

RESEARCH CENTRE

**Grenoble - Rhône-Alpes**

IN PARTNERSHIP WITH:

CNRS, Ecole normale supérieure de Lyon,  
Université Claude Bernard (Lyon 1)

2020

ACTIVITY REPORT

Project-Team

ROMA

**Optimisation des ressources : modèles,  
algorithmes et ordonnancement**

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

**DOMAIN**

**Networks, Systems and Services,  
Distributed Computing**

**THEME**

**Distributed and High Performance  
Computing**

# Contents

<b>Project-Team ROMA</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>2</b>
<b>2 Overall objectives</b>	<b>3</b>
<b>3 Research program</b>	<b>3</b>
3.1 Resilience for very large scale platforms	4
3.2 Multi-criteria scheduling strategies	4
3.3 Solvers for sparse linear algebra	4
3.4 Combinatorial scientific computing	5
<b>4 Application domains</b>	<b>5</b>
<b>5 Highlights of the year</b>	<b>6</b>
5.1 Awards	6
<b>6 New software and platforms</b>	<b>6</b>
6.1 New software	6
6.1.1 MatchMaker	6
6.1.2 PaStiX	6
<b>7 New results</b>	<b>7</b>
7.1 Resilience for very large scale platforms	7
7.2 Multi-criteria scheduling strategies	8
7.2.1 Minimizing energy consumption	8
7.2.2 Optimizing memory usage and data movement	9
7.2.3 Scheduling stochastic jobs with budget constraints	10
7.2.4 Scheduling online requests	11
7.3 Solvers for sparse linear algebra	11
7.4 Algorithms for dense linear algebra	12
7.5 Combinatorial scientific computing	13
<b>8 Partnerships and cooperations</b>	<b>14</b>
8.1 International Initiatives	14
8.1.1 Inria International Labs	14
8.1.2 Inria Associate Team not involved in an ILL	15
8.1.3 Inria International Partners	15
8.2 International Research Visitors	15
8.2.1 Visits of International Scientists	15
8.2.2 Visits to International Teams	16
8.3 European Initiatives	16
8.3.1 FP7 & H2020 Projects	16
8.3.2 Collaborations in European Programs, except FP7 and H2020	16
8.4 National Initiatives	16
8.4.1 ANR	16
<b>9 Dissemination</b>	<b>16</b>
9.1 Promoting Scientific Activities	16
9.1.1 Scientific Events: Selection	16
9.1.2 Journal	17
9.1.3 Leadership within the Scientific Community	17
9.1.4 Scientific Expertise	17
9.1.5 Research Administration	18
9.2 Teaching - Supervision - Juries	18

9.2.1 Teaching . . . . .	18
9.2.2 Supervision . . . . .	18
9.2.3 Juries . . . . .	19
9.3 Popularization . . . . .	19
9.3.1 Articles and contents . . . . .	19
<b>10 Scientific production</b>	<b>19</b>
10.1 Major publications . . . . .	19
10.2 Publications of the year . . . . .	20
10.3 Other . . . . .	23
10.4 Cited publications . . . . .	23

## Project-Team ROMA

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

### Keywords

#### Computer sciences and digital sciences

- 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
- B4.5.1. – Green computing
- B5.2.3. – Aviation
- B5.5. – Materials

## 1 Team members, visitors, external collaborators

### Research Scientists

- Frédéric Vivien [Team leader, Inria, Senior Researcher, HDR]
- Loris Marchal [CNRS, Researcher, HDR]
- Bora Uçar [CNRS, Researcher, HDR]

### Faculty Members

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

### PhD Students

- Yishu Du [Université Tongji - Chine]
- Anthony Dugois [Inria, from Oct 2020]
- Redouane Elghazi [Univ de Franche-Comté, from Oct 2020]
- Yiqin Gao [Univ de Lyon]
- Maxime Gonthier [Inria, from Oct 2020]
- Changjiang Gou [East China Normal University de Shanghai, until Sep 2020]
- Li Han [East China Normal University de Shanghai, until Aug 2020]
- Aurelie Kong Win Chang [École Normale Supérieure de Lyon, until Nov 2020]
- Valentin Le Fèvre [École Normale Supérieure de Lyon, until Aug 2020]
- Ioannis Panagiotas [Inria, until Oct 2020]
- Filip Pawlowski [Huawei]
- Lucas Perotin [École Normale Supérieure de Lyon, from Sep 2020]
- Zhiwei Wu [East China Normal University de Shanghai, from Oct 2020]

### Interns and Apprentices

- Jules Bertrand [École Normale Supérieure de Lyon, from Apr 2020 until Jul 2020]
- Redouane Elghazi [École Normale Supérieure de Lyon, from Feb 2020 until Aug 2020]
- Thibault Marette [Univ Claude Bernard, from Apr 2020 until Jul 2020]
- Lucas Perotin [École Normale Supérieure de Lyon, Jun 2020]
- Helen Xu [Inria, until May 2020]

### Administrative Assistant

- Evelyne Blesle [Inria]

## External Collaborator

- Theo Mary [CNRS, from Oct 2020]

## 2 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 top supercomputers contain more than 100,000 cores, and volunteer computing grids gather millions of processors. 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.

## 3 Research program

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

### 3.1 Resilience for very large scale platforms

For HPC applications, scale is a major opportunity. The largest supercomputers contain tens of thousands of nodes and future platforms will certainly have to enroll even more computing resources to enter the Exascale era. Unfortunately, scale is also a major threat. Indeed, even if each node provides an individual MTBF (Mean Time Between Failures) of, say, one century, a machine with 100,000 nodes will encounter a failure every 9 hours in average, which is shorter than the execution time of many HPC applications.

To further darken the picture, several types of errors need to be considered when computing at scale. In addition to classical fail-stop errors (such as hardware failures), silent errors (a.k.a silent data corruptions) must be taken into account. The cause for silent errors may be for instance soft errors in L1 cache, or bit flips due to cosmic radiations. The problem is that the detection of a silent error is not immediate, and that they only manifest later, once the corrupted data has propagated and impacted the result.

Our work investigates new models and algorithms for resilience at extreme-scale. Its main objective is to cope with both fail-stop and silent errors, and to design new approaches that dramatically improve the efficiency of state-of-the-art methods. Application resilience currently involves a broad range of techniques, including fault prediction, error detection, error containment, error correction, checkpointing, replication, migration, recovery, etc. Extending these techniques, and developing new ones, to achieve efficient execution at extreme-scale is a difficult challenge, but it is the key to a successful deployment and usage of future computing platforms.

### 3.2 Multi-criteria 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. Our work mainly considers the following two platform characteristics:

**Energy consumption.** Power management in HPC is necessary due to both monetary and environmental constraints. Using dynamic voltage and frequency scaling (DVFS) is a widely used technique to decrease energy consumption, but it can severely degrade performance and increase execution time. Part of our work in this direction studies the trade-off between energy consumption and performance (throughput or execution time). Furthermore, our work also focuses on the optimization of the power-consumption of fault-tolerant mechanisms. The problem of the energy consumption of these mechanisms is especially important because resilience generally requires redundant computations and/or redundant communications, either in time (re-execution) or in space (replication), and because redundancy consumes extra energy.

**Memory usage and data movement.** In many scientific computations, memory is a bottleneck and should be carefully considered. Besides, data movements, between main memory and secondary storages (I/Os) or between different computing nodes (communications), are taking an increasing part of the cost of computing, both in term of performance and energy consumption. In this context, our work focuses on scheduling scientific applications described as task graphs both on memory constrained platforms, and on distributed platforms with the objective of minimizing communications. The task based representation of a computing application is very common in the scheduling literature but sees an increasing interest in the HPC field thanks to the use of runtime schedulers. Our work on memory-aware scheduling is naturally multi-criteria, as it is concerned with both memory consumption, performance and data-movements.

### 3.3 Solvers for sparse linear algebra

In this theme, we work on various aspects of sparse direct solvers for linear systems. Target applications lead to sparse systems made of millions of unknowns. In the scope of the PASTIX solver, co-developed with the Inria HiePACS team, there are two main objectives: reducing as much as possible memory requirements and exploiting modern parallel architectures through the use of runtime systems.

A first research challenge is to exploit the parallelism of modern computers, made of heterogeneous (CPUs+GPUs) nodes. The approach consists of using dynamic runtime systems (in the context of the PASTIX solver, PARSEC or STARPU) to schedule tasks.

Another important direction of research is the exploitation of low-rank representations. Low-rank approximations are commonly used to compress the representation of data structures. The loss of information induced is often negligible and can be controlled. In the context of sparse direct solvers, we exploit the notion of low-rank properties in order to reduce the demand in terms of floating-point operations and memory usage. To enhance sparse direct solvers using low-rank compression, two orthogonal approaches are followed: (i) integrate new strategies for a better scalability and (ii) use preprocessing steps to better identify how to cluster unknowns, when to perform compression and which blocks not to compress.

### 3.4 Combinatorial scientific computing

CSC is a term (coined 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. Most of the time, the research output includes experiments with real life data to validate the developed combinatorial algorithms and fine tune them.

In this context, our work targets (i) the preprocessing phases of direct methods, iterative methods, and hybrid methods for solving linear systems of equations; (ii) high performance tensor computations. The core topics covering our contributions include partitioning and clustering in graphs and hypergraphs, matching in graphs, data structures and algorithms for sparse matrices and tensors (different from partitioning), and task mapping and scheduling.

## 4 Application domains

Sparse linear system 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: 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.

Tensors, or multidimensional arrays, are becoming very important because of their use in many data analysis applications. The additional dimensions over matrices (or two dimensional arrays) enable gleaning information that is otherwise unreachable. Tensors, like matrices, come in two flavors: dense tensors and sparse tensors. Dense tensors arise usually in physical and simulation applications: signal processing for electroencephalography (also named EEG, electrophysiological monitoring method to record electrical activity of the brain); hyperspectral image analysis; compression of large grid-structured data coming from a high-fidelity computational simulation; quantum chemistry etc. Dense tensors also arise in a variety of statistical and data science applications. Some of the cited applications have structured sparsity in the tensors. We see sparse tensors, with no apparent/special structure, in data analysis and network science applications. Well known applications dealing with sparse tensors are: recommender systems; computer network traffic analysis for intrusion and anomaly detection; clustering in graphs and hypergraphs modeling various relations; knowledge graphs/bases such as those in learning natural languages.



## 5 Highlights of the year

### 5.1 Awards

Yves Robert received the 2020 IEEE Computer Society Charles Babbage Award “for contributions to parallel algorithms and scheduling techniques.”

Filip Pawlowski received an innovation award at the MIT/Amazon/IEEE Graph Challenge for his paper titled "Combinatorial Tiling for Sparse Neural Networks", coauthored with Rob H. Bisseling (Utrecht University), Bora Uçar (CNRS and LIP), and Albert-Jan Yzelman (Huawei).

Anne Benoit was elected chair of the IEEE Technical Committee on Parallel Processing for two years (2020–2021).

## 6 New software and platforms

### 6.1 New software

#### 6.1.1 MatchMaker

**Name:** Maximum matchings in bipartite graphs

**Keywords:** Graph algorithmics, Matching

**Scientific Description:** The implementations of ten exact algorithms and four heuristics for solving the problem of finding a maximum cardinality matchings in bipartite graphs are provided.

**Functional Description:** This software provides algorithms to solve the maximum cardinality matching problem in bipartite graphs.

**URL:** <https://gitlab.inria.fr/bora-ucar/matchmaker>

**Publications:** [hal-00786548](#), [hal-00763920](#)

**Contact:** Bora Uçar

**Participants:** Kamer Kaya, Johannes Langguth

#### 6.1.2 PaStiX

**Name:** Parallel Sparse matriX package

**Keywords:** Linear algebra, High-performance calculation, Sparse Matrices, Linear Systems Solver, Low-Rank compression

**Scientific Description:** PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handle a combination of 1D and 2D block distributions. A dynamic scheduling can also be applied to take care of NUMA architectures while taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations.

**Functional Description:** PaStiX is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) methods. It can handle low-rank compression techniques to reduce the computation and the memory complexity. Numerical algorithms are implemented in single or double precision (real or complex) for LLt, LDLt and LU factorization with static pivoting (for non symmetric matrices having a symmetric pattern). The PaStiX library uses the graph partitioning and sparse matrix block ordering packages Scotch or Metis.

The PaStiX solver is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes with GPU accelerators or KNL processors. In particular, we provide a high-performance version with a low memory overhead for

multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

The solver also provides some low-rank compression methods to reduce the memory footprint and/or the time-to-solution.

**URL:** <https://gitlab.inria.fr/solverstack/pastix>

**Authors:** Xavier Lacoste, Pierre Ramet, Mathieu Faverge, Pascal Hénon, Tony Delarue, Esragul Korkmaz, Grégoire Pichon

**Contacts:** Pierre Ramet, Mathieu Faverge

**Participants:** Tony Delarue, Grégoire Pichon, Mathieu Faverge, Esragul Korkmaz, Pierre Ramet

**Partners:** INP Bordeaux, Université de Bordeaux

## 7 New results

### 7.1 Resilience for very large scale platforms

The ROMA team has been working on resilience problems for several years. In 2020, we have focused on several problems. First we have studied the scheduling of jobs in the presence of errors, and we dealt with two scenarios, rigid jobs and moldable jobs. We have also investigated errors in linear algebra kernels, comparing ABFT, residual checking and other methods for matrix product. Finally we have revisited the famous Young/Daly formula that provides the optimal checkpoint period for divisible-load applications, assessing its validity for stochastic workloads.

**Resilient scheduling heuristics for rigid parallel jobs** We have focused on the resilient scheduling of parallel jobs on high performance computing (HPC) platforms to minimize the overall completion time, or makespan. We have revisited the problem by assuming that jobs are subject to transient or silent errors, and hence may need to be re-executed each time they fail to complete successfully. This work generalizes the classical framework where jobs are known offline and do not fail: in this classical framework, list scheduling that gives priority to longest jobs is known to be a 3-approximation when imposing to use shelves, and a 2-approximation without this restriction. We show that when jobs can fail, using shelves can be arbitrarily bad, but unrestricted list scheduling remains a 2-approximation. We have designed several heuristics, some list-based and some shelf-based, along with different priority rules and backfilling options. We have assessed and compared their performance through an extensive set of simulations, using both synthetic jobs and log traces from the Mira supercomputer.

This work has obtained the best paper award at the APDCM'2020 conference [17], and an extended version was published in IJNC [9].

**Resilient scheduling of moldable jobs to cope with silent errors** We have then focused on the resilient scheduling of moldable parallel jobs on high-performance computing (HPC) platforms. Moldable jobs allow for choosing a processor allocation before execution, and their execution time obeys various speedup models. The objective is to minimize the overall completion time of the jobs, or the makespan, when jobs can fail due to silent errors and hence may need to be re-executed after each failure until successful completion. Our work generalizes the classical scheduling framework for failure-free jobs. To cope with silent errors, we introduce two resilient scheduling algorithms, LPA-List and Batch-List, both of which use the List strategy to schedule the jobs. Without knowing a priori how many times each job will fail, LPA-List relies on a local strategy to allocate processors to the jobs, while Batch-List schedules the jobs in batches and allows only a restricted number of failures per job in each batch. We prove new approximation ratios for the two algorithms under several prominent speedup models (e.g., roofline, communication, Amdahl, power, monotonic, and a mixed model). An extensive set of simulations is conducted to evaluate different variants of the two algorithms, and the results show that they consistently outperform some baseline heuristics. Overall, our best algorithm is within a factor of 1.6 of a lower bound on average over the entire set of experiments, and within a factor of 4.2 in the worst case.

Preliminary results with a subset of speedup models have been published in Cluster 2020 [16], and an extended version has been submitted [38].

**Detection and correction of floating-point errors in matrix-matrix multiplication** This work compares several fault-tolerance methods for the detection and correction of floating-point errors in matrix-matrix multiplication. These methods include replication, triplication, Algorithm-Based Fault Tolerance (ABFT) and residual checking (RC). Error correction for ABFT can be achieved either by recovering the corrupted entries from the correct data and the checksums by solving a small-size linear system of equations, or by recomputing corrupted coefficients. We show that both approaches can be used for RC. We provide a synthetic presentation of all methods before discussing their pros and cons. We have implemented all these methods with calls to optimized BLAS routines, and we provide performance data for a wide range of failure rates and matrix sizes. In addition, with respect to the literature, this work considers relatively high error rates.

This work has been published in [26]. The extended version is available as a research report [52].

**Robustness of the Young/Daly formula for stochastic iterative applications** The Young/Daly formula for periodic checkpointing is known to hold for a divisible load application where one can checkpoint at any time-step. In a nutshell, the optimal period is  $P_{YD} = \sqrt{2\mu_P C}$  where  $\mu_P$  is the Mean Time Between Failures (MTBF) on the platform, and  $C$  is the checkpoint time. This work assesses the accuracy of the formula for applications decomposed into computational iterations where: (i) the duration of an iteration is stochastic, i.e., obeys a probability distribution law  $\mathcal{D}$  of mean  $\mu_D$ ; and (ii) one can checkpoint only at the end of an iteration. We first consider static strategies where checkpoints are taken after a given number of iterations  $k$  and provide a closed-form, asymptotically optimal, formula for  $k$ , valid for any distribution  $\mathcal{D}$ . We then show that using the Young/Daly formula to compute  $k$  (as  $k \cdot \mu_D = P_{YD}$ ) is a first order approximation of this formula. We also consider dynamic strategies where one decides to checkpoint at the end of an iteration only if the total amount of work since the last checkpoint exceeds a threshold  $w_{th}$ , and otherwise proceed to the next iteration. Similarly, we provide a closed-form formula for this threshold and show that  $P_{YD}$  is a first-order approximation of  $w_{th}$ . Finally, we provide an extensive set of simulations where  $\mathcal{D}$  is either Uniform, Gamma or truncated Normal, which shows the global accuracy of the Young/Daly formula, even when the distribution  $\mathcal{D}$  had a large standard deviation (and when one cannot use a first-order approximation). Hence we establish that the relevance of the formula goes well beyond its original framework.

This work has been published in [19]. The extended version is available as a research report [42].

## 7.2 Multi-criteria scheduling strategies

We report here the work undertaken by the ROMA team in multi-criteria strategies, which focuses on taking into account energy and memory constraints, but also budget constraints or specific constraints for scheduling online requests.

### 7.2.1 Minimizing energy consumption

Energy is a major concern, not only in large-scale computing platform as seen above, but also for embedded and real-time systems. We have conducted several studies to reduce the energy footprint of such platforms, with the additional constraint of ensuring performance and reliability bounds.

**Improved energy-aware strategies for periodic real-time tasks under reliability constraints** This work revisited the real-time scheduling problem recently introduced by Haque, Aydin and Zhu [56]. In this challenging problem, task redundancy ensures a given level of reliability while incurring a significant energy cost. By carefully setting processing frequencies, allocating tasks to processors and ordering task executions, we improve on the previous state-of-the-art approach with an average gain in energy of 20%. Furthermore, we establish the first complexity results for specific instances of the problem.

This work has been accepted at the RTSS 2019 conference [23] which was postponed to 2020 before being cancelled!

**Energy-aware strategies for reliability-oriented real-time task allocation on heterogeneous platforms**

Low energy consumption and high reliability are widely identified as increasingly relevant issues in real-time systems on heterogeneous platforms. In this work, we proposed a multi-criteria optimization strategy to minimize the expected energy consumption while enforcing the reliability threshold and meeting all task deadlines. The tasks are replicated to ensure a prescribed reliability threshold. The platforms are composed of processors with different (and possibly unrelated) characteristics, including speed profile, energy cost, and failure rate. We provided several mapping and scheduling heuristics towards this challenging optimization problem. Specifically, a novel approach was designed to control (i) how many replicas to use for each task, (ii) on which processor to map each replica and (iii) when to schedule each replica on its assigned processor. Different mappings achieve different levels of reliability and consume different amounts of energy. Scheduling matters because once a task replica is successful, the other replicas of that task are cancelled, which calls for minimizing the amount of temporal overlap between any replica pair. Some experiments were conducted for a comprehensive set of execution scenarios, with a wide range of processor speed profiles and failure rates. The comparison results revealed that our strategies perform better than the random baseline, with a gain of 40% in energy consumption, for nearly all cases. The absolute performance of the heuristics was assessed by a comparison with a lower bound; the best heuristics achieve an excellent performance, with an average value only 4% higher than the lower bound.

This work appeared in the proceedings of the ICPP 2020 conference [24].

**Reliable and energy-aware mapping of streaming series-parallel applications onto hierarchical platforms**

Streaming applications come from various application fields such as physics, and many can be represented as a series-parallel dependence graph. We aim at minimizing the energy consumption of such applications when executed on a hierarchical platform, by proposing novel mapping strategies. Dynamic voltage and frequency scaling (DVFS) is used to reduce the energy consumption, and we ensure a reliable execution by either executing a task at maximum speed, or by triplicating it. In this work, we propose a structure rule to partition the series-parallel applications, and we prove that the optimization problem is NP-complete. We are able to derive a dynamic programming algorithm for the special case of linear chains, which provides an interesting heuristic and a building block for designing heuristics for the general case. The heuristics performance is compared to a baseline solution, where each task is executed at maximum speed. Simulations demonstrate that significant energy savings can be obtained.

This work appeared in the proceedings of the SBAC-PAD 2020 conference [22].

### 7.2.2 Optimizing memory usage and data movement

We have continued our work on exploring the tradeoffs between memory usage and performance. In particular, we studied how to partition a tree of tasks and how to dynamically schedule a DAG of tasks on memory-limited platforms.

**Partitioning tree-shaped task graphs for distributed platforms with limited memory**

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. This work considers the parallel processing of such a tree and studies how to partition it 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 (i.e., it must fit in memory), and the goal is to minimize the total resulting processing time. We prove that this 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 appeared in the IEEE TPDS journal [13].

**Revisiting dynamic DAG scheduling under memory constraints for shared-memory platforms**

This work focuses on dynamic DAG scheduling under memory constraints. We target a shared-memory

platform equipped with  $p$  parallel processors. We aim at bounding the maximum amount of memory that may be needed by any schedule using  $p$  processors to execute the DAG. We refine the classical model that computes maximum cuts by introducing two types of memory edges in the DAG, black edges for regular precedence constraints and red edges for actual memory consumption during execution. A valid edge cut cannot include more than  $p$  red edges. This limitation had never been taken into account in previous works, and dramatically changes the complexity of the problem, which was polynomial and becomes NP-hard. We introduce an Integer Linear Program (ILP) to solve it, together with an efficient heuristic based on rounding the rational solution of the ILP. In addition, we propose an exact polynomial algorithm for series-parallel graphs. We provide an extensive set of experiments, both with randomly-generated graphs and with graphs arising from practical applications, which demonstrate the impact of resource constraints on peak memory usage.

A preliminary version of this work appeared in the proceedings of the APDCM 2020 workshop conference [15] and the complete study was published in the IJNC journal [7].

### 7.2.3 Scheduling stochastic jobs with budget constraints

We have also focused on the problem of scheduling jobs whose processing time is unknown before the computation, with a budget constraint. We have studied two variants of this problem: (i) when we have to maximize the number of completed jobs before a given deadline and (ii) when such jobs must be processed on a platform with fixed-size reservation.

**Scheduling independent stochastic tasks under deadline and budget constraints** This work discusses scheduling strategies for the problem of maximizing the expected number of tasks that can be executed on a cloud platform within a given budget and under a deadline constraint. Task execution times are not known before execution; instead, the only information available to the scheduler is that they obey some probability distribution. The main questions are how many processors to enroll and whether and when to interrupt tasks that have been executing for some time.

Our previous work had focused on the study when the probability distribution is known before execution. This work deals with the (much more) difficult problem when the the probability distribution is unknown to the scheduler. Then the scheduler needs to acquire some information before deciding for a cutting threshold: instead of allowing all tasks to run until completion, one may want to interrupt long-running tasks at some point. In addition, the cutting threshold may be reevaluated as new information is acquired when the execution progresses further. This work presents several strategies to determine a good cutting threshold, and to decide when to re-evaluate it. In particular, we use the Kaplan-Meier estimator to account for tasks that are still running when making a decision. The efficiency of our strategies is assessed through an extensive set of simulations with various budget and deadline values, and ranging over 14 standard probability distributions. The results are available as a research report [43] and have been submitted for publication.

**Reservation and Checkpointing Strategies for Stochastic Jobs** In this work, we are interested in scheduling and checkpointing stochastic jobs on a reservation-based platform, whose cost depends both (i) on the reservation made, and (ii) on the actual execution time of the job. Stochastic jobs are jobs whose execution time cannot be determined easily. They arise from the heterogeneous, dynamic and data-intensive requirements of new emerging fields such as neuroscience. In this study, we assume that jobs can be interrupted at any time to take a checkpoint, and that job execution times follow a known probability distribution. Based on past experience, the user has to determine a sequence of fixed-length reservation requests, and to decide whether the state of the execution should be checkpointed at the end of each request. The objective is to minimize the expected cost of a successful execution of the jobs. We provide an optimal strategy for discrete probability distributions of job execution times, and we design fully polynomial-time approximation strategies for continuous distributions with bounded support. These strategies are then experimentally evaluated and compared to standard approaches such as periodic-length reservations and simple checkpointing strategies (either checkpoint all reservations, or none). The impact of an imprecise knowledge of checkpoint and restart costs is also assessed experimentally.

This work has been published in [20].

### 7.2.4 Scheduling online requests

We have focused on the problem of scheduling requests that arrive over time. In this setting, the classical makespan objective function is no longer relevant, and one should focus on the flow (response time) or stretch metrics.

**Max-stretch minimization on an edge-cloud platform** We have considered the problem of scheduling independent jobs that are generated by processing units at the edge of the network. These jobs can either be executed locally, or sent to a centralized cloud platform that can execute them at greater speed. Such edge-generated jobs may come from various applications, such as e-health, disaster recovery, autonomous vehicles or flying drones. The problem is to decide where and when to schedule each job, with the objective to minimize the maximum stretch incurred by any job. The stretch of a job is the ratio of the time spent by that job in the system, divided by the minimum time it could have taken if the job was alone in the system. We formalize the problem and explain the differences with other models that can be found in the literature. We prove that minimizing the max-stretch is NP-complete, even in the simpler instance with no release dates (all jobs are known in advance). This result comes from the proof that minimizing the max-stretch with homogeneous processors and without release dates is NP-complete, a complexity problem that was left open before this work. We design several algorithms to propose efficient solutions to the general problem, and we conduct simulations based on real platform parameters to evaluate the performance of these algorithms.

This work will appear in the proceedings of IPDPS 2021 [37].

**Taming tail latency in key-value stores: a scheduling perspective** Distributed key-value stores employ replication for high availability. Yet, they do not always efficiently take advantage of the availability of multiple replicas for each value, and read operations often exhibit high tail latencies. Various replica selection strategies have been proposed to address this problem, together with local request scheduling policies. It is difficult, however, to determine what is the absolute performance gain each of these strategies can achieve. We present a formal framework allowing the systematic study of request scheduling strategies in key-value stores. We contribute a definition of the optimization problem related to reducing tail latency in a replicated key-value store as a minimization problem with respect to the maximum weighted flow criterion. By using scheduling theory, we show the difficulty of this problem, and therefore the need to develop performance guarantees. We also study the behavior of heuristic methods using simulations, which highlight which properties are useful for limiting tail latency: for instance, the EFT strategy—which uses the earliest available time of servers—exhibits a tail latency that is less than half that of state-of-the-art strategies, often matching the lower bound. Our study also emphasizes the importance of metrics such as the stretch to properly evaluate replica selection and local execution policies.

A preliminary version is available in the research report [54].

## 7.3 Solvers for sparse linear algebra

We continued our work on the optimization of sparse solvers by concentrating on data locality when mapping tasks to processors, and by studying the tradeoff between memory and performance when using low-rank compression.

### Improving mapping for sparse direct solvers - a trade-off between data locality and load balancing

In order to express parallelism, parallel sparse direct solvers take advantage of the elimination tree to exhibit tree-shaped task graphs, where nodes represent computational tasks and edges represent data dependencies. One of the pre-processing stages of sparse direct solvers consists of mapping computational resources (processors) to these tasks. The objective is to minimize the factorization time by exhibiting good data locality and load balancing. The proportional mapping technique is a widely used approach to solve this resource-allocation problem. It achieves good data locality by assigning the same processors to large parts of the elimination tree. However, it may limit load balancing in some cases. In this work, we propose a dynamic mapping algorithm based on proportional mapping. This new approach, named Steal, relaxes the data locality criterion to improve load balancing. In order to validate the newly introduced

method, we perform extensive experiments on the PaStiX sparse direct solver. It demonstrates that our algorithm enables better static scheduling of the numerical factorization while keeping good data locality.

This work appeared in the proceedings of the EuroPar 2020 conference [21].

**Trading performance for memory in sparse direct solvers using low-rank compression** Sparse direct solvers using Block Low-Rank compression have been proven efficient to solve problems arising in many real-life applications. Improving those solvers is crucial for being able to 1) solve larger problems and 2) speed up computations. A main characteristic of a sparse direct solver using low-rank compression is when compression is performed. There are two distinct approaches: (1) all blocks are compressed before starting the factorization, which reduces the memory as much as possible, or (2) each block is compressed as late as possible, which usually leads to better speedup. The objective of this work is to design a composite approach, to speedup computations while staying under a given memory limit. This should allow to solve large problems that cannot be solved with Approach 2 while reducing the execution time compared to Approach 1. We propose a memory-aware strategy where each block can be compressed either at the beginning or as late as possible. We first consider the problem of choosing when to compress each block, under the assumption that all information on blocks is perfectly known, i.e., memory requirement and execution time of a block when compressed or not. We show that this problem is a variant of the NP-complete Knapsack problem, and adapt an existing 2-approximation algorithm for our problem. Unfortunately, the required information on blocks depends on numerical properties and in practice cannot be known in advance. We thus introduce models to estimate those values. Experiments on the PaStiX solver demonstrate that our new approach can achieve an excellent trade-off between memory consumption and computational cost. For instance on matrix Geo1438, Approach 2 uses three times as much memory as Approach 1 while being three times faster. Our new approach leads to an execution time only 30% larger than Approach 2 when given a memory 30% larger than the one needed by Approach 1.

A preliminary version of this work is available in the research report [53].

## 7.4 Algorithms for dense linear algebra

Closely related to sparse linear algebra, several works of the ROMA team focus on dense linear algebra. We have studied the integration of  $\mathcal{H}$ -matrix kernels for enhancing dense LU factorization. We have also proposed an implementation of block-sparse tensor contraction on top of a dynamic runtime system.

**Using H-Matrices into generic tiled algorithm on top of runtime systems** In this work, we propose an extension of the CHAMELEON library to operate with hierarchical matrices ( $\mathcal{H}$ -Matrices) and hierarchical arithmetic, producing efficient solvers for dense linear systems arising in Boundary Element Methods (BEM). Our approach builds upon an open-source  $\mathcal{H}$ -Matrices library from Airbus, named HMAT-OSS, that collects sequential numerical kernels for both hierarchical and low-rank structures; the tiled algorithms and task-parallel decompositions available in CHAMELEON for the solution of linear systems; and the STARPU runtime system to orchestrate an efficient task-parallel (multi-threaded) execution on a multicore architecture. Using an application producing matrices with features close to real industrial applications, we present shared-memory results that demonstrate a fair level of performance, close to (and sometimes better than) the one offered by a pure  $\mathcal{H}$ -matrix approach, as proposed by Airbus HMAT proprietary (and non open-source) library. Hence, this combination CHAMELEON + HMAT-OSS proposes the most efficient fully open-source software stack to solve dense compressible linear systems on shared memory architectures (distributed memory is under development).

This work appeared in the proceedings of the PDSEC 2020 workshop of IPDPS [18] and at SIAM PP'20 [30].

**Tensor operations on distributed-memory platforms with multi-GPU nodes** Many domains of scientific simulation (chemistry, condensed matter physics, data science) increasingly eschew dense tensors for block-sparse tensors, sometimes with additional structure (recursive hierarchy, rank sparsity, etc.). Distributed-memory parallel computation with block-sparse tensorial data is paramount to minimize the time-to-solution (*e.g.*, to study dynamical problems or for real-time analysis) and to accommodate

problems of realistic size that are too large to fit into the host/device memory of a single node equipped with accelerators. Unfortunately, computation with such irregular data structures is a poor match to the dominant imperative, bulk-synchronous parallel programming model. In this work, we focus on the critical element of block-sparse tensor algebra, namely binary tensor contraction, and report on an efficient and scalable implementation using the task-focused PARSEC runtime. High performance of the block-sparse tensor contraction on the Summit supercomputer is demonstrated for synthetic data as well as for real data involved in electronic structure simulations of unprecedented size.

This work is available as a research report [49] and will appear in the proceedings of IPDPS'21.

## 7.5 Combinatorial scientific computing

We worked on combinatorial problems arising in sparse matrix and tensors computations. The computations involved direct methods for solving sparse linear systems, inference in sparse neural networks, and tensor factorizations. The combinatorial problems were based on matchings on bipartite graphs, partitionings, and hyperedge queries. An earlier submission, on implementing graph and sparse matrix algorithms on a special architecture, has been published in this period.

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

This work has appeared in the proceedings of CSC2020 [29].

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

This work appears in the proceedings of ALENEX2020 [25].

**Combinatorial tiling for sparse neural networks** Sparse deep neural networks (DNNs) emerged as the result of search for networks with less storage and lower computational complexity. The sparse DNN inference is the task of using such trained DNN networks to classify a batch of input data. We propose an efficient, hybrid model- and data-parallel DNN inference using hypergraph models and partitioners. We exploit tiling and weak synchronization to increase cache reuse, hide load imbalance, and hide synchronization costs. Finally, a blocking approach allows application of this new hybrid inference procedure for deep neural networks. We initially experiment using the hybrid tiled inference approach only, using the first five layers of networks from the IEEE HPEC 2019 Graph Challenge, and attain up to 2x speedup versus a data-parallel baseline.

This work appears in the proceedings of 2020 IEEE High Performance Extreme Computing (HPEC), Sep 2020, Waltham, MA, United States, and received an innovation award at the MIT/Amazon/IEEE Graph Challenge held within HPEC [28].

**Engineering fast almost optimal algorithms for bipartite graph matching** We consider the maximum cardinality matching problem in bipartite graphs. There are a number of exact, deterministic algorithms for this purpose, whose complexities are high in practice. There are randomized approaches for special classes of bipartite graphs. Random 2-out bipartite graphs, where each vertex chooses two neighbors at random from the other side, form one class for which there is an  $O(m + n \log n)$ -time Monte Carlo algorithm. Regular bipartite graphs, where all vertices have the same degree, form another class for which there is an expected  $O(m + n \log n)$ -time Las Vegas algorithm. We investigate these two algorithms and turn them into practical heuristics with randomization. Experimental results show that the heuristics are



fast and obtain near optimal matchings. They are also more robust than the state of the art heuristics used in the cardinality matching algorithms, and are generally more useful as initialization routines.

This work appears in the proceedings of ESA 2020 - European Symposium on Algorithms, Sep 2020, Pisa, Italy [27].

**Programming strategies for irregular algorithms on the Emu Chick** The Emu Chick prototype implements migratory memory-side processing in a novel hardware system. Rather than transferring large amounts of data across the system interconnect, the Emu Chick moves lightweight thread contexts to near-memory cores before the beginning of each remote memory read. Previous work has characterized the performance of the Chick prototype in terms of memory bandwidth and programming differences from more typical, non-migratory platforms, but there has not yet been an analysis of algorithms on this system. This work evaluates irregular algorithms that could benefit from the lightweight, memory-side processing of the Chick and demonstrates techniques and optimization strategies for achieving performance in sparse matrix-vector multiply operation (SpMV), breadth-first search (BFS), and graph alignment across up to eight distributed nodes encompassing 64 nodelets in the Chick system. We also define and justify relative metrics to compare prototype FPGA-based hardware with established ASIC architectures. The Chick currently supports up to 68x scaling for graph alignment, 80 MTEPS for BFS on balanced graphs, and 50% of measured STREAM bandwidth for SpMV.

This work appears in the journal ACM Transactions on Parallel Computing [14].

**Algorithms and data structures for hyperedge queries** In this work [39], we consider the problem of querying the existence of hyperedges in hypergraphs. More formally, we are given a hypergraph, and we need to answer queries of the form “does the following set of vertices form a hyperedge in the given hypergraph?”. Our aim is to set up data structures based on hashing to answer these queries as fast as possible. We propose an adaptation of a well-known perfect hashing approach for the problem at hand. We analyze the space and run time complexity of the proposed approach, and experimentally compare it with the state of the art hashing-based solutions. Experiments demonstrate that the proposed approach has shorter query response time than the other considered alternatives, while having the shortest or the second shortest construction time.

## 8 Partnerships and cooperations

### 8.1 International Initiatives

#### 8.1.1 Inria International Labs

**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 reimaged 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 simulations 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.

### 8.1.2 Inria Associate Team not involved in an IIL

#### PEACHTREE

**Title:** PEACHTREE

**Duration:** 2020 - 2022

**Coordinator:** Bora Uçar

#### Partners:

- Translational Data Analytics (TDA) Lab lead by Ümit V. Çatalyürek, Georgia Institute of Technology, Atlanta, GA (United States)

**Inria contact:** Bora Uçar

**Summary:** Tensors, or multidimensional arrays, are becoming very important because of their use in many data analysis applications. The additional dimensions over matrices (or two dimensional arrays) enable gleaning information that is otherwise unreachable. A remarkable example comes from the Netflix Challenge. The aim of the challenge was to improve the company's algorithm for predicting user ratings on movies using a dataset containing a set of ratings of users on movies. The winning algorithm, when the challenge was concluded, had to use the time dimension on top of user x movie rating, during the analysis. Tensors from many applications, such as the mentioned one, are sparse, which means that not all entries of the tensor are relevant or known. The PeachTree project investigates the building blocks of numerical parallel tensor computation algorithms on shared memory systems, and designs a set of scheduling and combinatorial tools for achieving efficiency. Finally it proposes an efficient library containing the numerical algorithms, scheduling and combinatorial tools.

### 8.1.3 Inria International Partners

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

As for research, the JORISS program funds collaborative research projects between ENS Lyon and ECNU. Anne Benoit and Mingsong Chen have lead a JORISS project on scheduling and resilience in cloud computing. Frédéric Vivien and Jing Liu (ECNU) are leading a JORISS project on resilience for real-time applications. In the context of this collaboration two students from ECNU, Li Han and Changjiang Gou, have joined Roma for their PhD. After defending her PhD in 2020, Li Han has been hired as an associate professor at ECNU. A new student, Zhiwei Wu, has joined Roma for his PhD in October 2020.

## 8.2 International Research Visitors

### 8.2.1 Visits of International Scientists

- Helen Xu, a PhD student from MIT, visited the Roma team starting from February 2020. Because of the COVID pandemic her visit had to be cut short.

### 8.2.2 Visits to International Teams

#### 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.

## 8.3 European Initiatives

### 8.3.1 FP7 & H2020 Projects

#### 8.3.2 Collaborations in European Programs, except FP7 and H2020

**PIKS: Parallel Implementation of Karp–Sipser heuristic** Matching is a fundamental combinatorial problem that has a wide range of applications. PIKS project focuses on the data reduction rules for the cardinality matching problem proposed by Karp and Sipser and designs efficient parallel algorithms. PIKS project is funded by PHC AURORA programme. PHC AURORA is the French-Norwegian Hubert Curien Partnership. It is implemented in Norway by the Norwegian Research Council, and in France by Ministry of Europe and Foreign Affairs (Ministère de l'Europe et des Affaires étrangères) and by the Ministry Higher Education, Research and Innovation (Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation). PIKS project is carried out by Johannes Langguth from Simula Research Laboratory, Ioannis Panagiotas, from ENS de Lyon at first and then at LIP6 of the Sorbonne University, and Bora Uçar, from CNRS and LIP, ENS de Lyon.

## 8.4 National Initiatives

### 8.4.1 ANR

**ANR Project SOLHARIS (2019-2023), 4 years.** The ANR Project SOLHAR was launched in November 2019, for a duration of 48 months. It gathers five academic partners (the HiePACS, ROMA, RealOpt, STORM and TADAAM) INRIA project-teams, and CNRS-IRIT) and two industrial partners (CEA/CESTA and Airbus CRT). This project aims at producing scalable methods for direct methods for the solution of sparse linear systems on large scale and heterogeneous computing platforms, based on task-based runtime systems.

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

# 9 Dissemination

## 9.1 Promoting Scientific Activities

### 9.1.1 Scientific Events: Selection

**Chair of Conference Program Committees** Bora Uçar was the program vice chair of HiPC2020.

#### Member of the Conference Program Committees

- Anne Benoit was a member of the program committees of IPDPS'20, IPDPS'21, SC'21, SBAC-PAD'20, Compas'20, Compas'21, SuperCheck'21,
- Loris Marchal was a member of the program committee of ICPP'20.
- Grégoire Pichon was a member of the program committee of COMPAS'20.

- Yves Robert was a member of the program committees of SC'20, and of five workshops: FTXS'20, PMBS'20, SCALA'20 (colocated with SC), Resilience'20 (colocated with EuroPar), and SuperCheck'21.
- Bora Uçar was a member of the program committee of HeteroPar 18, 18th International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms (with EuroPar 2020); ICPP 49th International Conference on Parallel Processing, 17-20 August 2020, Edmonton, AB, Canada.
- Frédéric Vivien was a member of the program committees of IPDPS'20, PDP 2020, IPDPS'21, and PDP 2021.

### 9.1.2 Journal

#### Member of the Editorial Boards

- Anne Benoit is Associate Editor (in Chief) of the journal of Parallel Computing: Systems and Applications (ParCo).
- Yves Robert is a member of the editorial board of ACM Transactions on Parallel Computing (TOPC), the International Journal of High Performance Computing (IJHPCA) and the Journal of Computational Science (JOCS).
- Bora Uçar is a member of the editorial board of IEEE Transactions on Parallel and Distributed Systems (IEEE TPDS), SIAM Journal on Scientific Computing (SISC), SIAM Journal on Matrix Analysis and Applications (SIMAX), and Parallel Computing.
- Frédéric Vivien is a member of the editorial board of the Journal of Parallel and Distributed Computing.

#### Reviewer - Reviewing Activities

- Anne Benoit reviewed papers for JPDC.
- Loris Marchal made reviews for Concurrency and Computation: Practise and Experience (CCPE).
- Grégoire Pichon made reviews for Parallel Computing, SIAM SIMAX, Transactions on Parallel and Distributed Systems (TPDS).
- Yves Robert reviewed papers for IEEE TPDS, IEEE TC, TOPC and IJHPCA.
- Bora Uçar reviewed papers Concurrency and Computation: Practice and Experience, Chemometrics and Intelligent Laboratory Systems.

### 9.1.3 Leadership within the Scientific Community

- Anne Benoit is elected as chair of IEEE TCPP, the Technical Committee on Parallel Processing (2020-2021). She serves in the steering committees of IPDPS and HCW.
- Yves Robert serves in the steering committee of IPDPS, HCW and HeteroPar.
- Bora Uçar is elected as the secretary of SIAM Activity Group on Applied and Computational Discrete Algorithms (for the period Jan 21 – Dec 22).
- Bora Uçar serves in the steering committee of HiPC (2019–2021)

### 9.1.4 Scientific Expertise

- Frédéric Vivien is an elected member of the scientific council of the École normale supérieure de Lyon.
- Frédéric Vivien is a member of the scientific council of the IRMIA labex <http://labex-irmia.u-strasbg.fr/>.

### 9.1.5 Research Administration

- Frédéric Vivien was the vice-head of the LIP laboratory until December 2020.

## 9.2 Teaching - Supervision - Juries

### 9.2.1 Teaching

- Licence: Anne Benoit, Responsable of the L3 students at ENS Lyon, France
- Licence: Anne Benoit, Algorithmique avancée, 48h, L3, ENS Lyon, France
- Master: Anne Benoit, Parallel and Distributed Algorithms and Programs, 42h, M1, ENS Lyon, France
- Master: Loris Marchal, Data-Aware Algorithms, 30h, M2 Informatique Fondamentale, ENS Lyon, France.
- Master: Grégoire Pichon, Compilation / traduction des programmes, 22.5h, M1, Univ. Lyon 1, France
- Master: Grégoire Pichon, Programmation système et temps réel, 27.5h, M1, Univ. Lyon 1, France
- Master: Grégoire Pichon, Réseaux, 12h, M1, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Introduction aux réseaux et au web, 36h, L1, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Système d'exploitation, 25.5h, L2, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Programmation concurrente, 24h, L3, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Réseaux, 24h, L3, Univ. Lyon 1, France
- Master: Yves Robert, Responsable of Master Informatique Fondamentale, ENS Lyon, France
- Licence: Yves Robert, Algorithmique, 48h, L3, ENS Lyon, France
- Licence: Yves Robert, Probabilités, 48h, L3, ENS Lyon, France

### 9.2.2 Supervision

- PhD defended: Changjiang Gou, "Task scheduling on distributed platforms under memory and energy constraints", defended on September 25, 2020, funding: China Scholarship Council, supervised by Anne Benoit & Loris Marchal.
- PhD defended: Li Han, "Algorithms for detecting and correcting silent and non-functional errors in scientific workflows", defended on May 6, 2020, advisors: Yves Robert and Frédéric Vivien.
- PhD interrupted: Aurélie Kong Win Chang, "Techniques de résilience pour l'ordonnement 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. In December 2020, Aurélie decided to move to Grenoble and start a new thesis in the SPADES team.
- PhD defended: Valentin Le Fèvre, "Resilient scheduling algorithms for large-scale platforms", defended on June 18, 2020, funding: ENS Lyon, advisors: Anne Benoit and Yves Robert.
- PhD defended: Ioannis Panagiotas, "High performance algorithms for big data graph and hyper-graph problems", defended on October 9, 2020, funding: INRIA, advisor: Bora Uçar.
- PhD defended: Filip Pawlowski, "High performance tensor computations", defended on December 7, 2020, funding: CIFRE, advisors: Bora Uçar and Albert-Jan Yzelman (Huawei).
- PhD in progress: Yishu Du, "Resilience for numerical methods", started in December 2019, funding: China Scholarship Council and INRIA, advisors: Yves Robert and Loris Marchal.

- PhD in progress: Yiqin Gao, “Replication Algorithms for Real-time Tasks with Precedence Constraints”, started in October 2018, funding: ENS Lyon, advisors: Yves Robert and Frédéric Vivien.
- PhD in progress: Lucas Perotin, “Fault-tolerant scheduling of parallel jobs”, started in October 2020, funding: ENS Lyon, advisors: Anne Benoit and Yves Robert.
- PhD in progress: Zhiwei Wu, “Energy-aware strategies for periodic scientific workflows under reliability constraints on heterogeneous platforms”, started in October 2020, funding: China Scholarship Council, advisors: Frédéric Vivien, Yves Robert, Li Han (ECNU) and Jing Liu (ECNU).

### 9.2.3 Juries

- Anne Benoit was a reviewer and a member of the jury for the thesis of Valentin Honoré (October 2020, Université de Bordeaux), and for the thesis of Clément Mommessin (December 2020, Université de Grenoble).
- Loris Marchal is a responsible of the competitive selection of ENS Lyon students for Computer Science, and is thus a member of the jury of this competitive exam.
- Loris Marchal was a reviewer and member of the jury for the thesis of Massinissa Ait Aba, defended in June 2020.
- Yves Robert is a member of the 2020 ACM/IEEE-CS George Michael HPC Fellowship committee, the 2021 IEEE Fellow Committee, and the 2021 IEEE Charles Babbage Award Committee. In 2020 he will chair the ACM/IEEE-CS George Michael HPC Fellowship committee.
- Bora Uçar was a member (opponent) of the Doctorate Committee for Jan-Willem Buurlage, Leiden University, the Netherlands, July 1st, 2020. Title: Real-Time Tomographic Reconstruction, supervised by Joost Batenburg and Rob Bisseling.

## 9.3 Popularization

### 9.3.1 Articles and contents

- Anne Benoit was interviewed by Interstices in February 2020 on the subject “Quand des erreurs se produisent dans les supercalculateurs” [55].
- Yves Robert, together with George Bosilca, Aurélien Bouteiller and Thomas Herault, gave a full-day tutorial at SC’20 on *Fault-tolerant techniques for HPC and Big Data: theory and practice*.

## 10 Scientific production

### 10.1 Major publications

- [1] A. Benoit, T. Héroult, V. Le Fèvre and Y. Robert. ‘Replication Is More Efficient Than You Think’. In: *SC 2019 - International Conference for High Performance Computing, Networking, Storage, and Analysis (SC’19)*. Denver, United States, Nov. 2019. URL: <https://hal.inria.fr/hal-02273142>.
- [2] M. Bougeret, H. Casanova, M. Rabie, Y. Robert and F. Vivien. ‘Checkpointing strategies for parallel jobs.’ In: *SuperComputing (SC) - International Conference for High Performance Computing, Networking, Storage and Analysis, 2011*. United States, 2011, pp. 1–11. URL: <https://hal.archives-ouvertes.fr/hal-00738504>.
- [3] J. Dongarra, T. Héroult and Y. Robert. ‘Fault Tolerance Techniques for High-Performance Computing’. In: *Fault-Tolerance Techniques for High-Performance Computing*. Ed. by T. Héroult and Y. Robert. Springer, May 2015, p. 83. URL: <https://hal.inria.fr/hal-01200488>.
- [4] F. Dufossé and B. Uçar. ‘Notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices’. In: *Linear Algebra and its Applications* 497 (Feb. 2016), pp. 108–115. DOI: [10.1016/j.laa.2016.02.023](https://doi.org/10.1016/j.laa.2016.02.023). URL: <https://hal.inria.fr/hal-01270331>.

- [5] L. Eyraud-Dubois, L. Marchal, O. Sinnen and F. Vivien. ‘Parallel scheduling of task trees with limited memory’. In: *ACM Transactions on Parallel Computing* 2.2 (July 2015), p. 36. DOI: [10.1145/2779052](https://doi.org/10.1145/2779052). URL: <https://hal.inria.fr/hal-01160118>.
- [6] L. Marchal, B. Simon and F. Vivien. ‘Limiting the memory footprint when dynamically scheduling DAGs on shared-memory platforms’. In: *Journal of Parallel and Distributed Computing* 128 (Feb. 2019), pp. 30–42. DOI: [10.1016/j.jpdc.2019.01.009](https://doi.org/10.1016/j.jpdc.2019.01.009). URL: <https://hal.inria.fr/hal-02025521>.

## 10.2 Publications of the year

### International journals

- [7] G. Bathie, L. Marchal, Y. Robert and S. Thibault. ‘Dynamic DAG Scheduling Under Memory Constraints for Shared-Memory Platforms’. In: *International Journal of Networking and Computing* (2020), pp. 1–29. DOI: [10.15803/ijnc.11.1\\_27](https://doi.org/10.15803/ijnc.11.1_27). URL: <https://hal.inria.fr/hal-03029847>.
- [8] O. Beaumont, T. Lambert, L. Marchal and B. Thomas. ‘Performance Analysis and Optimality Results for Data-Locality Aware Tasks Scheduling with Replicated Inputs’. In: *Future Generation Computer Systems* 111 (Oct. 2020), pp. 582–598. DOI: [10.1016/j.future.2019.08.024](https://doi.org/10.1016/j.future.2019.08.024). URL: <https://hal.inria.fr/hal-02275473>.
- [9] A. Benoit, V. Le Fèvre, P. Raghavan, Y. Robert and H. Sun. ‘Resilient Scheduling Heuristics for Rigid Parallel Jobs’. In: *International Journal of Networking and Computing* (2020). URL: <https://hal.inria.fr/hal-03029874>.
- [10] Y. Caniou, E. Caron, A. Kong Win Chang and Y. Robert. ‘Budget-aware scheduling algorithms for scientific workflows with stochastic task weights on IaaS Cloud platforms \*’. In: *Concurrency and Computation: Practice and Experience* (2020). URL: <https://hal.inria.fr/hal-03029318>.
- [11] L.-C. Canon, A. K. W. Chang, Y. Robert and F. Vivien. ‘Scheduling independent stochastic tasks under deadline and budget constraints’. In: *International Journal of High Performance Computing Applications* 34.2 (1st Mar. 2020), pp. 246–264. DOI: [10.1177/1094342019852135](https://doi.org/10.1177/1094342019852135). URL: <https://hal.inria.fr/hal-02291031>.
- [12] L.-C. Canon, L. Marchal, B. Simon and F. Vivien. ‘Online Scheduling of Task Graphs on Heterogeneous Platforms’. In: *IEEE Transactions on Parallel and Distributed Systems* 31.3 (1st Mar. 2020), pp. 721–732. DOI: [10.1109/TPDS.2019.2942909](https://doi.org/10.1109/TPDS.2019.2942909). URL: <https://hal.inria.fr/hal-02291268>.
- [13] C. Gou, A. Benoit and L. Marchal. ‘Partitioning tree-shaped task graphs for distributed platforms with limited memory’. In: *IEEE Transactions on Parallel and Distributed Systems* 31.7 (3rd Mar. 2020), pp. 1533–1544. URL: <https://hal.inria.fr/hal-03024579>.
- [14] E. Hein, S. Eswar, A. Yaşar, J. Li, J. Young, T. Conte, Ü. V. Çatalyürek, R. Vuduc, J. Riedy and B. Uçar. ‘Programming Strategies for Irregular Algorithms on the Emu Chick’. In: *ACM Transactions on Parallel Computing* 7.4 (21st Oct. 2020), pp. 1–25. DOI: [10.1145/3418077](https://doi.org/10.1145/3418077). URL: <https://hal.inria.fr/hal-02991204>.

### International peer-reviewed conferences

- [15] G. Bathie, L. Marchal, Y. Robert and S. Thibault. ‘Revisiting dynamic DAG scheduling under memory constraints for shared-memory platforms’. In: IPDPS - 2020 - IEEE International Parallel and Distributed Processing Symposium Workshops. New Orleans / Virtual, United States, 18th May 2020, pp. 1–10. DOI: [10.1109/IPDPSW50202.2020.00102](https://doi.org/10.1109/IPDPSW50202.2020.00102). URL: <https://hal.inria.fr/hal-03024626>.
- [16] A. Benoit, V. Le Fèvre, L. Perotin, P. Raghavan, Y. Robert and H. Sun. ‘Resilient Scheduling of Moldable Jobs on Failure-Prone Platforms’. In: CLUSTER 2020 - IEEE International Conference on Cluster Computing. Kobe, Japan, 14th Sept. 2020, pp. 1–29. URL: <https://hal.inria.fr/hal-03028773>.

- [17] A. Benoit, V. Le Fèvre, P. Raghavan, Y. Robert and H. Sun. ‘Design and Comparison of Resilient Scheduling Heuristics for Parallel Jobs’. In: APDCM 2020 - Workshop on Advances in Parallel and Distributed Computational Models (colocated with IPDPS). New Orleans, LA, United States, 18th May 2020, pp. 1–27. URL: <https://hal.inria.fr/hal-03029842>.
- [18] R. Carratalá-Sáez, M. Faverge, G. Pichon, G. Sylvand and E. S. Quintana-Ortí. ‘Tiled Algorithms for Efficient Task-Parallel H-Matrix Solvers’. In: PDSEC 2020 - 21st IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing. News Orleans, United States, 22nd May 2020, pp. 1–10. URL: <https://hal.inria.fr/hal-02513433>.
- [19] Y. Du, L. Marchal, Y. Robert and G. Pallez. ‘Robustness of the Young/Daly formula for stochastic iterative applications’. In: ICPP 2020 - 49th International Conference on Parallel Processing. Edmonton / Virtual, Canada, 17th Aug. 2020, pp. 1–11. DOI: [10.1145/3404397.3404419](https://doi.org/10.1145/3404397.3404419). URL: <https://hal.inria.fr/hal-03024618>.
- [20] A. Gainaru, B. Goglin, V. Honoré, G. Pallez, P. Raghavan, Y. Robert and H. Sun. ‘Reservation and Checkpointing Strategies for Stochastic Jobs’. In: IPDPS 2020 - 34th IEEE International Parallel and Distributed Processing Symposium. New Orleans, LA / Virtual, United States, 18th May 2020, pp. 1–26. URL: <https://hal.inria.fr/hal-03029298>.
- [21] C. Gou, A. Al Zoobi, A. Benoit, M. Faverge, L. Marchal, G. Pichon and P. Ramet. ‘Improving mapping for sparse direct solvers: A trade-off between data locality and load balancing’. In: EuroPar 2020 - 26th International European Conference on Parallel and Distributed Computing. Warsaw / Virtual, Poland, 24th Aug. 2020, pp. 1–16. URL: <https://hal.inria.fr/hal-02973315>.
- [22] C. Gou, A. Benoit, M. Chen, L. Marchal and T. Wei. ‘Reliable and energy-aware mapping of streaming series-parallel applications onto hierarchical platforms’. In: SBAC-PAD 2020 - IEEE 32nd International Symposium on Computer Architecture and High Performance Computing. Porto, Portugal, 8th Sept. 2020, pp. 1–11. URL: <https://hal.inria.fr/hal-03024594>.
- [23] L. Han, L.-C. Canon, J. Liu, Y. Robert and F. Vivien. ‘Improved energy-aware strategies for periodic real-time tasks under reliability constraints’. In: RTSS 2019 - 40th IEEE Real-Time Systems Symposium. York, United Kingdom, 18th Feb. 2020, pp. 1–13. URL: <https://hal.inria.fr/hal-02271704>.
- [24] L. Han, Y. Gao, J. Liu, Y. Robert and F. Vivien. ‘Energy-aware strategies for reliability-oriented real-time task allocation on heterogeneous platforms’. In: ICPP 2020 - 49th International Conference on Parallel Processing. Edmonton Alberta, Canada, 17th Aug. 2020, pp. 1–11. DOI: [10.1145/3404397.3404419](https://doi.org/10.1145/3404397.3404419). URL: <https://hal.inria.fr/hal-02977879>.
- [25] K. Kaya, J. Langguth, I. Panagiotas and B. Uçar. ‘Karp-Sipser based kernels for bipartite graph matching’. In: ALENEX20 - SIAM Symposium on Algorithm Engineering and Experiments. Salt Lake City, Utah, United States, 5th Jan. 2020, pp. 1–12. URL: <https://hal.inria.fr/hal-02350734>.
- [26] V. Le Fèvre, T. Herault, J. Langou and Y. Robert. ‘A comparison of several fault-tolerance methods for the detection and correction of floating-point errors in matrix-matrix multiplication’. In: Resilience 2020 - 12th Workshop on Resiliency in High Performance Computing in Clusters, Clouds, and Grids (colocated with Euro-Par). Warsaw, Poland, Aug. 2020, pp. 1–14. URL: <https://hal.inria.fr/hal-03029309>.
- [27] I. Panagiotas and B. Uçar. ‘Engineering fast almost optimal algorithms for bipartite graph matching’. In: ESA 2020 - European Symposium on Algorithms. Pisa, Italy, Feb. 2020. URL: <https://hal.inria.fr/hal-02463717>.
- [28] F. Pawłowski, R. H. Bisseling, B. Uçar and A.-J. N. Yzelman. ‘Combinatorial Tiling for Sparse Neural Networks’. In: 2020 IEEE High Performance Extreme Computing (virtual conference). Waltham, MA, United States, Sept. 2020. URL: <https://hal.inria.fr/hal-02910997>.
- [29] R. Portase and B. Uçar. ‘Matrix symmetrization and sparse direct solvers’. In: CSC 2020 - SIAM Workshop on Combinatorial Scientific Computing. Seattle, United States, 2020, pp. 1–10. URL: <https://hal.inria.fr/hal-02417778>.



### Conferences without proceedings

- [30] R. Carratalá-Sáez, M. Faverge, G. Pichon, E. S. Quintana-Ortí and G. Sylvand. ‘Exploiting Generic Tiled Algorithms Toward Scalable H-Matrices Factorizations on Top of Runtime Systems’. In: SIAM PP20 - SIAM Conference on Parallel Processing for Scientific Computing. Seattle, United States, 12th Feb. 2020. URL: <https://hal.inria.fr/hal-03149953>.

### Doctoral dissertations and habilitation theses

- [31] C. Gou. ‘Task Mapping and Load-balancing for Performance, Memory, Reliability and Energy’. Université de Lyon; East China normal university (Shanghai), 25th Sept. 2020. URL: <https://tel.archives-ouvertes.fr/tel-03064581>.
- [32] L. Han. ‘Fault-tolerant and energy-aware algorithms for workflows and real-time systems’. Université de Lyon; East China normal university (Shanghai), 6th May 2020. URL: <https://tel.archives-ouvertes.fr/tel-02713064>.
- [33] V. Le Fèvre. ‘Resilient scheduling algorithms for large-scale platforms’. Université de Lyon, 18th June 2020. URL: <https://tel.archives-ouvertes.fr/tel-02947051>.
- [34] I. Panagiotas. ‘On matchings and related problems in graphs, hypergraphs, and doubly stochastic matrices’. Université de Lyon, 9th Oct. 2020. URL: <https://tel.archives-ouvertes.fr/tel-03011794>.
- [35] F. i. Pawlowski. ‘High-performance dense tensor and sparse matrix kernels for machine learning’. Université de Lyon, 11th Dec. 2020. URL: <https://tel.archives-ouvertes.fr/tel-03116812>.

### Reports & preprints

- [36] G. Bathie, L. Marchal, Y. Robert and S. Thibault. *Revisiting dynamic DAG scheduling under memory constraints for shared-memory platforms*. Inria, Feb. 2020. URL: <https://hal.inria.fr/hal-02488399>.
- [37] A. Benoit, R. Elghazi and Y. Robert. *Max-stretch minimization on an edge-cloud platform*. Inria - Research Centre Grenoble – Rhône-Alpes, Oct. 2020, p. 37. URL: <https://hal.inria.fr/hal-02972296>.
- [38] A. Benoit, V. Le Fèvre, L. Perotin, P. Raghavan, Y. Robert and H. Sun. *Resilient Scheduling of Moldable Parallel Jobs to Cope with Silent Errors*. Inria - Research Centre Grenoble – Rhône-Alpes, Jan. 2021. URL: <https://hal.inria.fr/hal-02614215>.
- [39] J. Bertrand, F. Dufossé and B. Uçar. *Algorithms and data structures for hyperedge queries*. Inria Grenoble Rhône-Alpes, 1st Feb. 2021, p. 21. URL: <https://hal.inria.fr/hal-03127673>.
- [40] R. Carratalá-Sáez, M. Faverge, G. Pichon, G. Sylvand and E. S. Quintana-Ortí. *Tiled Algorithms for Efficient Task-Parallel H-Matrix Solvers*. Inria, Feb. 2020. URL: <https://hal.inria.fr/hal-02489269>.
- [41] Y. Du, L. Marchal, G. Pallez and Y. Robert. *Optimal Checkpointing Strategies for Iterative Applications*. Inria - Research Centre Grenoble – Rhône-Alpes, Oct. 2020. URL: <https://hal.inria.fr/hal-02980455>.
- [42] Y. Du, L. Marchal, G. Pallez and Y. Robert. *Robustness of the Young/Daly formula for stochastic iterative applications*. Inria Grenoble Rhône-Alpes, Mar. 2020. URL: <https://hal.inria.fr/hal-02514107>.
- [43] Y. Gao, Y. Robert and F. Vivien. *Resource-Constrained Scheduling of Stochastic Tasks With Unknown Probability Distribution*. Inria - Research Centre Grenoble – Rhône-Alpes, Nov. 2020. URL: <https://hal.inria.fr/hal-02989801>.
- [44] M. Gonthier, L. Marchal and S. Thibault. *Locality-Aware Scheduling of Independant Tasks for Runtime Systems*. Inria, 2021. URL: <https://hal.inria.fr/hal-03144290>.

- [45] C. Gou, A. Al Zoobi, A. Benoit, M. Faverge, L. Marchal, G. Pichon and P. Ramet. *Improving mapping for sparse direct solvers: A trade-off between data locality and load balancing*. Inria Rhône-Alpes, Feb. 2020, p. 21. URL: <https://hal.inria.fr/hal-02491495>.
- [46] C. Gou, A. Benoit, M. Chen, L. Marchal and T. Wei. *Reliable and energy-aware mapping of streaming series-parallel applications onto hierarchical platforms*. INRIA, June 2020. URL: <https://hal.inria.fr/hal-02859980>.
- [47] L. Han, Y. Gao, J. Liu, Y. Robert and F. Vivien. *Energy-aware strategies for reliability-oriented real-time task allocation on heterogeneous platforms*. Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP, 5th Mar. 2020. URL: <https://hal.inria.fr/hal-02500381>.
- [48] T. Herault, Y. Robert, G. Bosilca, R. Harrison, C. A. Lewis and E. F. Valeev. *Distributed-memory multi-GPU block-sparse tensor contraction for electronic structure*. Inria - Research Centre Grenoble – Rhône-Alpes, June 2020. URL: <https://hal.inria.fr/hal-02872813>.
- [49] T. Herault, Y. Robert, G. Bosilca, R. J. Harrison, C. A. Lewis, E. F. Valeev and J. J. Dongarra. *Distributed-memory multi-GPU block-sparse tensor contraction for electronic structure (revised version)*. Inria - Research Centre Grenoble – Rhône-Alpes, Oct. 2020, p. 34. URL: <https://hal.inria.fr/hal-02970659>.
- [50] A. Kong Win Chang, Y. Caniou, E. Caron and Y. Robert. *Budget-aware workflow scheduling with DIET*. Inria Grenoble Rhône-Alpes, Dec. 2020. URL: <https://hal.inria.fr/hal-03080468>.
- [51] E. Korkmaz, M. Faverge, G. Pichon and P. Ramet. *Deciding Non-Compressible Blocks in Sparse Direct Solvers using Incomplete Factorization*. Inria Bordeaux - Sud Ouest, 2021, p. 16. URL: <https://hal.inria.fr/hal-03152932>.
- [52] V. Le Fèvre, T. Herault, J. Langou and Y. Robert. *A comparison of several fault-tolerance methods for the detection and correction of floating-point errors in matrix-matrix multiplication*. Inria - Research Centre Grenoble – Rhône-Alpes, June 2020. URL: <https://hal.inria.fr/hal-02867859>.
- [53] L. Marchal, T. Marette, G. Pichon and F. Vivien. *Trading Performance for Memory in Sparse Direct Solvers using Low-rank Compression*. INRIA, 23rd Oct. 2020. URL: <https://hal.inria.fr/hal-02976233>.
- [54] S. B. Mokhtar, L.-C. Canon, A. Dugois, L. Marchal and E. Rivière. *Taming Tail Latency in Key-Value Stores: a Scheduling Perspective (extended version)*. 17th Feb. 2021. URL: <https://hal.inria.fr/hal-03144818>.

### 10.3 Other

#### Scientific popularization

- [55] A. Benoit and J. Jongwane. ‘Quand des erreurs se produisent dans les supercalculateurs’. In: *Interstices* (10th Feb. 2020). URL: <https://hal.inria.fr/hal-02490956>.

### 10.4 Cited publications

- [56] M. A. Haque, H. Aydin and D. Zhu. ‘On reliability management of energy-aware real-time systems through task replication’. In: *IEEE Transactions on Parallel and Distributed Systems* 28.3 (2017), pp. 813–825.