IN PARTNERSHIP WITH:
CNRS

Ecole normale supérieure de Lyon

Université Claude Bernard (Lyon 1)

Activity Report 2019

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. - Methods in 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. Team, Visitors, External Collaborators

Research Scientists

Frédéric Vivien [Team leader, Inria, Senior Researcher, HDR]
Jean-Yves L'Excellent [Inria, Researcher, until Jan 2019, HDR]
Loris Marchal [CNRS, Researcher, HDR]
Bora Uçar [CNRS, Researcher, HDR]

Faculty Members

Anne Benoit [École Normale Supérieure de Lyon, Associate Professor, HDR]
Grégoire Pichon [Univ Claude Bernard, Associate Professor, from Sep 2019]
Yves Robert [École Normale Supérieure de Lyon, Professor, HDR]

Samuel Thibault [Univ de Bordeaux, Associate Professor, from Feb 2019 until Jul 2019, HDR]

PhD Students

Yishu Du [China Scholarship Council, PhD Student, from Dec 2019]
Yiqin Gao [Univ de Lyon, PhD Student]
Changjiang Gou [China Scholarship Council, PhD Student]
Li Han [China Scholarship Council, PhD Student]
Aurélie Kong Win Chang [École Normale Supérieure de Lyon, PhD Student]
Valentin Le Fèvre [École Normale Supérieure de Lyon, PhD Student]
Ioannis Panagiotas [Inria, PhD Student]
Filip Pawlowski [CIFRE Huawei, PhD Student]

Interns and Apprentices

Gabriel Bathie [École Normale Supérieure de Lyon, from Jun 2019 until Jul 2019]
Anthony Dugois [Univ Claude Bernard, from May 2019 until Jul 2019]
Pierre Antoine Rouby [Inria, from May 2019 until Jul 2019]

Administrative Assistants

Solene Audoux [Inria, Administrative Assistant, until Apr 2019]
Evelyne Blesle [Inria, Administrative Assistant, from Sep 2019]
Virginie Bouyer [Inria, Administrative Assistant, from Apr 2019 until Aug 2019]

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 [45] each includes more than 500,000 cores and deliver a sustained performance of more than 10 Peta FLOPS. The volunteer computing platform BOINC [41] 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 [44], had an average of 2.33 faults per day during the period from August 2008 to February 2010 [68]. 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" [43] 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 [39] 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 [42]).

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 [50], [56], [57], [67], 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 $2 \leq \alpha \leq 3$ [48], [49], [54]; 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 [69] 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 [65]. 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 [69]; (ii) *replication* consists in executing the same task on several processors simultaneously, in order to meet the reliability constraints [47]; and (iii) *checkpointing* consists in "saving" the work done at some certain instants, hence reducing the amount of work lost when a failure occurs [64].

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 [40]. 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" [60]. 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 [58], [59], [63]. 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 general sparse matrix and tensor computations; and (ii) the mapping of tasks (mostly the sub-tasks of the mentioned solvers) onto modern computing platforms. We focus on the development and the 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 [52] 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 [51] 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 [53], [66], [61]. 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 [55]. 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.

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 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 the 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

- Jean-Yves L'Excellent co-created the MUMPS technologies start-up and left the team to work full time for MUMPS technologies.
- Grégoire Pichon joined the team as an Associate Professor of University Claude Bernard, Lyon 1.
- Anne Benoit was elected chair of the IEEE Technical Committee on Parallel Processing.
- Anne Benoit received on February 2019 the award for Editorial Excellence as Associate Editor of the IEEE Transactions on Parallel and Distributed Systems during 2018.
- Yves Robert received the 2020 IEEE-CS Charles Babbage Award *for contributions to parallel algorithms and scheduling techniques.* This award covers all aspects of parallel computing including computational aspects, novel applications, parallel algorithms, theory of parallel computation, parallel computing technologies, among others. Further information about the award, including a list of past recipients, may be found at https://www.computer.org/web/awards/charles-babbage. The award consists of a $1,000 honorarium, certificate, and the invitation to present a paper and/or presentation at the annual IEEE-CS International Parallel and Distributed Processing Symposium (IPDPS).

5.1.1. Awards

BEST PAPER AWARD:

[16]

F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Effective heuristics for matchings in hypergraphs*, in "SEA2 2019 - International Symposium on Experimental Algorithms - Special Event", Kalamata, Greece, Springer, 2019, pp. 248-264 [*DOI :* 10.1007/978-3-030-34029-2_17], https://hal.inria.fr/hal-02417475

6. New Software and Platforms

6.1. MUMPS

A MUltifrontal Massively Parallel Solver

KEYWORDS: High-Performance Computing - Direct solvers - Finite element modelling

FUNCTIONAL DESCRIPTION: MUMPS is a software library to solve large sparse linear systems (AX=B) on sequential and parallel distributed memory computers. It implements a sparse direct method called the multifrontal method. It is used worldwide in academic and industrial codes, in the context numerical modeling of physical phenomena with finite elements. Its main characteristics are its numerical stability, its large number of features, its high performance and its constant evolution through research and feedback from its community of users. Examples of application fields include structural mechanics, electromagnetism, geophysics, acoustics, computational fluid dynamics. MUMPS has been developed by INPT(ENSEEIHT)-IRIT, Inria, CERFACS, University of Bordeaux, CNRS and ENS Lyon. Since January 2019, it is developed and licensed by Mumps Technologies SAS.

NEWS OF THE YEAR: In June 2019, a new version of MUMPS, MUMPS 5.2.1, was released by Mumps Technologies.

- 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 Brémond, 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 - Mumps Technologies SAS
- Contact: Jean-Yves L'Excellent
- URL: http://mumps-solver.org/

7. New Results

7.1. Creation of the start-up "Mumps Technologies SAS"

In January 2019, Jean-Yves L'Excellent left the ROMA team to co-found with Patrick Amestoy and Chiara Puglisi the company "Mumps Technologies" around the free software library MUMPS (Cecill-C licence). MUMPS solves large systems of sparse linear equations on high-performance computers in a robust and effective way. Mumps Technologies carries on collaborations and R&D activities to keep the MUMPS software library state-of-the-art and freely available, while offering to its clients a set of services.

7.2. Scheduling independent stochastic tasks under deadline and budget constraints

This work discusses scheduling strategies for the problem of maximizing the expected number of tasks that can be executed on a cloud platform within a given budget and under a deadline constraint. The execution times of tasks follow IID probability laws. The main questions are how many processors to enroll and whether and when to interrupt tasks that have been executing for some time. We provide complexity results and an asymptotically optimal strategy for the problem instance with discrete probability distributions and without deadline. We extend the latter strategy for the general case with continuous distributions and a deadline and we design an efficient heuristic which is shown to outperform standard approaches when running simulations for a variety of useful distribution laws.

The findings were published in a journal [8].

7.3. Online scheduling of task graphs on heterogeneous platforms

Modern computing platforms commonly include accelerators. We target the problem of scheduling applications modeled as task graphs on hybrid platforms made of two types of resources, such as CPUs and GPUs. We consider that task graphs are uncovered dynamically, and that the scheduler has information only on the available tasks, i.e., tasks whose predecessors have all been completed. Each task can be processed by either a CPU or a GPU, and the corresponding processing times are known. Our study extends a previous $4\sqrt{m/k}$-competitive online algorithm by Amaris et al. [46], where m is the number of CPUs and k the number of GPUs ($m \geq k$). We prove that no online algorithm can have a competitive ratio smaller than $\sqrt{m/k}$. We also study how adding flexibility on task processing, such as task migration or spoliation, or increasing the knowledge of the scheduler by providing it with information on the task graph, influences the lower bound. We provide a $(2\sqrt{m/k}+1)$-competitive algorithm as well as a tunable combination of a system-oriented heuristic and a competitive algorithm; this combination performs well in practice and has a competitive ratio in $\Theta(\sqrt{m/k})$. We also adapt all our results to the case of multiple types of processors. Finally, simulations on different sets of task graphs illustrate how the instance properties impact the performance of the studied algorithms and show that our proposed tunable algorithm performs the best among the online algorithms in almost all cases and has even performance close to an offline algorithm.

The findings were published in a journal [9].

7.4. A generic approach to scheduling and checkpointing workflows

This work deals with scheduling and checkpointing strategies to execute scientific workflows on failure-prone large-scale platforms. To the best of our knowledge, this work is the first to target fail-stop errors for arbitrary workflows. Most previous work addresses soft errors, which corrupt the task being executed by a processor but do not cause the entire memory of that processor to be lost, contrarily to fail-stop errors. We revisit classical mapping heuristics such as HEFT and MINMIN and complement them with several checkpointing strategies. The objective is to derive an efficient trade-off between checkpointing every task (CKPTALL), which is an overkill when failures are rare events, and checkpointing no task (CKPTNONE), which induces dramatic re-execution overhead even when only a few failures strike during execution. Contrarily to previous work, our approach applies to arbitrary workflows, not just special classes of dependence graphs such as M-SPGS (Minimal Series-Parallel Graphs). Extensive experiments report significant gain over both CKPTALL and CKPTNONE, for a wide variety of workflows.

The findings were published in a journal [10].

7.5. Limiting the memory footprint when dynamically scheduling DAGs on shared-memory platforms

Scientific workflows are frequently modeled as Directed Acyclic Graphs (DAGs) of tasks, which represent computational modules and their dependences in the form of data produced by a task and used by another one. This formulation allows the use of runtime systems which dynamically allocate tasks onto the resources of increasingly complex computing platforms. However, for some workflows, such a dynamic schedule may run out of memory by processing too many tasks simultaneously. This paper 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 DAGs which is expressive enough to emulate complex memory behaviors. We then exhibit a polynomial-time algorithm that computes the maximum 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. Our solution consists in adding new fictitious edges, while trying to minimize the critical path of the graph. After proving that this problem is 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 impact on the quality of the final parallel schedule.

The findings were published in a journal [12].

7.6. Scheduling independent stochastic tasks on heterogeneous cloud platforms

This work introduces scheduling strategies to maximize the expected number of independent tasks that can be executed on a cloud platform within a given budget and under a deadline constraint. The cloud platform is composed of several types of virtual machines (VMs), where each type has a unit execution cost that depends upon its characteristics. The amount of budget spent during the execution of a task on a given VM is the product of its execution length by the unit execution cost of that VM. The execution lengths of tasks follow a variety of standard probability distributions (exponential, uniform, half-normal, etc.), which is known beforehand and whose mean and standard deviation both depend upon the VM type. Finally, there is a global available budget and a deadline constraint, and the goal is to successfully execute as many tasks as possible before the deadline is reached or the budget is exhausted (whichever comes first). On each VM, the scheduler can decide at any instant to interrupt the execution of a (long) running task and to launch a new one, but the budget already spent for the interrupted task is lost. The main questions are which VMs to enroll, and whether and when to interrupt tasks that have been executing for some time. We assess the complexity of the problem by showing its NP-completeness and providing a 2-approximation for the asymptotic case where budget and deadline both tend to infinity. Then we introduce several heuristics and compare their performance by running an extensive set of simulations.

This work has been presented at the Cluster 2019 conference [17].

7.7. Improved energy-aware strategies for periodic real-time tasks under reliability constraints

This work revisited the real-time scheduling problem recently introduced by Haque, Aydin and Zhu [62]. In this challenging problem, task redundancy ensures a given level of reliability while incurring a significant energy cost. By carefully setting processing frequencies, allocating tasks to processors and ordering task executions, we improve on the previous state-of-the-art approach with an average gain in energy of 20%. Furthermore, we establish the first complexity results for specific instances of the problem.

This work has been accepted at the RTSS 2019 conference [18].

7.8. Multilevel algorithms for acyclic partitioning of directed acyclic graphs

We investigate the problem of partitioning the vertices of a directed acyclic graph into a given number of parts. The objective function is to minimize the number or the total weight of the edges having end points in different parts, which is also known as the edge cut. The standard load balancing constraint of having an equitable partition of the vertices among the parts should be met. Furthermore, the partition is required to be acyclic; i.e., the interpart edges between the vertices from different parts should preserve an acyclic dependency structure among the parts. In this work, we adopt the multilevel approach with coarsening, initial partitioning, and refinement phases for acyclic partitioning of directed acyclic graphs. We focus on two-way partitioning (sometimes called bisection), as this scheme can be used in a recursive way for multiway partitioning. 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. We also propose effective ways to use the standard undirected graph partitioning methods in our multilevel scheme. We perform a large set of experiments on a dataset consisting of (i) graphs coming from an application and (ii) some others corresponding to matrices from a public collection. We report significant improvements compared to the current state of the art.

This work is published in a journal [11].

7.9. A multi-dimensional Morton-ordered block storage for mode-oblivious tensor computations

Computation on tensors, treated as multidimensional arrays, revolve around generalized basic linear algebra subroutines (BLAS). We propose a novel data structure in which tensors are blocked and blocks are stored in an order determined by Morton order. This is not only proposed for efficiency reasons, but also to induce efficient performance regardless of which mode a generalized BLAS call is invoked for; we coin the term mode-oblivious to describe data structures and algorithms that induce such behavior. Experiments on one of the most bandwidth-bound generalized BLAS kernel, the tensor–vector multiplication, not only demonstrate superior performance over two state-of-the-art variants by up to 18%, but additionally show that the proposed data structure induces a 71% less sample standard deviation for tensor–vector multiplication across d modes, where d varies from 2 to 10. Finally, we show our data structure naturally expands to other tensor kernels and demonstrate up to 38% higher performance for the higher-order power method.

This work is published in a journal [13]

7.10. Effective heuristics for matchings in hypergraphs

The problem of finding a maximum cardinality matching in a d-partite, d-uniform hypergraph is an important problem in combinatorial optimization and has been theoretically analyzed. We first generalize some graph matching heuristics for this problem. We then propose a novel heuristic based on tensor scaling to extend the matching via judicious hyperedge selections. Experiments on random, synthetic and real-life hypergraphs show that this new heuristic is highly practical and superior to the others on finding a matching with large cardinality.

This work is published in the proceedings of SEA^2, where it has received the best paper award [16].

7.11. Karp-Sipser based kernels for bipartite graph matching

We consider Karp–Sipser, a well known matching heuristic in the context of data reduction for the maximum cardinality matching problem. We describe an efficient implementation as well as modifications to reduce its time complexity in worst case instances, both in theory and in practical cases. We compare experimentally against its widely used simpler variant and show cases for which the full algorithm yields better performance .

This work appears in the proceedings of ALENEX2020 [20]

7.12. Efficient and effective sparse tensor reordering

This paper formalizes the problem of reordering a sparse tensor to improve the spatial and temporal locality of operations with it, and proposes two reordering algorithms for this problem, which we call BFS-MCS and Lexi-Order. The BFS-MCS method is a Breadth First Search (BFS)-like heuristic approach based on the maximum cardinality search family; Lexi-Order is an extension of doubly lexical ordering of matrices to tensors. We show the effects of these schemes within the context of a widely used tensor computation, the Candecomp/Parafac decomposition (CPD), when storing the tensor in three previously proposed sparse tensor formats: coordinate (COO), compressed sparse fiber (CSF), and hierarchical coordinate (HiCOO). A new partition-based superblock scheduling is also proposed for HiCOO format to improve load balance. On modern multicore CPUs, we show Lexi-Order obtains up to 4.14$\times$ speedup on sequential HiCOO-Mttkrp and 11.88$\times$ speedup on its parallel counterpart. The performance of COO-and CSF-based Mttkrps also improves. Our two reordering methods are more effective than state-of-the-art approaches.

This work appears in the proceedings of ICS2019 [21].

7.13. High performance tensor–vector multiplication on shared-memory systems

Tensor–vector multiplication is one of the core components in tensor computations. We have recently investigated high performance, single core implementation of this bandwidth-bound operation. Here, we investigate its efficient, shared-memory implementations. Upon carefully analyzing the design space, we implement a number of alternatives using OpenMP and compare them experimentally. Experimental results on up to 8 socket systems show near peak performance for the proposed algorithms.

This work appears in the proceedings of PPAM2019 and is supported with a technical report [22], [36].

7.14. Matrix symmetrization and sparse direct solvers

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

This work will appear in the proceedings of CSC2020 [23].

7.15. A scalable clustering-based task scheduler for homogeneous processors using DAG partitioning

When scheduling a directed acyclic graph (DAG) of tasks on computational platforms, a good trade-off between load balance and data locality is necessary. List-based scheduling techniques are commonly used greedy approaches for this problem. The downside of list-scheduling heuristics is that they are incapable of making short-term sacrifices for the global efficiency of the schedule. In this work, we describe new list-based scheduling heuristics based on clustering for homogeneous platforms, under the realistic duplex single-port communication model. Our approach uses an acyclic partitioner for DAGs for clustering. The clustering enhances the data locality of the scheduler with a global view of the graph. Furthermore, since the partition is acyclic, we can schedule each part completely once its input tasks are ready to be executed. We present an extensive experimental evaluation showing the trade-offs between the granularity of clustering and the parallelism, and how this affects the scheduling. Furthermore, we compare our heuristics to the best state-of-the-art list-scheduling and clustering heuristics, and obtain more than three times better makespan in cases with many communications.

This work appears in the proceedings of IPDPS 2019 [25].

7.16. Improving Locality-Aware Scheduling with Acyclic Directed Graph Partitioning

We investigate efficient execution of computations, modeled as Directed Acyclic Graphs (DAGs), on a single processor with a two-level memory hierarchy, where there is a limited fast memory and a larger slower memory. Our goal is to minimize execution time by minimizing redundant data movement between fast and slow memory. We utilize a DAG partitioner that finds localized, acyclic parts of the whole computation that can fit into fast memory, and minimizes the edge cut among the parts. We propose a new scheduler that executes each part one-by-one, obeying the dependency among parts, aiming at reducing redundant data movement needed by cut-edges. Extensive experimental evaluation shows that the proposed DAG-based scheduler significantly reduces redundant data movement.

This work will appear in the proceedings of PPAM 2019 [24].

7.17. Replication Is More Efficient Than You Think

We revisit replication coupled with checkpointing for fail-stop errors. Replication enables the application to survive many fail-stop errors, thereby allowing for longer checkpointing periods. Previously published works use replication with the no-restart strategy, which never restart failed processors until the application crashes. We introduce the restart strategy where failed processors are restarted after each checkpoint. which may introduce additional overhead during checkpoints but prevents the application configuration from degrading throughout successive checkpointing periods. We show how to compute the optimal checkpointing period for this strategy, which is much larger than the one with no-restart, thereby decreasing I/O pressure. We show through simulations that using the restart strategy significantly decreases the overhead induced by replication, in terms of both total execution time and energy consumption.

This work appears in the proceedings of SC 2019 [15], [28].

7.18. Generic matrix multiplication for multi-GPU accelerated distributed-memory platforms over PaRSEC

We introduce a generic and flexible matrix-matrix multiplication algorithm $C = A \times B$ for state-of-the-art computing platforms. Typically, these platforms are distributed-memory machines whose nodes are equipped with several accelerators. To the best of our knowledge, SLATE is the only library that provides a publicly available implementation on such platforms, and it is currently limited to problem instances where the C matrix can entirely fit in the memory of the GPU accelerators. Our algorithm relies on the classical tile-based outer-product algorithm, but enhances it with several control dependencies to increase data re-use and to optimize communication flow from/to the accelerators within each node. The algorithm is written with the PaRSEC runtime system, which allows for a fast and generic implementation, while achieving close-to-peak performance.

This work appears in the proceedings of Scala 2019 [19].

7.19. Reservation strategies for stochastic jobs

We are interested in scheduling stochastic jobs on a reservation-based platform. Specifically, we consider jobs whose execution time follows a known probability distribution. The platform is reservation-based, meaning that the user has to request fixed-length time slots. The cost then depends on both (i) the request duration (pay for what you ask); and (ii) the actual execution time of the job (pay for what you use).

A reservation strategy determines a sequence of increasing-length reservations, which are paid for until one of them allows the job to successfully complete. The goal is to minimize the total expected cost of the strategy. We provide some properties of the optimal solution, which we characterize up to the length of the first reservation. We then design several heuristics based on various approaches, including a brute-force search of the first reservation length while relying on the characterization of the optimal strategy, as well as the discretization of the target continuous probability distribution together with an optimal dynamic programming algorithm for the discrete distribution.

We evaluate these heuristics using two different platform models and cost functions: The first one targets a cloud-oriented platform (e.g., Amazon AWS) using jobs that follow a large number of usual probability distributions (e.g., Uniform, Exponential, LogNormal, Weibull, Beta), and the second one is based on interpolating traces from a real neuroscience application executed on an HPC platform. An extensive set of simulation results show the effectiveness of the proposed reservation-based approaches for scheduling stochastic jobs.

This work appears in the proceedings of IPDPS 2019 [14].

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

In the context of a consortium (http://mumps-consortium.org) of users of the MUMPS library (http://mumps-solver.org), we had partnership contracts with EDF, ALTAIR, FFT-MSC Software, Michelin, LSTC, Siemens, ESI Group, Total, Safran, LBNL, Airbus, and SHELL. Following the creation of the start-up Mumps Technologies in January 2019, these contracts (scientific exchanges, support, organization of point-to-point and plenary meetings, releases in advance, ...) have been transferred to Mumps Technologies.

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. ANR

ANR Project SOLHARIS (2019-2013), 4 years. The ANR Project SOLHAR was launched in November 2019, for a duration of 48 months. It gathers five academic partners (the HiePACS, ROMA, RealOpt, STORM and TADAAM) Inria project-teams, and CNRS-IRIT) and two industrial partners (CEA/CESTA and Airbus CRT). This project aims at producing scalable methods for direct methods for the solution of sparse linear systems on large scale and heterogeneous computing platforms, based on task-based runtime systems.

The proposed research is organized along three distinct research thrusts. The first objective deals with the development of scalable linear algebra solvers on task-based runtimes. The second one focuses on the deployement of runtime systems on large-scale heterogeneous platforms. The last one is concerned with scheduling these particular applications on a heterogeneous and large-scale environment.

9.2. International Initiatives

9.2.1. Inria International Labs

9.2.1.1. 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.

9.2.2. Inria International Partners

9.2.2.1. 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.

9.2.3. 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 four 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. Anne Benoit and Mingsong Chen have lead a JORISS project on scheduling and resilience in cloud computing. Frédéric Vivien and Jing Liu (ECNU) are leading a JORISS project on resilience for real-time applications. In the context of this collaboration two students from ECNU, Li Han and Changjiang Gou, have joined Roma for their PhD.

9.3. International Research Visitors

9.3.1. Visits to International Teams

9.3.1.1. 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 since 2011. He collaborates with several ICL researchers on high-performance linear algebra and resilience methods at scale.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events: Selection

10.1.1.1. Chair of Conference Program Committees

- Anne Benoit is the chair of HPC Algorithms Track of ISC 2020.
- Bora Uçar was the chair of the Algorithms Track of HiPC 2019; he has organized a mini-symposium at ICIAM 2019 on combinatorial scientific computing (12 talks) with Aydin Buluc and Alex Pothen.
- Frédéric Vivien was Global Chair of Topic 10 (Theory and Algorithms for Parallel Computation and Networking) of Euro-Par 2019, Göttingen, Germany, August 26-30 2019.

10.1.1.2. Member of the Conference Program Committees

- Anne Benoit was a member of the program committees of **IPDPS'20**, **IPDPS'19**, **SC'19**, **ESA'19**, and **Compas'19**.
- Loris Marchal was/is a member of the program committees of **HiPC 2019**, **HPCS 2019**, **ICPP 2019** and **ICPP 2020**.
- Grégoire Pichon was a member of the program committee of **Compas'19**.

- Yves Robert was a member of the program committee of **FTXS'19**, **Scala'19**, **PMBS'19** workshops co-located with SC'19 in Denver CO, and of the new conference with proceedings inside **SiamPP'20**.
- Bora Uçar was a member of the program committee of **PPAM 2019**, **ISC**, **ICPP**, **IPDPS 2020 Workshops**.
- Frédéric Vivien was a member of the program committees of **IPDPS'20**, **PDP 2020**, and **PDP 2019**.

10.1.1.3. Reviewer

- Loris Marchal has reviewed paper for the following conferences: **SPAA 2019**, **SC 2019**, **PODC 2019**,

10.1.2. Journal

10.1.2.1. Member of the Editorial Boards

- Yves Robert is Associate Editor of JPDC (Elsevier Journal of Parallel and Distributed Computing), TOPC (ACM Trans. On Parallel Computing), IJHPCA (Int. J. High Performance Computing and Applications) and JOCS (J. Computational Science).
- Bora Uçar is a member of the editorial board of Parallel Computing, SIAM Journal on Matrix Analysis and Applications, SIAM Journal on Scientific Computing, and IEEE Transactions on Parallel and Distributed Computing.
- Frédéric Vivien is a member of the editorial board of the Journal of Parallel and Distributed Computing.

10.1.2.2. Reviewer - Reviewing Activities

- Loris Marchal has reviewed papers for the journal CCPE (Concurrency and Computation: Practice and Experience).
- Yves Robert reviewed papers for journals IEEE TC, IEEE TPDS, JPDC, ACM TOPC, IEEE Trans. Cloud Computing.
- Bora Uçar has reviewed papers for journals IEEE TPDS, ACM TOMS, SIAM SISC, Frontiers of Information Technology & Electronic Engineering.

10.1.3. Invited Talks

- Yves Robert delivered a keynote presentation at HPCS'2019 in Dublin (Int. Conf. on High Performance Computing & Simulation)

10.1.4. Leadership within the Scientific Community

- Anne Benoit is a member of the Steering Committee of IPDPS and HCW (Heterogeneity in Computing Workshop, co-located with IPDPS).
- Yves Robert is a member of the Steering Committee of IPDPS and HCW .
- Bora Uçar has participated in the steering committees of IPDPS, CSC, and HiPC; he has also served as vice-chair of IEEE Technical Committee on Parallel Processing.

10.1.5. Scientific Expertise

- Frédéric Vivien was re-elected to the scientific council of the École normale supérieure de Lyon. He is also a member of the scientific council of the IRMIA labex http://labex-irmia.u-strasbg.fr/.
- Yves Robert is an expert for the Horizon 2020 program of the European Commission and has reviewed two projects in 2019.
- Yves Robert chaired the HCERES evaluation committee for the Skoltech PhD program in Computational and Data-Intensive Science and Engineering, in Moscow, Russia.

10.1.6. Research Administration

- Frédéric Vivien is the vice-head of the LIP laboratory since September 2017.

10.2. Teaching - Supervision - Juries

- Yves Robert was a Committee member for the HDR of Bora Uçar. .

10.2.1. Teaching

Licence: Anne Benoit, Responsible of the L3 students at ENS Lyon, France

Master: Yves Robert, Responsible of Master Informatique Fondamentale, ENS Lyon, France

Licence: Anne Benoit, Algorithmique avancée, 48, L3, ENS Lyon, France

Licence: Yves Robert, Algorithmique, 48, L3, ENS Lyon, France

Licence: Yves Robert, Proabilités, 48, L3, ENS Lyon, France

Licence: Grégoire Pichon, Réseaux, 34, L3, Univ. Lyon 1, France

Licence: Loris Marchal, Compilation (practicals), 14, L3, Univ. Lyon 1, France

Licence: Loris Marchal, Operating systems (practicals), 12, L2, Univ. Lyon 1, France

Master: Anne Benoit, Parallel and Distributed Algorithms and Programs, 42, M1, ENS Lyon, France

Master : Bora Uçar, Combinatorial Scientific Computing (with Fanny Dufossé), 36, M2 Informatique Fondamentale, ENS Lyon, France.

Master : Loris Marchal, Data-Aware Algorithms, 30, M2 Informatique Fondamentale, ENS Lyon 1, France.

10.2.2. Supervision

PhD in progress: Yishu Du, "Resilience for numerical methods", started in December 2019, funding: China Scholarship Council and Inria, advisors: Yves Robert and Loris Marchal.

PhD in progress: Yiqin Gao, "Replication Algorithms for Real-time Tasks with Precedence Constraints", started in October 2018, funding: ENS Lyon, advisors: Yves Robert and Frédéric Vivien.

PhD in progress: Changjiang Gou, "Task scheduling on distributed platforms under memory and energy constraints", started in Oct. 2016, funding: China Scholarship Council supervised by Anne Benoit & Loris Marchal.

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

PhD in progress: 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: 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, advisor: Bora Uçar.

PhD in progress: Filip Pawlowski, "High performance tensor computations", started in October 2017, funding: CIFRE, advisors: Yves Robert, Bora Uçar and Albert-Jan Yzelman (Huawei).

10.2.3. Juries

- Loris Marchal is a responsible of the competitive selection of ENS Lyon students for Computer Science, and is thus a member of the jury of this competitive exam.

11. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] B. UÇAR. *Partitioning, matching, and ordering: Combinatorial scientific computing with matrices and tensors*, ENS de Lyon, September 2019, Habilitation à diriger des recherches, https://hal.inria.fr/tel-02377874

Articles in International Peer-Reviewed Journals

[2] P. R. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY. *Bridging the gap between flat and hierarchical low-rank matrix formats: the multilevel BLR format*, in "SIAM Journal on Scientific Computing", May 2019, vol. 41, n° 3, pp. A1414-A1442 [*DOI :* 10.1137/18M1182760], https://hal.archives-ouvertes.fr/hal-01774642

[3] P. R. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY. *Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures*, in "ACM Transactions on Mathematical Software", February 2019, vol. 45, n° 1, pp. 1-23 [*DOI :* 10.1145/3242094], https://hal.archives-ouvertes.fr/hal-01955766

[4] P. R. AMESTOY, J.-Y. L'EXCELLENT, G. MOREAU. *On exploiting sparsity of multiple right-hand sides in sparse direct solvers*, in "SIAM Journal on Scientific Computing", 2019, vol. 41, n° 1, pp. A269-A291 [*DOI :* 10.1137/17M1151882], https://hal.inria.fr/hal-01955659

[5] G. AUPY, A. BENOIT, B. GOGLIN, L. POTTIER, Y. ROBERT. *Co-scheduling HPC workloads on cache-partitioned CMP platforms*, in "International Journal of High Performance Computing Applications", April 2019, vol. 33, n° 6, pp. 1221-1239 [*DOI :* 10.1177/1094342019846956], https://hal.inria.fr/hal-02093172

[6] O. BEAUMONT, T. LAMBERT, L. MARCHAL, B. THOMAS. *Performance Analysis and Optimality Results for Data-Locality Aware Tasks Scheduling with Replicated Inputs*, in "Future Generation Computer Systems", October 2019, pp. 1-28 [*DOI :* 10.1016/J.FUTURE.2019.08.024], https://hal.inria.fr/hal-02275473

[7] A. BENOIT, A. CAVELAN, F. CIORBA, V. LE FÈVRE, Y. ROBERT. *Combining Checkpointing and Replication for Reliable Execution of Linear Workflows with Fail-Stop and Silent Errors*, in "International Journal of Networking and Computing", 2019, vol. 9, n° 1, pp. 2-27 [*DOI :* 10.15803/IJNC.9.1_2], https://hal.inria.fr/hal-02082369

[8] L.-C. CANON, A. K. W. CHANG, Y. ROBERT, F. VIVIEN. *Scheduling independent stochastic tasks under deadline and budget constraints*, in "International Journal of High Performance Computing Applications", June 2019, pp. 1-19 [*DOI :* 10.1177/1094342019852135], https://hal.inria.fr/hal-02291031

[9] L.-C. CANON, L. MARCHAL, B. SIMON, F. VIVIEN. *Online Scheduling of Task Graphs on Heterogeneous Platforms*, in "IEEE Transactions on Parallel and Distributed Systems", 2019, pp. 1-12, forthcoming [*DOI :* 10.1109/TPDS.2019.2942909], https://hal.inria.fr/hal-02291268

[10] L. HAN, V. LE FÈVRE, L.-C. CANON, Y. ROBERT, F. VIVIEN. *A Generic Approach to Scheduling and Checkpointing Workflows*, in "International Journal of High Performance Computing Applications", May 2019, pp. 1-19 [*DOI :* 10.1177/1094342019866891], https://hal.inria.fr/hal-02140295

[11] J. HERRMANN, Y. M. ÖZKAYA, B. UÇAR, K. KAYA, U. V. CATALYUREK. *Multilevel Algorithms for Acyclic Partitioning of Directed Acyclic Graphs*, in "SIAM Journal on Scientific Computing", July 2019, vol. 41, n° 4, pp. A2117-A2145 [*DOI :* 10.1137/18M1176865], https://hal.inria.fr/hal-02306566

[12] L. MARCHAL, B. SIMON, F. VIVIEN. *Limiting the memory footprint when dynamically scheduling DAGs on shared-memory platforms*, in "Journal of Parallel and Distributed Computing", February 2019, vol. 128, pp. 30-42 [*DOI :* 10.1016/J.JPDC.2019.01.009], https://hal.inria.fr/hal-02025521

[13] F. PAWŁOWSKI, B. UÇAR, A.-J. YZELMAN. *A multi-dimensional Morton-ordered block storage for mode-oblivious tensor computations*, in "Journal of computational science", March 2019, pp. 1-35, forthcoming [*DOI :* 10.1016/J.JOCS.2019.02.007], https://hal.inria.fr/hal-02082524

International Conferences with Proceedings

[14] G. AUPY, A. GAINARU, V. HONORÉ, P. RAGHAVAN, Y. ROBERT, H. SUN. *Reservation Strategies for Stochastic Jobs*, in "IPDPS 2019 - 33rd IEEE International Parallel and Distributed Processing Symposium", Rio de Janeiro, Brazil, IEEE, May 2019, pp. 166-175 [*DOI :* 10.1109/IPDPS.2019.00027], https://hal.inria.fr/hal-01968419

[15] A. BENOIT, T. HÉRAULT, V. LE FÈVRE, Y. ROBERT. *Replication Is More Efficient Than You Think*, in "SC 2019 - International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'19)", Denver, United States, November 2019, https://hal.inria.fr/hal-02273142

[16] *Best Paper*
F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Effective heuristics for matchings in hypergraphs*, in "SEA2 2019 - International Symposium on Experimental Algorithms - Special Event", Kalamata, Greece, Springer, 2019, pp. 248-264 [*DOI :* 10.1007/978-3-030-34029-2_17], https://hal.inria.fr/hal-02417475.

[17] Y. GAO, L.-C. CANON, Y. ROBERT, F. VIVIEN. *Scheduling independent stochastic tasks on heterogeneous cloud platforms*, in "IEEE Cluster 2019 - International Conference on Cluster Computing", Albuquerque, United States, IEEE, September 2019, pp. 1-11, https://hal.inria.fr/hal-02271675

[18] L. HAN, L.-C. CANON, J. LIU, Y. ROBERT, F. VIVIEN. *Improved energy-aware strategies for periodic real-time tasks under reliability constraints*, in "RTSS 2019 - 40th IEEE Real-Time Systems Symposium", York, United Kingdom, February 2020, https://hal.inria.fr/hal-02271704

[19] T. HERAULT, Y. ROBERT, G. BOSILCA, J. DONGARRA. *Generic Matrix Multiplication for Multi-GPU Accelerated Distributed-Memory Platforms over PaRSEC*, in "ScalA 2019 - IEEE/ACM 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems", Denver, United States, IEEE, November 2019, pp. 33-41 [*DOI :* 10.1109/SCALA49573.2019.00010], https://hal.inria.fr/hal-02436180

[20] K. KAYA, J. LANGGUTH, I. PANAGIOTAS, B. UÇAR. *Karp-Sipser based kernels for bipartite graph matching*, in "SIAM Symposium on Algorithm Engineering and Experiments (ALENEX20)", Salt Lake City, Utah, United States, January 2020, https://hal.inria.fr/hal-02350734

[21] J. LI, B. UÇAR, U. V. CATALYUREK, J. SUN, K. BARKER, R. VUDUC. *Efficient and effective sparse tensor reordering*, in "ICS 2019 - ACM International Conference on Supercomputing", Phoenix, United States, June 2019, pp. 227-237 [*DOI :* 10.1145/3330345.3330366], https://hal.inria.fr/hal-02306569

[22] F. PAWŁOWSKI, B. UÇAR, A.-J. YZELMAN. *High performance tensor-vector multiplication on shared-memory systems*, in "PPAM 2019 - 13th International Conference on Parallel Processing and Applied Mathematics", Bialystok, Poland, September 2019, pp. 1-11, https://hal.inria.fr/hal-02332496

[23] R. PORTASE, B. UÇAR. *Matrix symmetrization and sparse direct solvers*, in "SIAM Workshop on Combinatorial Scientific Computing 2020", Seattle, United States, 2020, https://hal.inria.fr/hal-02417778

[24] Y. M. ÖZKAYA, A. BENOIT, U. V. CATALYUREK. *Is Acyclic Directed Graph Partitioning Effective for Locality-Aware Scheduling?*, in "PPAM 2019 - 13th International Conference on Parallel Processing and Applied Mathematics", Bialystok, Poland, September 2019, https://hal.inria.fr/hal-02273122

[25] Y. M. ÖZKAYA, A. BENOIT, B. UÇAR, J. HERRMANN, U. V. CATALYUREK. *A scalable clustering-based task scheduler for homogeneous processors using DAG partitioning*, in "IPDPS 2019 - 33rd IEEE International Parallel & Distributed Processing Symposium", Rio de Janeiro, Brazil, IEEE, May 2019, pp. 155-165 [*DOI :* 10.1109/IPDPS.2019.00026], https://hal.inria.fr/hal-02082794

Scientific Books (or Scientific Book chapters)

[26] L. MARCHAL, É. SAULE, O. SINNEN. *Special Issue Proposal for the Parallel Computing Journal: HeteroPar 2016 and HCW 2016 Workshops*, Elsevier, April 2019, https://hal.inria.fr/hal-02423211

Research Reports

[27] A. BENOIT, C. GOU, L. MARCHAL. *Partitioning tree-shaped task graphs for distributed platforms with limited memory*, Inria Grenoble Rhône-Alpes, March 2019, n^{o} RR-9115, pp. 1-34, https://hal.inria.fr/hal-01644352

[28] A. BENOIT, T. HÉRAULT, V. LE FÈVRE, Y. ROBERT. *Replication Is More Efficient Than You Think*, Inria - Research Centre Grenoble – Rhône-Alpes, 2019, n^{o} RR-9278, https://hal.inria.fr/hal-02265925

[29] A. BENOIT, V. LE FÈVRE, P. RAGHAVAN, Y. ROBERT, H. SUN. *Design and Comparison of Resilient Scheduling Heuristics for Parallel Jobs*, Inria - Research Centre Grenoble – Rhône-Alpes, October 2019, n^{o} RR-9296, pp. 1-29, https://hal.inria.fr/hal-02317464

[30] L.-C. CANON, A. KONG WIN CHANG, Y. ROBERT, F. VIVIEN. *Scheduling independent stochastic tasks under deadline and budget constraints. Extended Version*, Inria Grenoble Rhône-Alpes, February 2019, n^{o} RR-9257, pp. 1-38, https://hal.inria.fr/hal-02025785

[31] A. GAINARU, B. GOGLIN, V. HONORÉ, G. PALLEZ, P. RAGHAVAN, Y. ROBERT, H. SUN. *Reservation and Checkpointing Strategies for Stochastic Jobs (Extended Version)*, Inria & Labri, Univ. Bordeaux ; Department of EECS, Vanderbilt University, Nashville, TN, USA ; Laboratoire LIP, ENS Lyon & University of Tennessee Knoxville, Lyon, France, October 2019, n^{o} RR-9294, https://hal.inria.fr/hal-02328013

[32] Y. GAO, L.-C. CANON, Y. ROBERT, F. VIVIEN. *Scheduling independent stochastic tasks on heterogeneous cloud platforms*, Inria - Research Centre Grenoble – Rhône-Alpes, May 2019, n^{o} RR-9275, https://hal.inria.fr/hal-02141253

[33] Y. GAO, L.-C. CANON, F. VIVIEN, Y. ROBERT. *Scheduling stochastic tasks on heterogeneous cloud platforms under budget and deadline constraints*, Inria - Research Centre Grenoble – Rhône-Alpes, February 2019, n^{o} RR-9260, pp. 1-34, https://hal.inria.fr/hal-02047434

[34] L. HAN, L.-C. CANON, J. LIU, Y. ROBERT, F. VIVIEN. *Improved energy-aware strategies for periodic real-time tasks under reliability constraints*, Inria - Research Centre Grenoble – Rhône-Alpes, February 2019, n^{o} RR-9259, pp. 1-38, https://hal.inria.fr/hal-02056520

[35] T. HÉRAULT, Y. ROBERT, G. BOSILCA, J. DONGARRA. *Generic matrix multiplication for multi-GPU accelerated distributed-memory platforms over PaRSEC*, Inria Grenoble - Rhone-Alpes, September 2019, n° RR-9289, https://hal.inria.fr/hal-02282529

[36] F. PAWŁOWSKI, B. UÇAR, A.-J. YZELMAN. *High performance tensor-vector multiplies on shared memory systems*, Inria - Research Centre Grenoble – Rhône-Alpes, May 2019, n° RR-9274, pp. 1-20, https://hal.inria.fr/hal-02123526

Other Publications

[37] A. AZAD, B. UÇAR, A. POTHEN. *Trends in Combinatorial Analysis: Complex Data, Machine Learning, and High-Performance Computing*, SIAM, September 2019, pp. 1-3, https://hal.inria.fr/hal-02304457

[38] C. MOMMESSIN, O. BEAUMONT, L.-C. CANON, L. EYRAUD-DUBOIS, G. LUCARELLI, L. MARCHAL, B. SIMON, D. TRYSTRAM. *Scheduling on Two Types of Resources: a Survey*, January 2020, https://arxiv.org/abs/1909.11365 - working paper or preprint, https://hal.inria.fr/hal-02432381

References in notes

[39] *Blue Waters Newsletter*, dec 2012

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

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

[42] *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

[43] *System Resilience at Extreme Scale: white paper*, 2008, DARPA, https://pdfs.semanticscholar.org/9fcb/154d6afce23cd9951fd7c116b86255d91b5c.pdf

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

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

[46] M. AMARIS, G. LUCARELLI, C. MOMMESSIN, D. TRYSTRAM. *Generic Algorithms for Scheduling Applications on Hybrid Multi-core Machines*, in "Euro-Par 2017: Parallel Processing", 2017, pp. 220–231

[47] 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

[48] 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

[49] 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

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

[51] 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

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

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

[54] 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

[55] 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

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

[57] 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

[58] 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

[59] 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

[60] 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

[61] 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

[62] M. A. HAQUE, H. AYDIN, D. ZHU. *On reliability management of energy-aware real-time systems through task replication*, in "IEEE Transactions on Parallel and Distributed Systems", 2017, vol. 28, n^{o} 3, pp. 813–825

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

[64] 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

[65] 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

[66] 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

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

[68] 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

[69] 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