagnetic (EM) problems. The solver uses a low-rank representation for the off-diagonal blocks of the intermediate dense matrices arising in the multifrontal method to reduce the computational load. A numerical threshold, the so-called BLR threshold, controlling the accuracy of low-rank representations was optimized by balancing errors in the computed EM fields against savings in floating point operations (flops). Simulations were carried out over large-scale 3-D resistivity models representing typical scenarios for marine controlled-source EM surveys, and in particular the SEG SEAM model which contains an irregular salt body. The flop count, size of factor matrices and elapsed run time for matrix factorization are reduced dramatically by using BLR representations and can go down to, respectively, 10, 30 and 40 per cent

of their full-rank values for our largest system with N = 20.6 million unknowns. The reductions are almost independent of the number of MPI tasks and threads at least up to $90 \times 10 = 900$ cores. The BLR savings increase for larger systems, which reduces the factorization flop complexity from $O(N^2)$ for the full-rank solver to $O(N^m)$ with $m = 1.4$–1.6. The BLR savings are significantly larger for deep-water environments that exclude the highly resistive air layer from the computational domain. A study in a scenario where simulations are required at multiple source locations shows that the BLR solver can become competitive in comparison to iterative solvers as an engine for 3-D controlled-source electromagnetic Gauss–Newton inversion that requires forward modelling for a few thousand right-hand sides.

This work has been published in the Geophysical Journal International [16].

7.8. On exploiting sparsity of multiple right-hand sides in sparse direct solvers

Participants: Patrick Amestoy [INP-IRIT], Jean-Yves L'Excellent, Gilles Moreau.

The cost of the solution phase in sparse direct methods is sometimes critical. It can be larger than the one of the factorization in applications where systems of linear equations with thousands of right-hand sides (RHS) must be solved. In this work, we focus on the case of multiple *sparse* RHS with different nonzero structures in each column. Given a factorization $A = LU$ of a sparse matrix A and the system $AX = B$ (or $LY = B$ when focusing on the forward elimination), the sparsity of B can be exploited in two ways. First, *vertical* sparsity is exploited by pruning unnecessary nodes from the elimination tree, which represents the dependencies between computations in a direct method. Second, we explain how *horizontal* sparsity can be exploited by working on a subset of RHS columns at each node of the tree. A combinatorial problem must then be solved in order to permute the columns of B and minimize the number of operations. We propose a new algorithm to build such a permutation, based on the tree and on the sparsity structure of B. We then propose an original approach to split the columns of B into a minimal number of blocks (to preserve flexibility in the implementation or maintain high arithmetic intensity, for example), while reducing the number of operations down to a given threshold. Both algorithms are motivated by geometric intuitions and designed using an algebraic approach, and they can be applied to general systems of linear equations. We demonstrate the effectiveness of our algorithms on systems coming from real applications and compare them to other standard approaches. Finally, we give some perspectives and possible applications for this work.

This work is available as a research report [34] and has been submitted to a journal.

7.9. Revisiting temporal failure independence in large scale systems

Participants: Guillaume Aupy [Inria Tadaam], Leonardo Bautista Gomez [Barcelona Supercomputing Center, Spain], Yves Robert, Frédéric Vivien.

This work revisits the *failure temporal independence* hypothesis which is omnipresent in the analysis of resilience methods for HPC. We explain why a previous approach is incorrect, and we propose a new method to detect failure cascades, i.e., series of non-independent consecutive failures. We use this new method to assess whether public archive failure logs contain failure cascades. Then we design and compare several cascade-aware checkpointing algorithms to quantify the maximum gain that could be obtained, and we report extensive simulation results with archive and synthetic failure logs. Altogether, there are a few logs that contain cascades, but we show that the gain that can be achieved from this knowledge is not significant. The conclusion is that we can wrongly, but safely, assume failure independence!

This work is available as a research report and has been submitted to a journal. A preliminary version appears in the proceedings of the FTS'17 workshop.

7.10. Co-scheduling Amdahl applications on cache-partitioned systems

Participants: Guillaume Aupy [Inria Tadaam], Anne Benoit, Sicheng Dai [East China Normal University, China], Loïc Pottier, Padma Raghavan [Vanderbilt University, Nashville TN, USA], Yves Robert, Manu Shantharam [San Diego Supercomputer Center, San Diego CA, USA].

Cache-partitioned architectures allow subsections of the shared last-level cache (LLC) to be exclusively reserved for some applications. This technique dramatically limits interactions between applications that are concurrently executing on a multi-core machine. Consider n applications that execute concurrently, with the objective to minimize the makespan, defined as the maximum completion time of the n applications. Key scheduling questions are: (i) which proportion of cache and (ii) how many processors should be given to each application? In this work, we provide answers to (i) and (ii) for Amdahl applications. Even though the problem is shown to be NP-complete, we give key elements to determine the subset of applications that should share the LLC (while remaining ones only use their smaller private cache). Building upon these results, we design efficient heuristics for Amdahl applications. Extensive simulations demonstrate the usefulness of co-scheduling when our efficient cache partitioning strategies are deployed.

This work is available as a research report and has been accepted for publication in the IJHPCA journal.

7.11. Coping with silent and fail-stop errors at scale by combining replication and checkpointing

Participants: Anne Benoit, Franck Cappello [Argonne National Laboratory, USA], Aurélien Cavelan [University of Basel, Switzerland], Padma Raghavan [Vanderbilt University, Nashville TN, USA], Yves Robert, Hongyang Sun [Vanderbilt University, Nashville TN, USA].

This work provides a model and an analytical study of replication as a technique to detect and correct silent errors, as well as to cope with both silent and fail-stop errors on large-scale platforms. Fail-stop errors are immediately detected, unlike silent errors for which a detection mechanism is required. To detect silent errors, many application-specific techniques are available, either based on algorithms (ABFT), invariant preservation or data analytics, but replication remains the most transparent and least intrusive technique. We explore the right level (duplication, triplication or more) of replication for two frameworks: (i) when the platform is subject only to silent errors, and (ii) when the platform is subject to both silent and fail-stop errors. A higher level of replication is more expensive in terms of resource usage but enables to tolerate more errors and to correct some silent errors, hence there is a trade-off to be found. Replication is combined with checkpointing and comes with two flavors: *process replication* and *group replication*. Process replication applies to message-passing applications with communicating processes. Each process is replicated, and the platform is composed of process pairs, or triplets. Group replication applies to black-box applications, whose parallel execution is replicated several times. The platform is partitioned into two halves (or three thirds). In both scenarios, results are compared before each checkpoint, which is taken only when both results (duplication) or two out of three results (triplication) coincide. If not, one or more silent errors have been detected, and the application rolls back to the last checkpoint, as well as when fail-stop errors have struck. We provide a detailed analytical study for all of these scenarios, with formulas to decide, for each scenario, the optimal parameters as a function of the error rate, checkpoint cost, and platform size. We also report a set of extensive simulation results that nicely corroborates the analytical model.

This work is available as a research report and has been submitted to a journal. A preliminary version appears in the proceedings of the FTXS'17 workshop.

7.12. Optimal checkpointing period with replicated execution on heterogeneous platforms

Participants: Anne Benoit, Aurélien Cavelan [University of Basel, Switzerland], Valentin Le Fèvre, Yves Robert.

In this work, we design and analyze strategies to replicate the execution of an application on two different platforms subject to failures, using checkpointing on a shared stable storage. We derive the optimal pattern size W for a periodic checkpointing strategy where both platforms concurrently try and execute W units of work before checkpointing. The first platform that completes its pattern takes a checkpoint, and the other platform interrupts its execution to synchronize from that checkpoint. We compare this strategy to a simpler on-failure checkpointing strategy, where a checkpoint is taken by one platform only whenever the other platform

encounters a failure. We use first or second-order approximations to compute overheads and optimal pattern sizes, and show through extensive simulations that these models are very accurate. The simulations show the usefulness of a secondary platform to reduce execution time, even when the platforms have relatively different speeds: in average, over a wide range of scenarios, the overhead is reduced by 30%. The simulations also demonstrate that the periodic checkpointing strategy is globally more efficient, unless platform speeds are quite close.

This work is available as a research report. A preliminary version appears in the proceedings of the FTXS'17 workshop.

7.13. A Failure Detector for HPC Platforms

Participants: George Bosilca [ICL, University of Tennessee Knoxville, USA], Aurélien Bouteiller [ICL, University of Tennessee Knoxville, USA], Amina Guermouche [Telecom SudParis, France], Thomas Hérault [ICL, University of Tennessee Knoxville, USA], Yves Robert, Pierre Sens [LIP6, Université Paris 6, France].

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, that can 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 is available as a research report and has been accepted for publication in the IJHPCA journal. A preliminary version appears in the proceedings of the SC'16 conference.

7.14. Budget-aware scheduling algorithms for scientific workflows on IaaS cloud platforms

Participants: Yves Caniou [Inria Avalon], Eddy Caron [Inria Avalon], Aurélie Kong Win Chang, Yves Robert.

This work introduces several budget-aware algorithms to deploy scientific workflows on IaaS cloud platforms, where users can request Virtual Machines (VMs) of different types, each with specific cost and speed parameters. We use a realistic application/platform model with stochastic task weights, and VMs communicating through a datacenter. We extend two well-known algorithms, HEFT and MinMin, and make scheduling decisions based upon machine availability *and* available budget. During the mapping process, the budget-aware algorithms make conservative assumptions to avoid exceeding the initial budget; we further improve our results with refined versions that aim at re-scheduling some tasks onto faster VMs, thereby spending any budget fraction leftover by the first allocation. These refined variants are much more time-consuming than the former algorithms, so there is a trade-off to find in terms of scalability. We report an extensive set of simulations with workflows from the Pegasus benchmark suite. Budget-aware algorithms generally succeed in achieving efficient makespans while enforcing the given budget, and despite the uncertainty in task weights.

This work is available as a research report and has been submitted to a journal.

7.15. Resilience for stencil computations with latent errors

Participants: Aurélien Cavelan [University of Basel, Switzerland], Andrew Chien [University of Chicago, USA], Aiman Fang [University of Chicago, USA], Yves Robert.

Projections and measurements of error rates in near-exascale and exascale systems suggest a dramatic growth, due to extreme scale (10^9 cores), concurrency, software complexity, and deep submicron transistor scaling. Such a growth makes resilience a critical concern, and may increase the incidence of errors that "escape", silently corrupting application state. Such errors can often be revealed by application software tests but with long latencies, and thus are known as *latent errors*. We explore how to efficiently recover from latent errors, with an approach called application-based focused recovery (ABFR). Specifically we present a case study of stencil computations, a widely useful computational structure, showing how ABFR focuses recovery effort where needed, using intelligent testing and pruning to reduce recovery effort, and enables recovery effort to be overlapped with application computation. We analyze and characterize the ABFR approach on stencils, creating a performance model parameterized by error rate and detection interval (latency). We compare projections from the model to experimental results with the Chombo stencil application, validating the model and showing that ABFR on stencil can achieve a significant reductions in error recovery cost (up to 400x) and recovery latency (up to 4x). Such reductions enable efficient execution at scale with high latent error rates.

This work is available as a research report . A short version appears in the proceedings of the ICPP'17 conference.

7.16. Checkpointing workflows for fail-stop errors

Participants: Louis-Claude Canon, Henri Casanova [University of Hawai'i at Manoa, USA], Li Han, Yves Robert, Frédéric Vivien.

We consider the problem of orchestrating the execution of workflow applications structured as Directed Acyclic Graphs (DAGs) on parallel computing platforms that are subject to fail-stop failures. The objective is to minimize expected overall execution time, or makespan. A solution to this problem consists of a schedule of the workflow tasks on the available processors and of a decision of which application data to checkpoint to stable storage, so as to mitigate the impact of processor failures. For general DAGs this problem is hopelessly intractable. In fact, given a solution, computing its expected makespan is still a difficult problem. To address this challenge, we consider a restricted class of graphs, Minimal Series-Parallel Graphs (GSPGs). It turns out that many real-world workflow applications are naturally structured as GSPGs. For this class of graphs, we propose a recursive list-scheduling algorithm that exploits the GSPG structure to assign sub-graphs to individual processors, and uses dynamic programming to decide which tasks in these sub-gaphs should be checkpointed. Furthermore, it is possible to efficiently compute the expected makespan for the solution produced by this algorithm, using a first-order approximation of task weights and existing evaluation algorithms for 2-state probabilistic DAGs. We assess the performance of our algorithm for production workflow configurations, comparing it to (i) an approach in which all application data is checkpointed, which corresponds to the standard way in which most production workflows are executed today; and (ii) an approach in which no application data is checkpointed. Our results demonstrate that our algorithm strikes a good compromise between these two approaches, leading to lower checkpointing overhead than the former and to better resilience to failure than the latter. To the best of our knowledge, this is the first scheduling/checkpointing algorithm for workflow applications with fail-stop failures that considers workflow structures more general than mere linear chains of tasks.

This work is available as a research report and has been submitted to a journal. A short version appears in the proceedings of the IEEE Cluster'17 conference.

7.17. Optimal Cooperative Checkpointing for Shared High-Performance Computing Platforms

Participants: Dorian Arnold [Emory University, Atlanta, GA, USA], George Bosilca [ICL, University of Tennessee Knoxville, USA], Aurélien Bouteiller [ICL, University of Tennessee Knoxville, USA], Jack Dongarra [ICL, University of Tennessee Knoxville, USA], Kurt Ferreira [Center for Computing Research, Sandia National Laboratory, USA], Thomas Hérault [ICL, University of Tennessee Knoxville, USA], Yves Robert.

In high-performance computing environments, input/output (I/O) from various sources often contend for scarce available bandwidth. Adding to the I/O operations inherent to the failure-free execution of an application, I/O from checkpoint/restart (CR) operations (used to ensure progress in the presence of failures) places an additional burden as it increases I/O contention, leading to degraded performance. In this work, we consider a cooperative scheduling policy that optimizes the overall performance of concurrently executing CR-based applications which share valuable I/O resources. First, we provide a theoretical model and then derive a set of necessary constraints needed to minimize the global *waste* on the platform. Our results demonstrate that the optimal checkpoint interval as defined by Young/Daly, while providing a sensible metric for a single application, is not sufficient to optimally address resource contention at the platform scale. We therefore show that combining optimal checkpointing periods with I/O scheduling strategies can provide a significant improvement on the overall application performance, thereby maximizing platform throughput. Overall, these results provide critical analysis and direct guidance on checkpointing large-scale workloads in the presence of competing I/O while minimizing the impact on application performance.

This work is available as a research report and has been submitted to a conference.

7.18. Parallel Code Generation of Synchronous Programs for a Many-core Architecture

Participant: Matthieu Moy.

Embedded systems tend to require more and more computational power. Many-core architectures are good candidates since they offer power and are considered more time predictable than classical multi-cores.

Data-flow Synchronous languages such as Lustre or Scade are widely used for avionic critical software. Programs are described by networks of computational nodes. Implementation of such programs on a many-core architecture must ensure a bounded response time and preserve the functional behavior by taking interference into account.

We consider the top-level node of a Lustre application as a software architecture description where each sub-node corresponds to a potential parallel task. Given a mapping (tasks to cores), we automatically generate code suitable for the targeted many-core architecture. This code uses hardware synchronization mechanisms and time-triggered execution. This minimizes memory interferences and allows usage of a framework to compute the Worst-Case Response Time.

This work was accepted for publication at the DATE 2018 conference [82].

7.19. Optimizing Affine Control with Semantic Factorizations

Participants: Christophe Alias, Alexandru Plesco.

Hardware accelerators generated by polyhedral synthesis techniques make an extensive use of affine expressions (affine functions and convex polyhedra) in control and steering logic. Since the control is pipelined, these affine objects must be evaluated at the same time for different values, which forbids aggressive reuse of operators.

In this work, we propose a method to factorize a collection of affine expressions without preventing pipelining. Our key contributions are (i) to use semantic factorizations exploiting arithmetic properties of addition and multiplication and (ii) to rely on a cost function whose minimization ensures a correct usage of FPGA resources. Our algorithm is totally parametrized by the cost function, which can be customized to fit a target FPGA. Experimental results on a large pool of linear algebra kernels show a significant improvement compared to traditional low-level RTL optimizations. In particular, we show how our method reduces resource consumption by revealing hidden strength reductions.

This work has been published in ACM TACO [5]

7.20. Improving Communication Patterns in Polyhedral Process Networks

Participant: Christophe Alias.

Process networks are a natural intermediate representation for HLS and more generally automatic parallelization. Compiler optimizations for parallelism and data locality restructure deeply the execution order of the processes, hence the read/write patterns in communication channels. This breaks most FIFO channels, which have to be implemented with addressable buffers. Expensive hardware is required to enforce synchronizations, which often results in dramatic performance loss.

In this work, we present an algorithm to partition the communications so that most FIFO channels can be recovered after a loop tiling, a key optimization for parallelism and data locality. Experimental results show a drastic improvement of FIFO detection for regular kernels at the cost of (few) additional storage. As a bonus, the storage can even be reduced in some cases.

This work will be presented at the HiP3ES'2018 workshop [32].

7.21. Static Analyses of pointers

Participants: Laure Gonnord, Maroua Maalej.

The design and implementation of static analyses that disambiguate pointers has been a focus of research since the early days of compiler construction. One of the challenges that arise in this context is the analysis of languages that support pointer arithmetics, such as C, C++ and assembly dialects. In 2017, we contributed to this research area with a conference paper and a journal paper.

The CGO'17 paper[27] contributes to solve this challenge. We start 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.

Pentagons is an abstract domain invented by Logozzo and Fahndrich to validate array accesses in low-level programming languages. This algebraic structure provides a cheap "less-than check", which builds a partial order between the integer variables used in a program. In the Science of Computer Programming journal paper[15], we show how we have used the ideas available in Pentagons to design and implement a novel alias analysis. With this new algorithm, we are able to disambiguate pointers with off- sets, that commonly occur in C programs, in a precise and efficient way. Together with this new abstract domain we describe several implementation decisions that let us produce a practical pointer disambiguation algorithm on top of the LLVM compiler. Our alias analysis is able to handle programs as large as SPEC CPU2006's gcc in a few minutes. Furthermore, it improves on LLVM's industrial quality analyses. As an extreme example, we have observed a 4x improvement when analyzing SPEC's lbm.

7.22. Dataflow static analyses and optimisations

Participants: Laure Gonnord, Lionel Morel, Szabolcs-Marton Bagoly, Romain Fontaine.

Nowadays, parallel computers have become ubiquitous and current processors contain several execution cores, anywhere from a couple to hundreds. This multi-core tendency is due to constraints preventing the increase of clock frequencies, such as heat generation and power consumption. A variety of low-level tools exist to program these chips efficiently, but they are considered hard to program, to maintain, and to debug, because they may exhibit non-deterministic behaviors. We explore the potentiality of the data flow programming, which allows the programmer to specify only the operations to perform and their dependencies, without actually scheduling them. The work is published in two research reports: [48] and [49].

In [48], we explore the combination of a dataflow paradigm language, SigmaC, with the Polyhedral Model, which allows automatic parallelization and optimization of loop nests, in order to make the programming easier by delegating work to the compilers and static analyzers, in various case studies.

In [49], we explore the expressivity of the horn clause format for static analyses of Lustre programs with arrays. We propose a translation from a Lustre core language to horn clauses, with or without array variables.

8. Bilateral Contracts and Grants with Industry

8.1. MUMPS Consortium

In 2017, in the context of the MUMPS consortium (http://mumps-consortium.org), we worked in close collaboration with Toulouse INP to:

- sign or renew membership contracts with EDF, Altair, Michelin, LSTC, FFT-MSC, and with the Lawrence Berkeley National Laboratory, on top of the ongoing contracts with ESI-Group, Safran, Siemens and Total,
- organize point-to-point meetings with several members,
- provide technical support and scientific advice to members,
- provide experimental releases to members in advance,
- organize the third consortium committee meeting, at Altair (Grenoble).

Three engineers have been funded by the membership fees in 2017, for software engineering and software development, performance study and tuning, business development and management of the consortium. Half a year of a PhD student was funded by the membership fees (see Section 9.1). On top of their membership, an additional contract was signed with Michelin to provide a new functionality and study how to best exploit MUMPS recent features in their computing environment.

8.2. The XtremLogic Start-Up

XTREMLOGIC is a spin-off of Inria founded 6 years ago by Alexandru Plesco and Christophe Alias.

XTREMLOGIC leverages the results obtained in both HPC and polyhedral compilation communities to synthesize energy-efficient circuits for FPGA. The circuits commercialized by XTREMLOGIC target markets including HPC, data centers and artificial intelligence. The compiler technology transfered to XTREMLOGIC is the result of a tight collaboration between Christophe Alias and Alexandru Plesco.

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

9. Partnerships and Cooperations

9.1. Regional Initiatives

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

The doctoral program from Labex MILYON dedicated to applied research in collaboration with industrial partners funds 50% of a 3-year PhD grant (the other 50% being funded by the MUMPS consortium) to work on improvements of the solution phase of the MUMPS solver. The PhD aims at answering industrial needs in application domains where the cost of the solution phase of sparse direct solvers is critical.

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.

ANR JCJC Project CODAS (2018-2022), 4 years. The ANR project CODAS was accepted in July 2017. He will be launched in February 2018. Il gathers a little team of five persons including Laure Gonnord (PI) and Christophe Alias.

This project aims as studying the combination of formal methods such as abstract interpretation and term rewriting to address the challenge of scheduling complex data structures as well as complex flow graph.

9.3. International Initiatives

9.3.1. Inria International Labs

9.3.1.1. 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 scheduling and resilience. Yves Robert is the Inria executive director of JLESC.

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

9.3.2.1. Keystone

Title: Scheduling algorithms for sparse linear algebra at extreme scale

International Partner (Institution - Laboratory - Researcher):

Vanderbilt University (United States) - Electrical Engineering and Computer Science - 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.

9.3.3. Inria International Partners

9.3.3.1. Declared Inria International Partners

- Anne Benoit, Frederic 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.
- Laure Gonnord has a regular collaboration with Sylvain Collange (Inria Rennes) in the context of the PROSPIEL associate team.

9.3.4. 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 HPC computing. Anne Benoit and Minsong Chen are leading a JORISS project on scheduling and resilience in cloud computing. In the context of this collaboration two students from ECNU, Li Han and Changjiang Gou, have joined Roma for their PhD.

9.3.4.1. Informal International Partners

- Christophe Alias has a regular collaboration with Sanjay Rajopadhye from Colorado State University (USA); this collaboration also includes Guillaume Iooss (Inria Parkas) and Sylvain Collange (Inria Rennes).

9.4. International Research Visitors

9.4.1. Visits of International Scientists

9.4.1.1. Internships

- Louis-Claude Canon, Loris Marchal, and Frédéric Vivien supervised Dorel Butaciu, an Erasmus student, for three months (June–September 2017).
- Loris Marchal, Bertrand Simon and Frédéric Vivien supervised Hanna Nagy, an Erasmus student, for three months (June–September 2017).
- Laure Gonnord supervised Szabolcs-Marton Bagoly, an Erasmus student, for three months (June–September 2017).

9.4.2. Visits to International Teams

9.4.2.1. 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.
- Anne Benoit and Bora Uçar visit the School of Computational Science and Engineering Georgia Institute of Technology, Atlanta, GA, USA (August 2017–May 2018). During this stay, Anne Benoit taught the course CSE-6140 Computational Science and Engineering (CSE) Algorithms, taken by both senior level undergraduate and graduate students, and by distant learners. Anne and Bora are collaborating with Prof. Çatalyürek and his group members on problems of high performance computing including partitioning, load balancing and scheduling.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events Organisation

10.1.1.1. Member of the Organizing Committees

Marie Durand, Guillaume Joslin and Chiara Puglisi organized the fourth edition of the MUMPS User days in Montbonnot on June 1 and 2, 2017, see http://mumps.enseeiht.fr/index.php?page=ud_2017.

10.1.2. Scientific Events Selection

10.1.2.1. Steering committees

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

Bora Uçar serves in the steering committee of CSC.

10.1.2.2. Chair of Conference Program Committees

- Anne Benoit is the technical program chair of ICPP 2017, the technical papers co-chair of SC 2017, and the technical program chair of IPDPS 2018.
- Bora Uçar was the IPDPS 2017 Workshops Chair, IC3 2017 Algorithms track vice-chair. He is the general chair of IPDPS 2018.
- Yves Robert is the Panels Chair of SC'17.
- Yves Roibert serves as the Liaison between the steering committee and the program committee of IPDPS.

10.1.2.3. Member of the Conference Program Committees

- Matthieu Moy was a member of the program committee of the DUHDE and COMPASS workshops.
- Laure Gonnord was a member of the program committee of the VMCAI conference and NSAD workshop.
- Christophe Alias was a member of the program committee of the IMPACT and HIP3ES workshops.

- Loris Marchal was/is a member of the program committee of the SC 2017, IPDPS 2017 and IPDPS 2018 conferences.
- Bora Uçar was a member of the program committee of IA^3, ICCS 2017, Euro-Par 2017, ScalCom 2017, P^{3}MA, HiPC 2017, PPAM 2017, HPC4BD 2017.
- Yves Robert was a member of the program committee of FTS, FTXS, ICCS, and ISCIS
- Frédéric Vivien was a member of the program committee of EduPar 17, IPDPS 2017 and 2018, PDP 2017, SBAC-PAD 2017, and SC 2017.

10.1.2.4. Reviewer

- Matthieu Moy was reviewer for ECRTS.
- Laure Gonnord was reviewer for STACS, VMCAI, NSAD.
- Bora Uçar reviewed papers for IEEE Cluster 2017, ICPP 2017, IPDPS 2017, and ALENEX 2017.

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

- Anne Benoit is Associate Editor for JPDC (Elsevier Journal of Parallel and Distributed Computing) and TPDS (IEEE Transactions on Parallel and Distributed Systems).
- Yves Robert is Associate Editor for ACM TOPC (ACM Transactions on Parallel Computing), 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 a member of the editorial board of Parallel Computing.
- 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

- Matthieu Moy was reviewer for MICPRO 2017 (2 papers) and TCAD 2017.
- Christophe Alias was reviewer for ACM TACO and Proceedings of the IEEE.
- Loris Marchal was a reviewer for IEEE TPDS.
- Bora Uçar reviewed papers for SIAM SISC (5x), SIMAX (1x), IEEE TPDS (1x), Parallel Computing (2ex), Journal of Computational Science (1x).

10.1.4. Invited Talks

- Matthieu Moy gave an invited talk at "Journées nationales du GDR GPL", Montpellier, 13 - 16 June 2017.
- Laure Gonnord gave invited talks to University of Nice and University of Evry (spring 2017) and to University of Reunion (dec 2017).
- Christophe Alias gave invited talks to "Journées Compilation" (june 2017) and "Journée Calcul" (nov 2017).
- Yves Robert gave an invited talk at the FTS'17 workshop, Honolulu (September 2017).

10.1.5. Tutorials

Yves Robert gave a tutorial on *Fault-tolerance techniques for HPC platforms* at SC'17 (with Aurélien Bouteiller, George Bosilca, and Thomas Hérault).

10.1.6. Research Administration

Jean-Yves L'Excellent was a member of the direction board of the LIP laboratory until August 2017.

Yves Robert 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 the vide-head of the LIP laboratory since September 2017. He 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:

- Yves Robert, Algorithmique, L3, ENS Lyon
- Anne Benoit, Algorithmique avancée, L3, ENS Lyon
- Matthieu Moy, programmation concurrente, 24h EqTD, L3, UCBL, France
- Matthieu Moy, Unix, 9h EqTD, L1, UCBL, France
- Matthieu Moy, Projet informatique, 40h EqTD, L3, UCBL, France
- Matthieu Moy, Informatique, 30h EqTD, L1, Grenoble INP CPP, France
- Christophe Alias, Compilation (CM+TD 48h), Insa Centre Val de Loire

Master:

- Yves Robert is the head of the Master in Computer Science, ENS Lyon
- Anne Benoit, CSE Algorithms, undergraduate + graduate, Georgia Tech, USA
- Yves Robert, Scheduling at Scale, M2, ENS Lyon
- Matthieu Moy, programmation avancée, 15h EqTD, M1, UCBL, France
- Matthieu Moy, Projet de spécialité informatique, 18h EqTD, M1, Grenoble INP Ensimag, France
- Laure Gonnord, Compilation (CM+TD 76h), M1, Université Claude Bernard Lyon 1, and M1 Ecole Normale Supérieure de Lyon.
- Laure Gonnord, Complexité, M1 (TD 15h), Université Claude Bernard Lyon 1.
- Laure Gonnord, Compilation and Softawre Engineering(CM+TP 10h, with S. Mosser), M2 Ecole Normale Supérieure de Lyon.
- Christophe Alias, Optimisation d'applications embarquées (CM+TD 26h), Insa Centre Val de Loire
- Loris Marchal, Complexité, M1 (TD 15h), Université Claude Bernard Lyon 1.
- Frédéric Vivien, Algorithmique et Programmation Parallèles et Distribuées (CM 36 h), M1, École normale supérieure de Lyon, France.

Professional training:

- Jean-Yves L'Excellent taught during the training "Solvers for Computer Aided Engineering" (Jan. 30–Feb. 3 2017) co-organized by ENS Paris-Saclay, Hewlett-Packard Enterprises, Ansys, MUMPS, CMLA, Intel.

- Loris Marchal is responsible of the competitive selection of the students of ENS Lyon for Computer Science.

10.2.2. Supervision

PhD in progress : Amaury Graillat, Génération de code pour un many-core avec des contraintes temps réel fortes, started in Sept. 2015, supervised by Matthieu Moy (LIP) and Pascal Raymond (Verimag).

PhD in progress : Tristan Delizy, Gestion mémoire pour architectures à base de NVRAM, started in Oct. 2016, co-supervised by Tanguy Risset (CITI), Guillaume Salagnac (CITI), Kevin Marquet (CITI) and Matthieu Moy (LIP).

PhD in progress: Gabriel Busnot, Accélération SystemC pour la co-simulation multi- physique et la simulation de modèles hétérogènes en complexité, started in Nov. 2017. Co-supervised by Matthieu Moy (LIP) and Tanguy Sassolas (CEA LIST).

PhD defended in December 2017: Hamza Rihani, Many-Core Timing Analysis of Real-Time Systems and its Application to an Industrial Processor, Univ. Grenoble Alpes, defended on Dec. 1st 2017. Supervised by Matthieu Moy (LIP) and Claire Maiza (Verimag).

PhD defended in December 2017: Denis Becker, Parallel SystemC/TLM Simulation of Hardware Components Described for High-Level Synthesis, Univ. Grenoble Alpes, defended on Dec. 11st 2017. Supervised by Matthieu Moy (LIP) and Jérôme Cornet (STMicroelectronics).

PhD in progress: Amaury Graillat, Génération de code pour un many-core avec des contraintes temps réel fortes, started in Sept. 2015, supervised by Matthieu Moy (LIP) and Pascal Raymond (Verimag).

Phd defended in September 2017: Maroua Maalej “Low cost static analyses for compilers”, started in October 2014, advisors : Laure Gonnord and Frédéric Vivien.

HdR: Laure Gonnord, Contributions to program analysis: expressivity and scalability, defended in November 2018.

PhD in progress: Bertrand Simon, Memory aware task graph scheduling, started in Sept. 2015, supervised by Loris Marchal and Frédéric Vivien.

PhD in progress: Changjiang Gou, Communication- and memory-aware task graph scheduling, started in Oct. 2016, supervised by Anne Benoit and Loris Marchal.

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 Dec. 2015, supervised by Jean-Yves L’Excellent and Patrick Amestoy.

PhD defended in July 2017: 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 defended in September 2017: 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: 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: Valentin Le Fèvre, “Scheduling and resilience at scale”, started in October 2017, funding: ENS Lyon, advisors: Anne Benoit and Yves Robert.

PhD in progress: Ioannis Panagiotas, “High performance algorithms for big data graph and hypergraph problems”, started in October 2017, funding: Inria, advisors: Frédéric Vivien and Bora Uçar.

10.2.3. Juries

- Anne Benoit was "rapporteur" in the Jury of Thomas Lambert (U. Bordeaux) in September 2017.

- Yves Robert was “président du jury” for the Habilitation à Diriger des Rechereches of Abdou Guermouche and Pierre Ramet at Inria Bordeaux in November 2017.
- Matthieu Moy was examiner in the jury of Cédric BEN AOUN (LIP6, UPMC) in July 2017.
- Laure Gonnord was “rapporteur” in the Jury of Thomas Rubiano (LIPN) in December 2017.
- Laure Gonnord was examiner in the Jury of Jacob Lidman (Chalmers, Sweden) in December 2017.
- Laure Gonnord was examiner in the Jury of Maurica Fonenantsoa (Univ. Réunion) in december 2017.
- Laure Gonnord participated to the selection committee recruting an assistant professor (MCF) at University of Grenoble, in Spring 2017.
- Laure Gonnord participated to the selection committee recruting an assistant professor (MCF) at ENS Paris, in Spring 2017.
- Christophe Alias was “membre du jury” and “correcteur” for “concours E3A, épreuve d’Informatique MP” in May 2017.
- Jean-Yves L’Excellent was invited in the PhD jury of Théo Mary in November 2017.
- Yves Robert was “président du jury” for the Habilitation à Diriger des Rechereches of Abdou Guermouche and Pierre Ramet at Inria Bordeaux in November 2017.

11. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] A. CAVELAN. *Scheduling algorithms and resilience patterns for fail-stop and silent errors*, Université de Lyon, July 2017, https://tel.archives-ouvertes.fr/tel-01582228

[2] L. GONNORD. *Contributions to program analysis: expressivity and scalability*, Université Lyon 1 Claude Bernard, November 2017, Habilitation à diriger des recherches, https://tel.archives-ouvertes.fr/tel-01633065

[3] O. KAYA. *High Performance Parallel Algorithms for Tensor Decompositions*, Université de Lyon, September 2017, https://tel.archives-ouvertes.fr/tel-01623523

[4] M. MAALEJ. *Low-cost memory analyses for efficient compilers*, UNIVERSITÉ DE LYON, September 2017, https://hal.inria.fr/tel-01626398

Articles in International Peer-Reviewed Journals

[5] C. ALIAS, A. PLESCO. *Optimizing Affine Control with Semantic Factorizations*, in "ACM Transactions on Architecture and Code Optimization (TACO) ", December 2017, vol. 14, n^{o} 4, 27 p. , https://hal.inria.fr/hal-01470873

[6] P. AMESTOY, A. BUTTARI, J.-Y. L’EXCELLENT, T. MARY. *On the Complexity of the Block Low-Rank Multifrontal Factorization*, in "SIAM Journal on Scientific Computing", 2017, vol. 39, n^{o} 4, pp. A1710 - A1740 [*DOI :* 10.1137/16M1077192], https://hal.inria.fr/hal-01672943

[7] G. AUPY, A. BENOIT, S. DAI, L. POTTIER, P. RAGHAVAN, Y. ROBERT, M. SHANTHARAM. *Co-scheduling Amdahl applications on cache-partitioned systems*, in "International Journal of High Performance Computing Applications", June 2017 [*DOI :* 10.1177/1094342017710806], https://hal.archives-ouvertes.fr/hal-01670137

[8] G. AUPY, J. HERRMANN. *Periodicity in optimal hierarchical checkpointing schemes for adjoint computations*, in "Optimization Methods & Software", 2017, vol. 32, n^{o} 3, pp. 594-624 [*DOI :* 10.1080/10556788.2016.1230612], https://hal.inria.fr/hal-01654632

[9] A. BENOIT, L. LEFÈVRE, A.-C. ORGERIE, I. RAÏS. *Reducing the energy consumption of large scale computing systems through combined shutdown policies with multiple constraints*, in "International Journal of High Performance Computing Applications", January 2018, vol. 32, n^{o} 1, pp. 176-188 [*DOI :* 10.1177/1094342017714530], https://hal.inria.fr/hal-01557025

[10] A. BENOIT, L. POTTIER, Y. ROBERT. *Resilient co-scheduling of malleable applications*, in "International Journal of High Performance Computing Applications", May 2017 [*DOI :* 10.1177/1094342017704979], https://hal.archives-ouvertes.fr/hal-01670153

[11] G. BOSILCA, A. BOUTEILLER, A. GUERMOUCHE, T. HÉRAULT, Y. ROBERT, P. SENS, J. DONGARRA. *A Failure Detector for HPC Platforms*, in "International Journal of High Performance Computing Applications", 2017, https://hal.inria.fr/hal-01531522

[12] L.-C. CANON, P.-C. HÉAM, L. PHILIPPE. *Controlling the Correlation of Cost Matrices to Assess Scheduling Algorithm Performance on Heterogeneous Platforms*, in "Concurrency and Computation: Practice and Experience", June 2017 [*DOI :* 10.1002/CPE.4185], https://hal.inria.fr/hal-01664629

[13] L.-C. CANON, L. PHILIPPE. *On the Heterogeneity Bias of Cost Matrices for Assessing Scheduling Algorithms*, in "IEEE Transactions on Parallel and Distributed Systems", June 2017, https://hal.inria.fr/hal-01664636

[14] 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

[15] M. MAALEJ, V. PAISANTE, F. MAGNO QUINTAO PEREIRA, L. GONNORD. *Combining Range and Inequality Information for Pointer Disambiguation*, in "Science of Computer Programming", 2017, Final published version of https://hal.inria.fr/hal-01429777v2, https://hal.archives-ouvertes.fr/hal-01625402

[16] D. SHANTSEV, P. JAYSAVAL, S. DE LA KETHULLE DE RYHOVE, P. R. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY. *Large-scale 3-D EM modelling with a Block Low-Rank multifrontal direct solver*, in "Geophysical Journal International", June 2017, vol. 209, n^{o} 3, pp. 1558 - 1571 [*DOI :* 10.1093/GJI/GGX106], https://hal.inria.fr/hal-01672952

International Conferences with Proceedings

[17] G. AUPY, A. BENOIT, L. POTTIER, P. RAGHAVAN, Y. ROBERT, M. SHANTHARAM. *Co-scheduling algorithms for cache-partitioned systems*, in "APDCM 2017 - 19th Workshop on Advances in Parallel and Distributed Computational Models", Orlando (FL), United States, Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International, IEEE, May 2017, pp. 1-10 [*DOI :* 10.1109/IPDPSW.2017.60], https://hal.inria.fr/hal-01654660

[18] G. AUPY, C. BRASSEUR, L. MARCHAL. *Dynamic Memory-Aware Task-Tree Scheduling*, in "IPDPS 2017 - 31st IEEE International Parallel & Distributed Processing Symposium", Orlando, United States, proceedings of IPDPS 2017, May 2017, 10 p. , https://hal.inria.fr/hal-01472062

[19] G. AUPY, A. GAINARU, V. LE FÈVRE. *Periodic I/O scheduling for super-computers*, in "PMBS 2017 - 8th International Workshop High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation", Denver (CO), United States, November 2017, pp. 1-22, https://hal.inria.fr/hal-01654645

[20] G. AUPY, Y. ROBERT, F. VIVIEN. *Assuming failure independence: are we right to be wrong?*, in "FTS 2017 - 3rd International Workshop on Fault-Tolerant Systems", Honolulu (HI), United States, September 2017, pp. 1-8, https://hal.inria.fr/hal-01654639

[21] A. BENOIT, L. LEFÈVRE, A.-C. ORGERIE, I. RAÏS. *Shutdown Policies with Power Capping for Large Scale Computing Systems*, in "Euro-Par: International European Conference on Parallel and Distributed Computing", Santiago de Compostela, Spain, F. F. RIVERA, T. F. PENA, J. C. CABALEIRO (editors), Lecture Notes in Computer Science, springer, August 2017, vol. 10417, pp. 134 - 146 [*DOI :* 10.1109/COMST.2016.2545109], https://hal.archives-ouvertes.fr/hal-01589555

[22] L.-C. CANON, L. MARCHAL, F. VIVIEN. *Low-Cost Approximation Algorithms for Scheduling Independent Tasks on Hybrid Platforms*, in "Euro-Par 2017: 23rd International European Conference on Parallel and Distributed Computing", Santiago de Compostela, Spain, Springer, August 2017, https://hal.inria.fr/hal-01559898

[23] M. FAVERGE, J. LANGOU, Y. ROBERT, J. DONGARRA. *Bidiagonalization and R-Bidiagonalization: Parallel Tiled Algorithms, Critical Paths and Distributed-Memory Implementation*, in "IPDPS'17 - 31st IEEE International Parallel and Distributed Processing Symposium", Orlando, United States, May 2017, https://hal.inria.fr/hal-01484113

[24] A. GRAILLAT, M. MOY, P. RAYMOND, B. DUPONT DE DINECHIN. *Parallel Code Generation of Synchronous Programs for a Many-core Architecture*, in "Design, Automation and Test in Europe", Dresden, Germany, March 2018, https://hal.inria.fr/hal-01667594

[25] L. HAN, L.-C. CANON, H. CASANOVA, Y. ROBERT, F. VIVIEN. *Checkpointing Workflows for Fail-Stop Errors*, in "IEEE Cluster 2017", Honolulu, United States, September 2017, https://hal.inria.fr/hal-01559967

[26] J. HERRMANN, J. KHO, B. UÇAR, K. KAYA, U. V. CATALYUREK. *Acyclic Partitioning of Large Directed Acyclic Graphs*, in "CCGRID 2017 - 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing", Madrid, Spain, May 2017, pp. 371-380 [*DOI :* 10.1109/CCGRID.2017.101], https://hal.inria.fr/hal-01672010

[27] M. MAALEJ, V. PAISANTE, R. PEDRO, L. GONNORD, F. PEREIRA. *Pointer Disambiguation via Strict Inequalities*, in "Code Generation and Optimisation", Austin, United States, February 2017, pp. 134-147, https://hal.archives-ouvertes.fr/hal-01387031

[28] L. MARCHAL, S. MCCAULEY, B. SIMON, F. VIVIEN. *Minimizing I/Os in Out-of-Core Task Tree Scheduling*, in "19th Workshop on Advances in Parallel and Distributed Computational Models", Orlando, United States, May 2017, https://hal.inria.fr/hal-01491969

Conferences without Proceedings

[29] Y. CANIOU, E. CARON, A. KONG WIN CHANG, Y. ROBERT. *Budget-aware scheduling algorithms for scientific workflows on IaaS Cloud platforms*, in "WORKS 2017 - 12th Workshop Workflows in Support

of Large-Scale Science", Denver, United States, November 2017, 1 p. , https://hal.archives-ouvertes.fr/hal-01678736

[30] A. KONG WIN CHANG. *Ordonnancement multi-objectifs de workflows dans le cloud : un modèle plus réaliste avec tâches de durée stochastique*, in "Compas 2017 - Conférence d'informatique en Parallélisme, Architecture et Système", Sophia Antipolis, France, June 2017, pp. 1-7, https://hal.archives-ouvertes.fr/hal-01679699

Research Reports

[31] E. AGULLO, A. BUTTARI, M. BYCKLING, A. GUERMOUCHE, I. MASLIAH. *Achieving high-performance with a sparse direct solver on Intel KNL*, Inria Bordeaux Sud-Ouest ; CNRS-IRIT ; Intel corporation ; Université Bordeaux, February 2017, n° RR-9035, 15 p. , https://hal.inria.fr/hal-01473475

[32] C. ALIAS. *Improving Communication Patterns in Polyhedral Process Networks*, Inria Grenoble - Rhône-Alpes, December 2017, n° RR-9131, pp. 1-13, https://hal.inria.fr/hal-01665155

[33] P. R. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY. *Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures*, INPT-IRIT ; CNRS-IRIT ; Inria-LIP ; UPS-IRIT, April 2017, https://hal.archives-ouvertes.fr/hal-01505070

[34] P. AMESTOY, J.-Y. L'EXCELLENT, G. MOREAU. *On Exploiting Sparsity of Multiple Right-Hand Sides in Sparse Direct Solvers*, ENS de Lyon ; Inria Grenoble - Rhone-Alpes, December 2017, n° RR-9122, pp. 1-28, https://hal.inria.fr/hal-01649244

[35] G. AUPY, L. BAUTISTA GOMEZ, Y. ROBERT, F. VIVIEN. *Revisiting temporal failure independence in large scale systems*, Inria, December 2017, n° RR-9134, https://hal.inria.fr/hal-01672404

[36] G. AUPY, A. BENOIT, S. DAI, L. POTTIER, P. RAGHAVAN, Y. ROBERT, M. SHANTHARAM. *Co-scheduling Amdahl applications on cache-partitioned systems*, Inria, February 2017, n° RR-9021, 33 p. , https://hal.inria.fr/hal-01461157

[37] G. AUPY, A. GAINARU, V. LE FÈVRE. *Periodic I/O scheduling for super-computers*, Inria Bordeaux Sud-Ouest, February 2017, n° RR-9037, https://hal.inria.fr/hal-01474553

[38] G. AUPY, Y. ROBERT, F. VIVIEN. *Assuming failure independence: are we right to be wrong?*, Inria, July 2017, n° RR-9078, https://hal.inria.fr/hal-01556292

[39] O. BEAUMONT, T. LAMBERT, L. MARCHAL, B. THOMAS. *Matching-Based Assignement Strategies for Improving Data Locality of Map Tasks in MapReduce*, Inria - Research Centre Grenoble – Rhône-Alpes ; Inria Bordeaux Sud-Ouest, February 2017, n° RR-8968, https://hal.inria.fr/hal-01386539

[40] A. BENOIT, A. CAVELAN, F. CAPPELLO, P. RAGHAVAN, Y. ROBERT, H. SUN. *Coping with silent and fail-stop errors at scale by combining replication and checkpointing*, University of Basel ; Ecole Normale Supérieure de Lyon - ENS LYON ; Vanderbilt University ; University of Tennessee Knoxville, USA ; Argonne National Laboratory, October 2017, n° RR-9106, https://hal.inria.fr/hal-01616514

[41] A. BENOIT, A. CAVELAN, F. CAPPELLO, P. RAGHAVAN, Y. ROBERT, H. SUN. *Identifying the right replication level to detect and correct silent errors at scale*, Inria Grenoble Rhône-Alpes, Université de Grenoble, March 2017, n° RR-9047, https://hal.inria.fr/hal-01494678

[42] A. BENOIT, A. CAVELAN, V. LE FÈVRE, Y. ROBERT. *Optimal checkpointing period with replicated execution on heterogeneous platforms*, Inria, April 2017, n° RR-9055, https://hal.inria.fr/hal-01504936

[43] G. BOSILCA, A. BOUTEILLER, A. GUERMOUCHE, T. HÉRAULT, Y. ROBERT, P. SENS, J. DONGARRA. *A Failure Detector for HPC Platforms*, Inria, February 2017, n° RR-9024, https://hal.inria.fr/hal-01453086

[44] Y. CANIOU, E. CARON, A. KONG WIN CHANG, Y. ROBERT. *Budget-aware scheduling algorithms for scientific workflows on IaaS cloud platforms*, Inria, August 2017, n° RR-9088, 27 p. , https://hal.inria.fr/hal-01574491

[45] Y. CANIOU, E. CARON, A. KONG WIN CHANG, Y. ROBERT. *Budget-aware scheduling algorithms for scientific workflows with stochastic task weights on IaaS Cloud platforms*, Inria, November 2017, n° RR-9128, pp. 1-28, https://hal.inria.fr/hal-01651149

[46] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Further notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices*, Inria - Research Centre Grenoble – Rhône-Alpes, September 2017, n° RR-9095, https://hal.inria.fr/hal-01586245

[47] A. A. FANG, A. A. CAVELAN, Y. ROBERT, A. A. CHIEN. *Resilience for Stencil Computations with Latent Errors*, Inria, March 2017, n° RR-9042, https://hal.inria.fr/hal-01488409

[48] R. FONTAINE, L. MOREL, L. GONNORD. *Combining dataflow programming and polyhedral optimization, a case study*, Inria Rhône-Alpes ; CITI - CITI Centre of Innovation in Telecommunications and Integration of services ; LIP - ENS Lyon, July 2017, n° RT-0490, 40 p. , https://hal.archives-ouvertes.fr/hal-01572439

[49] L. GONNORD, S.-M. BAGOLY, L. MOREL. *Static Analysis via Horn Encoding from synchronous Dataflow Programs*, Université Lyon 1 Claude Bernard, LIP & INSA, CITI , October 2017, n° RT-0492, 25 p. , https://hal.inria.fr/hal-01614637

[50] C. GOU, A. BENOIT, L. MARCHAL. *Memory-aware tree partitioning on homogeneous platforms*, Inria Grenoble Rhône-Alpes, November 2017, n° RR-9115, pp. 1-25, https://hal.inria.fr/hal-01644352

[51] L. HAN, L.-C. CANON, H. CASANOVA, Y. ROBERT, F. VIVIEN. *Checkpointing Workflows for Fail-Stop Errors*, Inria, November 2017, n° RR-9068, 33 p. , https://hal.inria.fr/hal-01525378

[52] T. HÉRAULT, Y. ROBERT, A. BOUTEILLER, D. ARNOLD, K. B. FERREIRA, G. BOSILCA, J. DONGARRA. *Optimal Cooperative Checkpointing for Shared High-Performance Computing Platforms*, Inria, October 2017, n° RR-9109, pp. 1-20, https://hal.inria.fr/hal-01621295

[53] O. KAYA, Y. ROBERT, B. UÇAR. *Computing Dense Tensor Decompositions with Optimal Dimension Trees*, Inria, July 2017, n° RR-9080, https://hal.inria.fr/hal-01562399

[54] L. MARCHAL, L.-C. CANON, F. VIVIEN. *Low-Cost Approximation Algorithms for Scheduling Independent Tasks on Hybrid Platforms*, Inria - Research Centre Grenoble – Rhône-Alpes, June 2017, n° RR-9029, https://hal.inria.fr/hal-01475884

[55] L. MARCHAL, S. MCCAULEY, B. SIMON, F. VIVIEN. *Minimizing I/Os in Out-of-Core Task Tree Scheduling*, Inria, February 2017, n° RR-9025, https://hal.inria.fr/hal-01462213

[56] L. MARCHAL, H. NAGY, B. SIMON, F. VIVIEN. *Parallel scheduling of DAGs under memory constraints*, LIP - ENS Lyon, 2017, n° RR-9108, https://hal.inria.fr/hal-01620255

References in notes

[57] *Blue Waters Newsletter*, dec 2012

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

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

[60] *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

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

[62] *IEEE 1666 Standard: SystemC Language Reference Manual*, Open SystemC Initiative, 2011, http://www.accellera.org/

[63] *OSCI TLM-2.0 Language Reference Manual*, Open SystemC Initiative (OSCI), June 2008, http://www.accellera.org/downloads/standards

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

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

[66] *The Green500 List - November*, 2015, https://www.top500.org/green500/lists/2015/11/

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

[68] 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

[69] 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

[70] 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

[71] 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

[72] 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

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

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

[75] 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

[76] 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

[77] 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° 6, pp. 1317-1336

[78] 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

[79] 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

[80] 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

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

[82] A. GRAILLAT, M. MOY, P. RAYMOND, B. DUPONT DE DINECHIN. *Parallel Code Generation of Synchronous Programs for a Many-core Architecture*, in "Design, Automation and Test in Europe", Dresden, Germany, March 2018, https://hal.inria.fr/hal-01667594

[83] 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

[84] 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

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

[86] 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

[87] 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

[88] 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

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

[90] 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

[91] 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