IN PARTNERSHIP WITH:
CNRS

Ecole normale supérieure de Lyon

Université Claude Bernard (Lyon 1)

Activity Report 2018

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, HDR]
Loris Marchal [CNRS, Researcher, HDR]
Bora Uçar [CNRS, Researcher]

Faculty Members

Anne Benoit [Ecole Normale Supérieure Lyon, Associate Professor, HDR]
Louis-Claude Canon [Univ. de Franche-Comté, Associate Professor, until Aug 2018]
Yves Robert [Ecole Normale Supérieure Lyon, Professor, HDR]

Samuel Thibault [Univ. Bordeaux 1, Associate Professor, until July 2018]

Post-Doctoral Fellow

Adrien Rémy [Univ de Lyon, until Jun 2018]

PhD Students

Yiqin Gao [Univ de Lyon, from Oct 2018]
Changjiang Gou [China Scholarship Council]
Li Han [China Scholarship Council]
Aurélie Kong Win Chang [Ecole Normale Supérieure Lyon]
Valentin Le Fèvre [Ecole Normale Supérieure Lyon]
Gilles Moreau [Inria]
Ioannis Panagiotas [Inria]
Filip Pawlowski [CIFRE Huawei]
Loïc Pottier [Ecole Normale Supérieure Lyon, until Sep 2018]
Bertrand Simon [Ecole Normale Supérieure Lyon, until Aug 2018]
Issam Raïs [Inria]

Technical staff

Marie Durand [Inria, from Sep 2018]
Guillaume Joslin [Inria]
Chiara Puglisi [Inria]

Intern

Ali Al Zoobi [Inria, from Feb 2018 until Jun 2018]

Administrative Assistants

Solene Audoux [Inria, from Nov 2018]
Evelyne Blesle [Inria, from Mar 2018 until July 2018]

External Collaborators

Patrick Amestoy [INP Toulouse, external collaborator, HDR]
Alfredo Buttari [CNRS, external collaborator]

2. Overall Objectives

2.1. Overall Objectives

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

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

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

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

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

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

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

3. Research Program

3.1. Algorithms for probabilistic environments

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

3.1.1. Application resilience

An application is resilient if it can successfully produce a correct result in spite of potential faults in the underlying system. Application resilience can involve a broad range of techniques, including fault prediction, error detection, error containment, error correction, checkpointing, replication, migration, recovery, etc. Faults are quite frequent in the most powerful existing supercomputers. The Jaguar platform, which ranked third in the TOP 500 list in November 2011 [59], had an average of 2.33 faults per day during the period from August 2008 to February 2010 [84]. The mean-time between faults of a platform is inversely proportional to its number of components. Progresses will certainly be made in the coming years with respect to the reliability of individual components. However, designing and building high-reliability hardware components is far more expensive than using lower reliability top-of-the-shelf components. Furthermore, low-power components may not be available with high-reliability. Therefore, it is feared that the progresses in reliability will far from compensate the steady projected increase of the number of components in the largest supercomputers. Already, application failures have a huge computational cost. In 2008, the DARPA white paper on "System resilience at extreme scale" [58] stated that high-end systems wasted 20% of their computing capacity on application failure and recovery.

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

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

3.1.2. Scheduling strategies for applications with a probabilistic behavior

Static scheduling algorithms are algorithms where all decisions are taken before the start of the application execution. On the contrary, in non-static algorithms, decisions may depend on events that happen during the execution. Static scheduling algorithms are known to be superior to dynamic and system-oriented approaches in stable frameworks [65], [72], [73], [83], that is, when all characteristics of platforms and applications are perfectly known, known a priori, and do not evolve during the application execution. In practice, the prediction of application characteristics may be approximative or completely infeasible. For instance, the amount of computations and of communications required to solve a given problem in parallel may strongly depend on some input data that are hard to analyze (this is for instance the case when solving linear systems using full pivoting).

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

3.2. Platform-aware scheduling strategies

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

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

3.2.1. Energy-aware algorithms

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

Technically, a processor running at speed s dissipates s^α watts per unit of time with $2 \leq \alpha \leq 3$ [63], [64], [70]; 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 [85] 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 [81]. 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 [85]; (ii) *replication* consists in executing the same task on several processors simultaneously, in order to meet the reliability constraints [62]; and (iii) *checkpointing* consists in "saving" the work done at some certain instants, hence reducing the amount of work lost when a failure occurs [80].

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

3.2.2. Memory-aware algorithms

For many years, the bandwidth between memories and processors has increased more slowly than the computing power of processors, and the latency of memory accesses has been improved at an even slower pace. Therefore, in the time needed for a processor to perform a floating point operation, the amount of data transferred between the memory and the processor has been decreasing with each passing year. The risk is for an application to reach a point where the time needed to solve a problem is no longer dictated by the processor computing power but by the memory characteristics, comparable to the *memory wall* that limits CPU performance. In such a case, processors would be greatly under-utilized, and a large part of the computing

power of the platform would be wasted. Moreover, with the advent of multicore processors, the amount of memory per core has started to stagnate, if not to decrease. This is especially harmful to memory intensive applications. The problems related to the sizes and the bandwidths of memories are further exacerbated on modern computing platforms because of their deep and highly heterogeneous hierarchies. Such a hierarchy can extend from core private caches to shared memory within a CPU, to disk storage and even tape-based storage systems, like in the Blue Waters supercomputer [55]. It may also be the case that heterogeneous cores are used (such as hybrid CPU and GPU computing), and that each of them has a limited memory.

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

It is worth noting that the cost of moving data has lead to the development of so called "communication-avoiding algorithms" [76]. Our approach is orthogonal to these efforts: in communication-avoiding algorithms, the application is modified, in particular some redundant work is done, in order to get rid of some communication operations, whereas in our approach, we do not modify the application, which is provided as a task graph, but we minimize the needed memory peak only by carefully scheduling tasks.

3.3. High-performance computing and linear algebra

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

3.3.1. Direct solvers for sparse linear systems

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

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

Our research in the field is strongly related to the software package MUMPS, 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 [74], [75], [79]. 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 [67] relied on BLAS level 3 kernels (Basic Linear Algebra Subroutines) that enable to fully harness the computing power of a single CPU. Then the ScaLAPACK library [66] built upon LAPACK to provide a coarse-grain parallel version, where processors operate on large block-column panels. Inter-processor communications occur through highly tuned MPI send and receive primitives. The advent of multi-core processors has led to a major modification in these algorithms [69], [82], [77]. Each processor runs several threads in parallel to keep all cores within that processor busy. Tiled versions of the algorithms have thus been designed: dividing large block-column panels into several tiles allows for a decrease in the granularity down to a level where many smaller-size tasks are spawned. In the current panel, the diagonal tile is used to eliminate all the lower tiles in the panel. Because the factorization of the whole panel is now broken into the elimination of several tiles, the update operations can also be partitioned at the tile level, which generates many tasks to feed all cores.

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

We study solutions that are part-static part-dynamic, because such solutions have been shown to outperform purely dynamic ones [71]. 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 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 chair of 32nd IEEE IPDPS conference (IEEE International Parallel & Distributed Processing Symposium), held in Vancouver, Canada, May 21–25, 2018.
- Bora Uçar was the general chair of 32nd IEEE IPDPS conference (IEEE International Parallel & Distributed Processing Symposium), held in Vancouver, Canada, May 21–25, 2018.

5.1.1. Awards

BEST PAPER AWARD:

[29]
T. HÉRAULT, Y. ROBERT, A. BOUTEILLER, D. ARNOLD, K. B. FERREIRA, G. BOSILCA, J. DONGARRA. *Optimal Cooperative Checkpointing for Shared High-Performance Computing Platforms*, in "APDCM", Vancouver, Canada, 2018, https://hal.inria.fr/hal-01968441

6. New Software and Platforms

6.1. MUMPS

A MUltifrontal Massively Parallel Solver

KEYWORDS: High-Performance Computing - Direct solvers - Finite element modelling
FUNCTIONAL DESCRIPTION: MUMPS is a software library to solve large sparse linear systems (AX=B) on sequential and parallel distributed memory computers. It implements a sparse direct method called the multifrontal method. It is used worldwide in academic and industrial codes, in the context numerical modeling of physical phenomena with finite elements. Its main characteristics are its numerical stability, its large number of features, its high performance and its constant evolution through research and feedback from its community of users. Examples of application fields include structural mechanics, electromagnetism, geophysics, acoustics, computational fluid dynamics. MUMPS is developed by INPT(ENSEEIHT)-IRIT, Inria, CERFACS, University of Bordeaux, CNRS and ENS Lyon. In 2014, a consortium of industrial users has been created (http://mumps-consortium.org).

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. Several new features have been developed in 2017 and 2018 that are included in some MUMPS versions provided to partners for experimentation (e.g. in the context of industrial contracts). These features will appear in the future public versions, starting with MUMPS 5.2.0.

- 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/

7. New Results

7.1. Birkhoff–von Neumann decomposition

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, which supports a claim of Brualdi [68], shows that this family of heuristics can miss optimal decompositions. We also investigate the performance of two heuristics from this family theoretically. The findings are published in a journal [10].

7.2. Parallel sparse matrix-vector multiply

There are three common parallel sparse matrix-vector multiply algorithms: 1D 3 row-parallel, 1D column-parallel and 2D row-column-parallel. The 1D parallel algorithms offer the advantage of having only one communication phase. On the other hand, the 2D parallel algorithm is more scalable but it suffers from two communication phases. In this work, we introduce a novel concept of heterogeneous messages where a heterogeneous message may contain both input-vector entries and partially computed output-vector entries.

This concept not only leads to a decreased number of messages, but also enables fusing the input-and output-communication phases into a single phase. These findings are exploited to propose a 1.5D parallel sparse matrix-vector multiply algorithm which is called local row-column-parallel. This proposed algorithm requires a constrained fine-grain partitioning in which each fine-grain task is assigned to the processor that contains either its input-vector entry, or its output-vector entry, or both. We propose two methods to carry out the constrained fine-grain partitioning. We conduct our experiments on a large set of test matrices to evaluate the partitioning qualities and partitioning times of these proposed 1.5D methods. The findings are published in a journal [14].

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

We consider a variant of the well-known, NP-complete problem of minimum cut linear arrangement for directed acyclic graphs. In this variant, we are given a directed acyclic graph and we are asked to find a topological ordering such that the maximum number of cut edges at any point in this ordering is minimum. In our variant, the vertices and edges have weights, and the aim is to minimize the maximum weight of cut edges in addition to the weight of the last vertex before the cut. There is a known, polynomial time algorithm [78] for the cases where the input graph is a rooted tree. We focus on the instances where the input graph is a directed series-parallel graph, and propose a polynomial time algorithm, thus expanding the class of graphs for which a polynomial time algorithm is known. Directed acyclic graphs are used to model scientific applications where the vertices correspond to the tasks of a given application and the edges represent the dependencies between the tasks. In such models, the problem we address reads as minimizing the peak memory requirement in an execution of the application. Our work, combined with Liu's work on rooted trees addresses this practical problem in two important classes of applications. The findings are published in a journal [15].

7.4. Parallel Candecomp/Parafac decomposition of sparse tensors using dimension trees

Tensor factorization has been increasingly used to address various problems in many fields such as signal processing, data compression, computer vision, and computational data analysis. CANDECOMP/PARAFAC (CP) decomposition of sparse tensors has successfully been applied to many well-known problems in web search, graph analytics, recommender systems, health care data analytics, and many other domains. In these applications, computing the CP decomposition of sparse tensors efficiently is essential in order to be able to process and analyze data of massive scale. For this purpose, we investigate an efficient computation and parallelization of the CP decomposition for sparse tensors. We provide a novel computational scheme for reducing the cost of a core operation in computing the CP decomposition with the traditional alternating least squares (CP-ALS) based algorithm. We then effectively parallelize this computational scheme in the context of CP-ALS in shared and distributed memory environments, and propose data and task distribution models for better scalability. We implement parallel CP-ALS algorithms and compare our implementations with an efficient tensor factorization library, using tensors formed from real-world and synthetic datasets. With our algorithmic contributions and implementations, we report up to 3.95x, 3.47x, and 3.9x speedups in sequential, shared memory parallel, and distributed memory parallel executions over the state of the art, and up to 1466x overall speedup over the sequential execution using 4096 cores on an IBM BlueGene/Q supercomputer. The findings are published in a journal [13].

7.5. Approximation algorithms for maximum matchings in undirected graphs

We propose heuristics for approximating the maximum cardinality matching on undirected graphs. Our heuristics are based on the theoretical body of a certain type of random graphs, and are made practical for real-life ones. The idea is based on judiciously selecting a subgraph of a given graph and obtaining a maximum cardinality matching on this subgraph. We show that the heuristics have an approximation guarantee of around $0.866 - \log(n)/n$ for a graph with n vertices. Experiments for verifying the theoretical results in practice are provided. The findings are published in a conference proceedings [25].

7.6. SINA: A Scalable iterative network aligner

Given two graphs, network alignment asks for a potentially partial mapping between the vertices of the two graphs. This arises in many applications where data from different sources need to be integrated. Recent graph aligners use the global structure of input graphs and additional information given for the edges and vertices. We present SINA, an efficient, shared memory parallel implementation of such an aligner. Our experimental evaluations on a 32-core shared memory machine showed that SINA scales well for aligning large real-world graphs: SINA can achieve up to $28.5\times$ speedup, and can reduce the total execution time of a graph alignment problem with 2M vertices and 100M edges from 4.5 hours to under 10 minutes. To the best of our knowledge, SINA is the first parallel aligner that uses global structure and vertex and edge attributes to handle large graphs. The findings are published in a conference proceedings [34].

7.7. Acyclic partitioning of large 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 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 inter-part 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 multi-way 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 improvements, on average, around 59% compared to the current state of the art. The findings are published in a research report [50].

7.8. 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 by several researchers. In this work, we first devise heuristics for this problem by generalizing the existing cheap graph matching heuristics. Then, we 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. The findings are published in a research report [46].

7.9. Scaling matrices and counting the perfect matchings in graphs

We investigate efficient randomized methods for approximating the number of perfect matchings in bipartite graphs and general graphs. Our approach is based on assigning probabilities to edges. The findings are published in a research report [47].

7.10. 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. 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 better performance in cases with many communications. The findings are published in a research report [53].

7.11. Data-Locality Aware Dynamic Schedulers for Independent Tasks with Replicated Inputs

In this work we concentrate on a crucial parameter for efficiency in Big Data and HPC applications: data locality. We focus on the scheduling of a set of independant tasks, each depending on an input file. We assume that each of these input files has been replicated several times and placed in local storage of different nodes of a cluster, similarly of what we can find on HDFS system for example. We consider two optimization problems, related to the two natural metrics: makespan optimization (under the constraint that only local tasks are allowed) and communication optimization (under the constraint of never letting a processor idle in order to optimize makespan). For both problems we investigate the performance of dynamic schedulers, in particular the basic greedy algorithm we can for example find in the default MapReduce scheduler. First we theoretically study its performance, with probabilistic models, and provide a lower bound for communication metric and asymptotic behaviour for both metrics. Second we propose simulations based on traces from a Hadoop cluster to compare the different dynamic schedulers and assess the expected behaviour obtained with the theoretical study.

These findings have been presented at the CEBDA workshop [19].

7.12. Parallel scheduling of DAGs under memory constraints.

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 systems which dynamically allocate tasks onto the resources of increasingly complex and heterogeneous 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 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 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 impact on quality of the final parallel schedule.

This work has been presented at the IPDPS 2018 conference [31] and an extended version has been submitted to the Elsevier JPDC journal [52].

7.13. Online Scheduling of Task Graphs on Hybrid 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 [61], 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 $\Theta(\sqrt{m/k})$. 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.

This work has been presented at the EuroPar 2018 conference [24].

7.14. Memory-aware tree partitioning on homogeneous platforms

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 presented as a short paper in the PDP 2018 conference [27].

7.15. Reliability-aware energy optimization for throughput-constrained applications on MPSoC.

Multi-Processor System-on-Chip (MPSoC) has emerged as a promising platform to meet the increasing performance demand of embedded applications. However, due to limited energy budget, it is hard to guarantee that applications on MPSoC can be accomplished on time with a required throughput. The situation becomes even worse for applications with high reliability requirements, since extra energy will be inevitably consumed by task re-executions or duplicated tasks. Based on Dynamic Voltage and Frequency Scaling (DVFS) and task duplication techniques, this paper presents a novel energy-efficient scheduling model, which aims at minimizing the overall energy consumption of MPSoC applications under both throughput and reliability constraints. The problem is shown to be NP-complete, and several polynomial-time heuristics are proposed to tackle this problem. Comprehensive simulations on both synthetic and real application graphs show that our proposed heuristics can meet all the given constraints, while reducing the energy consumption.

This findings have been presented at the ICPADS 2018 conference [26].

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

Scientific workloads are often described by Directed Acyclic task Graphs. Indeed, DAGs represent both a theoretical model and the structure employed by dynamic runtime schedulers to handle HPC applications. A natural problem is then to compute a makespan-minimizing schedule of a given graph. In this paper, we are motivated by task graphs arising from multifrontal factorizations of sparse matrices and therefore work under the following practical model. Tasks are malleable (i.e., a single task can be allotted a time-varying number of processors) and their speedup behaves perfectly up to a first threshold, then speedup increases linearly, but not perfectly, up to a second threshold where the speedup levels off and remains constant.

After proving the NP-hardness of minimizing the makespan of DAGs under this model, we study several heuristics. We propose model-optimized variants for PROPSCHEDULING, widely used in linear algebra application scheduling, and FLOWFLEX. GREEDYFILLING is proposed, a novel heuristic designed for our speedup model, and we demonstrate that PROPSCHEDULING and GREEDYFILLING are 2-approximation algorithms. In the evaluation, employing synthetic data sets and task graphs arising from multifrontal factorization, the proposed optimized variants and GREEDYFILLING significantly outperform the traditional algorithms, whereby GREEDYFILLING demonstrates a particular strength for balanced graphs.

These findings have been published in the IEEE TPDS journal [16].

7.17. Performance and scalability of the block low-rank multifrontal factorization on multicore architectures

Matrices coming from elliptic Partial Differential Equations have been shown to have a low-rank property which 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 reasonably easy to use in a general purpose multifrontal solver and its potential compared to standard (full-rank) solvers has been demonstrated. Recently, new variants have been introduced and it was proved that they can further reduce the complexity but their performance remained to be analyzed. We develop a multithreaded BLR factorization, and analyze its efficiency and scalability in shared-memory multicore environments. We identify the challenges posed by the use of BLR approximations in multifrontal solvers and put forward several algorithmic variants of the BLR factorization that overcome these challenges by improving its efficiency and scalability. We illustrate the performance analysis of the BLR multifrontal factorization with numerical experiments on a large set of problems coming from a variety of real-life applications.

This work has been accepted for publication in the ACM Transactions on Mathematical Software [5].

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

The cost of the solution phase in sparse direct methods is sometimes critical. It can be larger than that 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. In this setting, vertical sparsity reduces the number of operations by avoiding computations on rows that are entirely zero, and horizontal sparsity goes further by performing each elementary solve operation only on a subset of the RHS columns. To maximize the exploitation of horizontal sparsity, we propose a new algorithm to build a permutation of the RHS columns. We then propose an original approach to split the RHS columns into a minimal number of blocks, while reducing the number of operations down to a given threshold. Both algorithms are motivated by geometric intuitions and designed using an algebraic approach, so that they can be applied to general systems. We demonstrate the effectiveness of our algorithms on systems coming from real applications and compare them to other standard approaches. We also give some perspectives and possible applications.

This work has been accepted for publication in the SIAM Journal on Scientific Computing [6].

7.19. Efficient use of sparsity by direct solvers applied to 3D controlled-source EM problems

Controlled-source electromagnetic (CSEM) surveying becomes a widespread method for oil and gas exploration, which requires fast and efficient software for inverting large-scale EM datasets. In this context, one often needs to solve sparse systems of linear equations with a *large* number of *sparse* right-hand sides, each corresponding to a given transmitter position. Sparse direct solvers are very attractive for these problems, especially when combined with low-rank approximations which significantly reduce the complexity and the cost of the factorization. In the case of thousands of right-hand sides, the time spent in the sparse triangular solve tends to dominate the total simulation time and here we propose several approaches to reduce it. A significant reduction is demonstrated for marine CSEM application by utilizing the sparsity of the right-hand sides (RHS) and of the solutions that results from the geometry of the problem. Large gains are achieved by restricting computations at the forward substitution stage to exploit the fact that the RHS matrix might have empty rows (*vertical sparsity*) and/or empty blocks of columns within a non-empty row (*horizontal sparsity*). We also adapt the parallel algorithms that were designed for the factorization to solve-oriented algorithms and describe performance optimizations particularly relevant for the very large numbers of right-hand sides of the

CSEM application. We show that both the operation count and the elapsed time for the solution phase can be significantly reduced. The total time of CSEM simulation can be divided by approximately a factor of 3 on all the matrices from our set (from 3 to 30 million unknowns, and from 4 to 12 thousands RHSs).

These findings are described in a technical report [37] and will be submitted for publication.

7.20. A Generic Approach to Scheduling and Checkpointing Workflows

We dealt with scheduling and checkpointing strategies to execute scientific workflows on failure-prone large-scale platforms. To the best of our knowledge, this work was 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 revisited classical mapping heuristics such as HEFT and MINMIN and complement them with several checkpointing strategies. The objective was 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.

This findings have been presented at the ICPP 2018 conference [28].

7.21. Scheduling independent stochastic tasks under deadline and budget constraints

We studied 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.

This findings have been presented at the SBAC-PAD 2018 conference [23].

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

- In 2018, 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 AIRBUS, FFT-MSC, and SHELL, on top of the ongoing contracts with EDF, ALTAIR, Michelin, LSTC, Siemens, ESI Group, Total, SAFRAN, LBNL,
  - 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 fourth consortium committee meeting, at SAFRAN (Saclay).

Three engineers have been funded by the membership fees in 2018, for software engineering and software development, performance study and tuning on modern architectures, business development, management of the consortium, and organization of the future of the consortium. Half a year of a PhD student was also funded by the membership fees (see Section 9.1). On top of their membership, an additional contract was finalized with Michelin to study a new functionality and understand how to best exploit MUMPS recent features in their computing environment.

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 funded 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 aimed at answering industrial needs in application domains where the cost of the solution phase of sparse direct solvers is critical. The PhD was defended on December 10, 2018 [2].

9.2. National Initiatives

9.2.1. ANR

ANR Project SOLHAR (2013-2018), 4,5 years. The ANR Project SOLHAR was launched in November 2013, for a duration of 48 months. It gathers five academic partners (the HiePACS, Cepage, ROMA and Runtime Inria project-teams, and CNRS-IRIT) and two industrial partners (CEA/CESTA and EADS-IW). This project aims at studying and designing algorithms and parallel programming models for implementing direct methods for the solution of sparse linear systems on emerging computers equipped with accelerators.

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

9.3. 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 (Vanderbilt University - Department of 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 is assessing the performance and robustness of the scheduling algorithms in applicative contexts.

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.

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. Anne Benoit and Minsong Chen are leading 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.

9.4. International Research Visitors

9.4.1. Visits to International Teams

9.4.1.1. 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.
- Anne Benoit and Bora Uçar visited the School of Computational Science and Engineering Georgia Institute of Technology, Atlanta, GA, USA. During their stay August 2017–June 2018, they worked with the research group of Prof. Umit V. Çatalyürek.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events Organisation

10.1.1.1. General Chair, Scientific Chair

- Bora Uçar was the general chair of 32nd IEEE IPDPS 2018 (IEEE International Parallel & Distributed Processing Symposium), held in Vancouver, Canada, May 21–25, 2018.

10.1.1.2. Member of the Organizing Committees

- Bora Uçar was a member of the organizing committee of ICGT 2018 (10th International Colloquium on Graph Theory and combinatorics), held in Lyon, July 9–13, 2018

10.1.2. Scientific Events Selection

10.1.2.1. Chair of Conference Program Committees

- Anne Benoit was the program chair of 32nd IEEE IPDPS 2018 (IEEE International Parallel & Distributed Processing Symposium), held in Vancouver, Canada, May 21–25, 2018. She was also the global chair for topic 3: "Scheduling and Load Balancing" of the 24th Int. European Conf. on Parallel and Distributed Computing (EuroPar 2018), held in Torino, Italy, August 27–31, 2018.

10.1.2.2. Member of the Conference Program Committees

- Bora Uçar was a member of the program committee of **IA**^{3}, 2018 The Eight Workshop on Irregular Applications: Architectures and Algorithms, in conjunction with SC'18, November 11–16, 2018, Dallas, Texas, USA; **CSC18**, The 8th SIAM Workshop on Combinatorial Scientific Computing, Bergen, Norway June 6-8, 2018; **HiPC 2018**; 25th IEEE International Conference on High Performance Computing, Data, and Analytics, Bengaluru, India, 17–20 December 2018; **SC18** Doctoral Showcase of The International Conference for High Performance Computing, Networking, Storage, and Analysis, Dallas, TX, USA; **33rd IEEE IPDPS 2019 Workshops Committee**, 33rd IEEE International Parallel and Distributed Processing Symposium, Rio de Janeiro, Brazil, May 20–24, 2019.
- Loris Marchal was a member of the program committee of **IPDPS 2018** , **ICPP 2018** and the workshop of IPDPS **APDCM 2018**.
- Jean-Yves L'Excellent was a member of the program committee of **CSC18**, The 8th SIAM Workshop on Combinatorial Scientific Computing, Bergen, Norway June 6-8, 2018.
- Frédéric Vivien was a member of the program committee of **IPDPS 2018**, **PDP 2018**; **EduPar 18**, and the Poster session of **SC18**.
- Yves Robert was a member of the program committee of the FTXS, Scala and PMBS workshops co-located with SC'18 in Dallas, TX.

10.1.2.3. Reviewer

Bora Uçar reviewed a paper for 33rd IEEE IPDPS 2019.

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

- Anne Benoit is Associate Editor (in Chief) of ParCo, the journal of Parallel Computing: Systems and Applications (from July 2018). She is also a member of the editorial board (Associate Editor) of TPDS, IEEE Transactions on Parallel and Distributed Systems since 2015, and of JPDC, the Journal of Parallel and Distributed Computing, since 2011.
- Bora Uçar is a member of the editorial board of Parallel Computing, April 2016–on going, and SIAM Journal on Matrix Analysis and Applications (SIMAX), May 2018–ongoing.
- Frédéric Vivien is Associate Editor of Parallel Computing (Elsevier) and of JPDC (Elsevier Journal of Parallel and Distributed Computing).
- Yves Robert is Associate Editor of JPDC (Elsevier Journal of Parallel and Distributed Computing) and TOPC (ACM Trans. On Parallel Computing).

10.1.3.2. Reviewer - Reviewing Activities

Bora Uçar reviewed papers for the journals SIAM Journal on Scientific Computing (4 in 2018); ACM Transactions on Mathematical Software (2 in 2018); IEEE Transactions on Parallel and Distributed Systems (1 in 2018); Future Generation Computer Systems (1 in 2018); IEEE Transactions on Signal Processing (1 in 2018); SIAM Journal on Matrix Analysis and applications (1 in 2018);

Anne Benoit, Loris Marchal, Yves Robert and Frédéric Vivien reviewed papers for the journals IEEE Transactions on Parallel and Distributed Systems and Elsevier Journal of Parallel and Distributed Computing.

10.1.4. Invited Talks

- Bora Uçar delivered an invited talk at the Scientific Computing Group's Seminar at the Emory University, Atlanta, USA, September 2017.
- Frédéric Vivien delivered the keynote presentation of the 8th IEEE Workshop PDCO, held in conjunction with IPDPS 2018, in Vancouver, Canada, on Monday May 21, 2018.
- Yves Robert delivered a keynote presentation at SBAC-PAD'2018, the 30th International Symposium on Computer. Architecture and High Performance Computing.
- Yves Robert delivered the keynote presentation at SCALA'2018, the 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, held in conjunction with SC'18

10.1.5. Leadership within the Scientific Community

- Anne Benoit is a member of the Steering Committee of HCW (Heterogeneity in Computing Workshop, co-located with IPDPS) since 2018.
- Yves Robert is a member of the Steering Committee of IPDPS and HCW . He is the liaison between the Steering and Program committees of IPDPS.
- Bora Uçar is a member of the Steering Committee of Combinatorial Scientific Computing (2014–on going); and IPDPS for the years 2017–2019. He is also a vice-chair of IEEE Technical Committee on Parallel Processing (TCPP).

10.1.6. Scientific Expertise

Yves Robert is an expert for the Horizon 2020 program of the European Commission and has reviews two projects in 2018.

10.1.7. Research Administration

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

Frédéric Vivien is the vice-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: Anne Benoit, Responsible of the L3 students at ENS Lyon, France

Licence: Yves Robert, Algorithmique, ENS Lyon, 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: Yves Robert, Scheduling at scale, 36, M2 Informatique Fondamentale, ENS Lyon, France

Master: Yves Robert, Responsible of M2 Informatique Fondamentale, ENS Lyon, France

Master : Loris Marchal, Complexity and calculability (practicals), 16, M1, Univ. Lyon 1, France.

10.2.2. Supervision

HdR: Loris Marchal, Memory and data aware scheduling, ENS Lyon, March 30, 2018.

PhD in progress: Yiqin Gao, "Replication Algorithms for Real-time Tasks with Precedence Constraints", started in October 2018, 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, 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: Gilles Moreau, On the solution phase of direct methods for sparse linear systems with multiple sparse right-hand sides, ENS Lyon, December 10, 2018, supervised by Jean-Yves L'Excellent and Patrick Amestoy.

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.

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).

PhD: Loïc Pottier, Co-scheduling for large-scale applications: memory and resilience, ENS Lyon, September 18, 2018, supervised by Anne Benoit & Yves Robert.

PhD: Issam Raïs, Discover, model and combine energy leverages for large scale energy efficient infrastructures, ENS Lyon, September 28, 2018, supervised by Laurent Lefèvre & Anne Benoit & Anne-Cécile Orgerie.

PhD: Bertrand Simon, Scheduling task graphs on modern computing platforms, ENS Lyon, July 4, 2018, supervised by Loris Marchal & Frédéric Vivien.

10.2.3. Juries

Yves Robert was a Reviewer for the HDR of Alfredo Buttari (Toulouse) and Head of the Committee for the HDR of Abdou Guermouche and Pierre Ramet (Bordeaux). At ENS Lyon, he was a Committee member for the HDR of Loris Marchal, and for the PhD of Loic Pottier.

10.3. Popularization

10.3.1. Interventions

- Frédéric Vivien took part in the committee which listened to the presentations of high-school students in the scope of a "MATh.en.JEANS" action (December 2018).
- Yves Robert gave the honorary speech for the Honoris Causa Diploma of ENS Lyon awarded to Marc Snir on November 9, 2018.

11. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] L. MARCHAL. *Memory and data aware scheduling*, École Normale Supérieure de Lyon, March 2018, Habilitation à diriger des recherches, https://hal.inria.fr/tel-01934712

[2] G. MOREAU. *On the Solution Phase of Direct Solvers for Sparse Linear Systems with Multiple Sparse Right-Hand Sides*, ENS Lyon ; Université de Lyon, December 2018, https://hal.archives-ouvertes.fr/tel-01959367

[3] L. POTTIER. *Co-scheduling for large-scale applications : memory and resilience*, Université de Lyon, September 2018, https://tel.archives-ouvertes.fr/tel-01892395

[4] B. SIMON. *Scheduling task graphs on modern computing platforms*, Université de Lyon, July 2018, https://tel.archives-ouvertes.fr/tel-01843558

Articles in International Peer-Reviewed Journals

[5] 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", 2018, https://hal.inria.fr/hal-01955766

[6] P. 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", 2018, pp. 1-19, https://hal.inria.fr/hal-01955659

[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", 2018, vol. 32, n^{o} 1, pp. 123-138, https://hal.inria.fr/hal-01968422

[8] 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

[9] H. CASANOVA, J. HERRMANN, Y. ROBERT. *Computing the expected makespan of task graphs in the presence of silent errors*, in "Parallel Computing", July 2018, vol. 75, pp. 41-60, https://hal.inria.fr/hal-01968433

[10] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Further notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices*, in "Linear Algebra and Applications", 2018, vol. 554, pp. 68–78 [*DOI :* 10.1016/J.LAA.2018.05.017], https://hal.inria.fr/hal-01586245

[11] L. HAN, L.-C. CANON, H. CASANOVA, Y. ROBERT, F. VIVIEN. *Checkpointing Workflows for Fail-Stop Errors*, in "IEEE Transactions on Computers", February 2018, vol. 67, n° 8, 16 p. [*DOI :* 10.1109/TC.2018.2801300], https://hal.inria.fr/hal-01701611

[12] O. KAYA, Y. ROBERT. *Computing Dense Tensor Decompositions with Optimal Dimension Trees*, in "Algorithmica", 2018, https://hal.inria.fr/hal-01974471

[13] O. KAYA, B. UÇAR. *Parallel Candecomp/Parafac Decomposition of Sparse Tensors Using Dimension Trees*, in "SIAM Journal on Scientific Computing", 2018, vol. 40, n° 1, pp. C99 - C130 [*DOI :* 10.1137/16M1102744], https://hal.inria.fr/hal-01397464

[14] E. KAYAASLAN, C. AYKANAT, B. UÇAR. *1.5D Parallel Sparse Matrix-Vector Multiply*, in "SIAM Journal on Scientific Computing", January 2018, vol. 40, n° 1, pp. C25 - C46 [*DOI :* 10.1137/16M1105591], https://hal.inria.fr/hal-01897555

[15] E. KAYAASLAN, T. LAMBERT, L. MARCHAL, B. UÇAR. *Scheduling series-parallel task graphs to minimize peak memory*, in "Theoretical Computer Science", January 2018, vol. 707, pp. 1-23 [*DOI :* 10.1016/J.TCS.2017.09.037], https://hal.inria.fr/hal-01891937

[16] L. MARCHAL, B. SIMON, O. SINNEN, F. VIVIEN. *Malleable task-graph scheduling with a practical speed-up model*, in "IEEE Transactions on Parallel and Distributed Systems", June 2018, vol. 29, n° 6, pp. 1357-1370 [*DOI :* 10.1109/TPDS.2018.2793886], https://hal.inria.fr/hal-01687189

International Conferences with Proceedings

[17] G. AUPY, A. BENOIT, B. GOGLIN, L. POTTIER, Y. ROBERT. *Co-scheduling HPC workloads on cache-partitioned CMP platforms*, in "IEEE Cluster 2018", Belfast, United Kingdom, Proceedings the 20th IEEE Cluster Conference, September 2018, pp. 335-345, https://hal.inria.fr/hal-01874154

[18] 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, May 2019, https://hal.inria.fr/hal-01968419

[19] O. BEAUMONT, T. LAMBERT, L. MARCHAL, B. THOMAS. *Data-Locality Aware Dynamic Schedulers for Independent Tasks with Replicated Inputs*, in "IPDPSW 2018 IEEE International Parallel and Distributed Processing Symposium Workshops", Vancouver, Canada, IEEE, May 2018, pp. 1-8 [*DOI :* 10.1109/IPDPSW.2018.00187], https://hal.inria.fr/hal-01878977

[20] A. BENOIT, A. CAVELAN, F. CIORBA, V. LE FÈVRE, Y. ROBERT. *Combining Checkpointing and Replication for Reliable Execution of Linear Workflows*, in "APDCM'18 workshop, in conjunction with IPDPS'18", Vancouver, Canada, May 2018, https://hal.inria.fr/hal-01963655

[21] A. BENOIT, S. PERARNAU, L. POTTIER, Y. ROBERT. *A performance model to execute workflows on high-bandwidth-memory architectures*, in "ICPP 2018 - 47th International Conference on Parallel Processing",

Eugene, OR, United States, ACM, August 2018, pp. 1-10 [*DOI :* 10.1145/3225058.3225110], https://hal.inria.fr/hal-01798726

[22] Y. CANIOU, E. CARON, A. KONG WIN CHANG, Y. ROBERT. *Budget-aware scheduling algorithms for scientific workflows with stochastic task weights on heterogeneous IaaS Cloud platforms*, in "IPDPSW 2018 - IEEE International Parallel and Distributed Processing Symposium Workshops", Vancouver, Canada, IEEE, May 2018, pp. 15-26 [*DOI :* 10.1109/IPDPSW.2018.00014], https://hal.inria.fr/hal-01808831

[23] L.-C. CANON, A. KONG WIN CHANG, Y. ROBERT, F. VIVIEN. *Scheduling independent stochastic tasks under deadline and budget constraints*, in "SBAC-PAD 2018 - 30th International Symposium on Computer Architecture and High Performance Computing", Lyon, France, September 2018, pp. 1-8, https://hal.inria.fr/hal-01868727

[24] L.-C. CANON, L. MARCHAL, B. SIMON, F. VIVIEN. *Online Scheduling of Task Graphs on Hybrid Platforms*, in "Euro-Par 2018 - 24th International European Conference On Parallel And Distributed Computing", Turin, Italy, August 2018, pp. 1-14, https://hal.inria.fr/hal-01828301

[25] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Approximation algorithms for maximum matchings in undirected graphs*, in "CSC 2018 - SIAM Workshop on Combinatorial Scientific Computing", Bergen, Norway, Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing, SIAM, June 2018, pp. 56-65 [*DOI :* 10.1137/1.9781611975215.6], https://hal.archives-ouvertes.fr/hal-01740403

[26] C. GOU, A. BENOIT, M. CHEN, L. MARCHAL, T. WEI. *Reliability-aware energy optimization for throughput-constrained applications on MPSoC*, in "ICPADS - 24th International Conference on Parallel and Distributed Systems", Sentosa, Singapore, IEEE, December 2018, pp. 1-10, https://hal.inria.fr/hal-01929927

[27] C. GOU, A. BENOIT, L. MARCHAL. *Memory-aware tree partitioning on homogeneous platforms*, in "PDP 2018 - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing", Cambridge, United Kingdom, March 2018, pp. 321-324 [*DOI :* 10.1109/PDP2018.2018.00056], https://hal.inria.fr/hal-01892022

[28] L. HAN, V. LE FÈVRE, L.-C. CANON, Y. ROBERT, F. VIVIEN. *A Generic Approach to Scheduling and Checkpointing Workflows*, in "ICPP 2018 - 47th International Conference on Parallel Processing", Eugene, OR, United States, ACM, August 2018, pp. 1-10 [*DOI :* 10.1145/3225058.3225145], https://hal.inria.fr/hal-01798627

[29] *Best Paper*
T. HÉRAULT, Y. ROBERT, A. BOUTEILLER, D. ARNOLD, K. B. FERREIRA, G. BOSILCA, J. DONGARRA. *Optimal Cooperative Checkpointing for Shared High-Performance Computing Platforms*, in "APDCM", Vancouver, Canada, 2018, https://hal.inria.fr/hal-01968441.

[30] V. LE FÈVRE, G. BOSILCA, A. BOUTEILLER, T. HÉRAULT, A. HORI, Y. ROBERT, J. DONGARRA. *Do Moldable Applications Perform Better on Failure-Prone HPC Platforms?*, in "Resilience - EuroPar workshop", Torino, Italy, 2018, pp. 787-799, https://hal.inria.fr/hal-01968448

[31] L. MARCHAL, H. NAGY, B. SIMON, F. VIVIEN. *Parallel scheduling of DAGs under memory constraints*, in "IPDPS 2018 - 32nd IEEE International Parallel and Distributed Processing Symposium", Vancouver, Canada, IEEE, May 2018, pp. 1-10 [*DOI :* 10.1109/IPDPS.2018.00030], https://hal.inria.fr/hal-01828312

[32] I. RAÏS, M. BOUTIGNY, L. LEFÈVRE, A.-C. ORGERIE, A. BENOIT. *Building the Table of Energy and Power Leverages for Energy Efficient Large Scale Systems*, in "HPCS: International Conference on High Performance Computing & Simulation", Orléans, France, July 2018, pp. 284-291 [*DOI :* 10.1109/HPCS.2018.00056], https://hal.archives-ouvertes.fr/hal-01845970

[33] I. RAÏS, L. LEFÈVRE, A.-C. ORGERIE, A. BENOIT. *Exploiting the Table of Energy and Power Leverages*, in "ICA3PP 2018 - 18th International Conference on Algorithms and Architectures for Parallel Processing", Guangzhou, China, November 2018, pp. 1-10, https://hal.archives-ouvertes.fr/hal-01927829

[34] A. YASAR, B. UÇAR, U. V. CATALYUREK. *SINA: A Scalable Iterative Network Aligner*, in "2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)", Barcelona, Spain, August 2018, https://hal.inria.fr/hal-01918744

Scientific Books (or Scientific Book chapters)

[35] G. AUPY, Y. ROBERT. *Scheduling for Fault-Tolerance: An Introduction*, in "Topic in parallel and distributed computing: Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms", Springer International Publishing, September 2018, pp. 143-170, https://hal.inria.fr/hal-01968454

Research Reports

[36] P. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, T. MARY. *Bridging the gap between flat and hierarchical low-rank matrix formats: the multilevel BLR format*, University of Manchester, April 2018, https://hal.archives-ouvertes.fr/hal-01774642

[37] P. R. AMESTOY, S. DE LA KETHULLE DE RYHOVE, J.-Y. L'EXCELLENT, G. MOREAU, D. V. SHANTSEV. *Efficient use of sparsity by direct solvers applied to 3D controlled-source EM problems*, Inria Grenoble Rhône-Alpes ; LIP - ENS Lyon, November 2018, n^{o} RR-9220, 26 p. , https://hal.inria.fr/hal-01912713

[38] G. AUPY, A. BENOIT, B. GOGLIN, L. POTTIER, Y. ROBERT. *Co-scheduling HPC workloads on cache-partitioned CMP platforms*, Inria, February 2018, n^{o} RR-9154, https://hal.inria.fr/hal-01719728

[39] G. AUPY, A. GAINARU, V. HONORÉ, P. RAGHAVAN, Y. ROBERT, H. SUN. *Reservation 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 2018, n^{o} RR-9211, pp. 1-38, https://hal.inria.fr/hal-01903592

[40] 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*, ROMA (Inria Rhône-Alpes / LIP Laboratoire de l'Informatique du Parallélisme) ; LIP - Laboratoire de l'Informatique du Parallélisme, December 2018, pp. 1-32, https://hal.inria.fr/hal-01955859

[41] A. BENOIT, A. CAVELAN, F. CIORBA, V. LE FÈVRE, Y. ROBERT. *Combining Checkpointing and Replication for Reliable Execution of Linear Workflows*, Inria - Research Centre Grenoble – Rhône-Alpes, February 2018, n^{o} RR-9152, pp. 1-36, https://hal.inria.fr/hal-01714978

[42] A. BENOIT, S. PERARNAU, L. POTTIER, Y. ROBERT. *A performance model to execute workflows on high-bandwidth memory architectures*, ENS Lyon ; Inria Grenoble Rhône-Alpes ; University of Tennessee

Knoxville ; Georgia Institute of Technology ; Argonne National Laboratory, April 2018, n^{o} RR-9165, pp. 1-28, https://hal.inria.fr/hal-01767888

[43] G. BOSILCA, A. BOUTEILLER, T. HÉRAULT, V. LE FÈVRE, Y. ROBERT, J. J. DONGARRA. *Distributed Termination Detection for HPC Task-Based Environments*, Inria - Research Centre Grenoble – Rhône-Alpes, June 2018, n^{o} RR-9181, pp. 1-28, https://hal.inria.fr/hal-01811823

[44] L.-C. CANON, A. KONG WIN CHANG, Y. ROBERT, F. VIVIEN. *Scheduling independent stochastic tasks deadline and budget constraints*, Inria - Research Centre Grenoble – Rhône-Alpes, June 2018, n^{o} RR-9178, pp. 1-34, https://hal.inria.fr/hal-01811885

[45] L.-C. CANON, L. MARCHAL, B. SIMON, F. VIVIEN. *Online Scheduling of Sequential Task Graphs on Hybrid Platforms*, LIP - ENS Lyon, February 2018, n^{o} RR-9150, https://hal.inria.fr/hal-01720064

[46] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Effective heuristics for matchings in hypergraphs*, Inria Grenoble Rhône-Alpes, November 2018, n^{o} RR-9224, pp. 1-18, https://hal.archives-ouvertes.fr/hal-01924180

[47] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Scaling matrices and counting the perfect matchings in graphs*, Inria Grenoble Rhône-Alpes, March 2018, n^{o} RR-9161, pp. 1-22, https://hal.inria.fr/hal-01743802

[48] C. GOU, A. BENOIT, M. CHEN, L. MARCHAL, T. WEI. *Reliability-aware energy optimization for throughput-constrained applications on MPSoC*, Laboratoire LIP, École Normale Supérieure de Lyon & CNRS & Inria, France ; Shanghai Key Lab. of Trustworthy Computing, East China Normal University, China ; Georgia Institute of Technology, USA, April 2018, n^{o} RR-9168, pp. 1-35, https://hal.inria.fr/hal-01766763

[49] L. HAN, V. LE FÈVRE, L.-C. CANON, Y. ROBERT, F. VIVIEN. *A Generic Approach to Scheduling and Checkpointing Workflows*, Inria, April 2018, n^{o} RR-9167, pp. 1-29, https://hal.inria.fr/hal-01766352

[50] J. HERRMANN, M. YUSUF ÖZKAYA, B. UÇAR, K. KAYA, U. V. CATALYUREK. *Acyclic partitioning of large directed acyclic graphs*, Inria - Research Centre Grenoble – Rhône-Alpes, March 2018, n^{o} RR-9163, https://hal.inria.fr/hal-01744603

[51] V. LE FÈVRE, G. BOSILCA, A. BOUTEILLER, T. HÉRAULT, A. HORI, Y. ROBERT, J. J. DONGARRA. *Do moldable applications perform better on failure-prone HPC platforms?*, Inria Grenoble Rhône-Alpes, May 2018, n^{o} RR-9174, pp. 1-24, https://hal.inria.fr/hal-01799498

[52] L. MARCHAL, B. SIMON, F. VIVIEN. *Limiting the memory footprint when dynamically scheduling DAGs on shared-memory platforms*, Inria Grenoble Rhône-Alpes, December 2018, n^{o} RR-9231, pp. 1-41, https://hal.inria.fr/hal-01948462

[53] M. Y. ÖZKAYA, A. BENOIT, B. UÇAR, J. HERRMANN, U. V. CATALYUREK. *A scalable clustering-based task scheduler for homogeneous processors using DAG partitioning*, Inria Grenoble Rhône-Alpes, June 2018, n^{o} RR-9185, pp. 1-30, https://hal.inria.fr/hal-01817501

References in notes

[54] *Blue Waters Newsletter*, dec 2012

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

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

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

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

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

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

[61] 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

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

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

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

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

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

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

[68] R. A. BRUALDI. *Notes on the Birkhoff algorithm for doubly stochastic matrices*, in "Canadian Mathematical Bulletin", 1982, vol. 25, n^{o} 2, pp. 191–199

[69] 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

[70] 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

[71] 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

[72] 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

[73] 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

[74] 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

[75] 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

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

[77] B. HADRI, H. LTAIEF, E. AGULLO, J. DONGARRA. *Tile QR Factorization with Parallel Panel Processing for Multicore Architectures*, in "IPDPS'10, the 24st IEEE Int. Parallel and Distributed Processing Symposium", 2010

[78] J. W. H. LIU. *An application of generalized tree pebbling to sparse matrix factorization*, in "SIAM Journal on Algebraic and Discrete Methods", 1987, vol. 8, n^{o} 3, pp. 375–395

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

[80] 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

[81] 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

[82] 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

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

[84] 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

[85] 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