RESEARCH CENTRE

Inria Lyon Center

IN PARTNERSHIP WITH:

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

2022

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

Inria

Contents

Project-Team ROMA

Creation of the 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

- Loris Marchal [Team leader, CNRS, Researcher, HDR]
- Suraj Kumar [INRIA, Researcher, from Oct 2022]
- Bora Uçar [CNRS, Senior Researcher, HDR]
- Frédéric Vivien [INRIA, Senior Researcher, HDR]

Faculty Members

- Anne Benoît [ENS DE LYON, Associate Professor, HDR]
- Grégoire Pichon [UNIV LYON I, Associate Professor]
- Yves Robert [ENS DE LYON, Professor, HDR]

Post-Doctoral Fellow

- Somesh Singh [INRIA]

PhD Students

- Brian Bantsoukissa [INRIA, from Oct 2022]
- Yishu Du [UNIV TONGJI & INRIA]
- Anthony Dugois [INRIA]
- Redouane Elghazi [UNIV FRANCHE-COMTE]
- Maxime Gonthier [INRIA]
- Lucas Perotin [ENS DE LYON]
- Zhiwei Wu [ECNU SHANGAI, until Aug 2022]

Interns and Apprentices

- Brian Bantsoukissa [ENS DE LYON, from Feb 2022 until Jul 2022]

Administrative Assistants

- Evelyne Blesle [INRIA, until Nov 2022]
- Chrystelle Mouton [INRIA, from Oct 2022]

External Collaborators

- Theo Mary [CNRS]
- Hongyang Sun [UNIV KANSAS]

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 three 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 meets 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 Sparse direct solvers and sparsity in computing

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.

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

- Suraj Kumar has joined the team as Inria permanent researcher.
- The SPARTACLUS ANR JCJC project lead by Grégoire Pichon has been accepted and funded (see Section 8).
- Brian Bantsoukissa has been granted a PhD funding by Inria Lyon (Moyens Incitatifs).
- The team has organized two international workshops (see Section 9.1.1).

- A new associate team with the University of Tennessee, Knoxville has been started (see Section 8).
- Julien Langou has been awarded an Inria Research Chair in the ROMA team, which will effectively start in 2023.

5.1 Awards

- Anne Benoit, Lucas Perotin, Yves Robert and Hongyang Sun have received the best paper award for their paper "Online Scheduling of Moldable Task Graphs under Common Speedup Models" at the ICPP 2022 conference [18].

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 matching 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 handles 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 a 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

Contact: Pierre Ramet

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 2022, we have focused on several problems.

7.1.1 Checkpointing à la Young/Daly: An Overview.

Participants: Anne Benoit, Yishu Du, Thomas Herault *(University of Tennessee, Knoxville)*, Loris Marchal, Guillaume Pallez *(Inria Bordeaux)*, Lucas Perotin, Yves Robert, Hongyang Sun *(University of Kansas)*, Frédéric Vivien.

The Young/Daly formula provides an approximation of the optimal checkpoint period for a parallel application executing on a supercomputing platform. The Young/Daly formula was originally designed for preemptible tightly-coupled applications. In this article we provide some background and survey various application scenarios to assess the usefulness and limitations of the formula.

This work has been invited for publication at IC3 2022 [17].

7.1.2 CheckpointingWorkflows à la Young/Daly Is Not Good Enough.

Participants: Anne Benoit, Lucas Perotin, Yves Robert, Hongyang Sun *(University of Kansas)*.

We have revisited checkpointing strategies when workflows composed of multiple tasks execute on a parallel platform. The objective is to minimize the expectation of the total execution time. For a single task, the Young/Daly formula provides the optimal checkpointing period. However, when many tasks execute simultaneously, the risk that one of them is severely delayed increases with the number of tasks. To mitigate this risk, a possibility is to checkpoint each task more often than with the Young/Daly strategy. But is it worth slowing each task down with extra checkpoints? Does the extra checkpointing make a difference globally? We have been answering these questions. On the theoretical side, we prove several negative results for keeping the Young/Daly period when many tasks execute concurrently, and we design novel checkpointing strategies that guarantee an efficient execution with high probability. On the practical side, we report comprehensive experiments that demonstrate the need to go beyond the Young/Daly period and to checkpoint more often for a wide range of application/platform settings.

This work has been published in ACM Transactions on Parallel Computing [8].

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 List and shelf schedules for independent parallel tasks to minimize the energy consumption with discrete or continuous speeds.

Participants: Anne Benoit, Louis-Claude Canon *(Univ. Besançon)*, Redouane Elghazi, Pierre-Cyrille Héam *(Univ. Besançon)*.

Scheduling independent tasks on a parallel platform is a widely-studied problem, in particular when the goal is to minimize the total execution time, or makespan ($P||C_{max}$ problem in Graham's notations). Also, many applications do not consist of sequential tasks, but rather parallel tasks, either rigid, with a fixed degree of parallelism, or moldable, with a variable degree of parallelism (i.e., for which we can decide at the execution on how many processors they are executed). Furthermore, since the energy consumption of data centers is a growing concern, both from an environmental and economical point of view, minimizing the energy consumption of a schedule is a main challenge to be addressed. One can then decide, for each task, on how many processors it is executed, and at which speed the processors are operated, with the goal to minimize the total energy consumption. We further focus on co-schedules, where tasks are partitioned into shelves, and we prove that the problem of minimizing the energy consumption remains NP-complete when static energy is consumed during the whole duration of the application. We are however able to provide an optimal algorithm for the schedule within one shelf, i.e., for a set of tasks that start at the same time. Several approximation results are derived, both with discrete and continuous speed models, and extensive simulations are performed to show the performance of the proposed algorithms.

This work appeared in the Journal of Parallel and Distributed Computing [7].

7.2.2 Dynamic Scheduling Strategies for Firm Semi-Periodic Real-Time Tasks.

Participants: Yiqin Gao, Yves Robert, Frédéric Vivien, Guillaume Pallez *(Inria Bordeaux)*.

This work introduces and assesses novel strategies to schedule firm semi-periodic real-time tasks. Jobs are released periodically and have the same relative deadline. Job execution times obey an arbitrary probability distribution and can take either bounded or unbounded values. We investigate several optimization criteria, the most prominent being the Deadline Miss Ratio (DMR). All previous work uses some admission policies but never interrupt the execution of an admitted job before its deadline. On the contrary, we introduce three new control parameters to dynamically decide whether to interrupt a job at any given time. We derive a Markov model and use its stationary distribution to determine the best value of each control parameter. Finally we conduct an extensive simulation campaign with 16 different probability distributions. The results nicely demonstrate how the new strategies help improve system performance compared with traditional approaches. In particular, we show that (i) compared to pre-execution admission rules, the control parameters make significantly better decisions; (ii) specifically, the key control parameter is to upper bound the waiting time of each job; (iii) the best scheduling strategy decreases the DMR by up to 0.35 over traditional competitors.

This work has been published in IEEE Transactions on Computers [13].

7.2.3 Minimizing I/Os in Out-of-Core Task Tree Scheduling.

Participants: Loris Marchal, Samuel McCauley *(Williams College)*, Bertrand Simon *(IN2P3 Computing Center / CNRS)*, Frédéric Vivien.

Scientific applications are usually described using directed acyclic graphs, where nodes represent tasks and edges represent dependencies between tasks. For some applications, this graph is a tree: each task produces a single result used solely by its parent. The temporary results of each task have to be stored between their production and their use.

In this work we focus on the case when the data manipulated are very large. Then, during an execution, all data may not fit together in memory. In such a case, some data have to be temporarily written to disk and evicted from memory. These data are later read from disk when they are needed for computation.

These Input/Output operations are very expensive; hence, our goal is to minimize their total volume. The order in which the tasks are processed considerably influences the amount of such Input/Output operations. Finding the schedule which minimizes this amount is an open problem that we revisit in this work.

We first formalize and generalize known results, and prove that existing solutions can be arbitrarily worse than the optimal. We then present an Integer Linear Program to solve it optimally. Finally, we propose a novel heuristic algorithm. We demonstrate its good performance through simulations on both synthetic and realistic trees built from actual scientific applications.

This work has been published in International Journal of Foundations of Computer Science [16].

7.2.4 Mapping Tree-shaped Workflows on Memory-heterogeneous Architectures.

Participants: Svetlana Kulagina *(Humboldt University of Berlin)*, Henning Meyerhenke *(Humboldt University of Berlin)*, Anne Benoit.

Directed acyclic graphs are commonly used to model scientific workflows, by expressing dependencies between tasks, as well as the resource requirements of the workflow. As a special case, rooted directed trees occur in several applications. Since typical workflows are modeled by huge trees, it is crucial to schedule them efficiently. We investigate the partitioning and mapping of tree-shaped workflows on target architectures where each processor can have a different memory size. Our three-step heuristic adapts and extends previous work for homogeneous clusters. In particular, we design a novel algorithm to assign subtrees to processors with different memory sizes, and we show how to select appropriate processors when splitting or merging subtrees. The experiments demonstrate that exploiting the heterogeneity reduces the makespan significantly compared to the state of the art for homogeneous memories.

This work has been published in the HeteroPar workshop, in conjunction with EuroPar [21].

7.2.5 Online Scheduling of Moldable Task Graphs under Common Speedup Models.

Participants: Anne Benoit, Lucas Perotin, Yves Robert, Hongyang Sun *(University of Kansas)*.

The problem of scheduling moldable tasks has been widely studied, in particular when tasks have dependencies (i.e., task graphs), or when tasks are released on-the-fly (i.e., online). However, few study has focused on both (i.e., online scheduling of moldable task graphs). We have derived constant competitive ratios for this problem under several common yet realistic speedup models for the tasks (roofline, communication, Amdahl, and a combination of them). We also provided the first lower bound on the competitive ratio of any deterministic online algorithm for arbitrary speedup model, which is not constant but depends on the number of tasks in the longest path of the graph.

This work has been published at ICPP [18], and was selected as best paper by the conference.

7.2.6 Mapping series-parallel streaming applications on hierarchical platforms with reliability and energy constraints.

Participants: Changjiang Gou, Anne Benoit, Mingsong Chen *(ECNU)*, Loris Marchal, Tongquan Wei *(ECNU)*.

Streaming applications come from various application fields such as physics, where data is continuously generated and must be processed on the fly. Typical streaming applications have a series-parallel dependence graph, and they are processed on a hierarchical failure-prone platform, as for instance in miniaturized satellites. The goal is to minimize the energy consumed when processing each data set, while ensuring real-time constraints in terms of processing time. 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, so that the time to execute a data set without failure is bounded. We propose a structure rule to partition the series-parallel applications and map the application onto the platform, and we prove that the optimization problem is NP-complete. We design a dynamic-programming algorithm for the special case of linear chains, which is optimal for a special class of schedules. Furthermore, this algorithm provides an interesting heuristic and a building block for designing heuristics for the general case. The heuristics are compared to a baseline solution, where each task is executed at maximum speed. Simulations on realistic settings demonstrate the good performance of the proposed heuristics; in particular, significant energy savings can be obtained.

This work has been published in JPDC [14].

7.2.7 Bounding the Flow Time in Online Scheduling with Structured Processing Sets.

Participants: Anthony Dugois, Loris Marchal, Louis-Claude Canon *(Univ. Besançon)*.

Replication in distributed key-value stores makes scheduling more challenging, as it introduces processing set restrictions, which limits the number of machines that can process a given task. We focus on the online minimization of the maximum response time in such systems, that is, we aim at bounding the latency of each task. When processing sets have no structure, Anand et al. (*Algorithmica*, 2017) derive a strong lower bound on the competitiveness of the problem: no online scheduling algorithm can have a competitive ratio smaller than $\Omega(m)$, where m is the number of machines. In practice, data replication schemes are regular, and structured processing sets may make the problem easier to solve. We derive new lower bounds for various common structures, including *inclusive*, *nested* or *interval* structures. In particular, we consider fixed sized intervals of machines, which mimic the standard replication strategy of key-value stores. We prove that EFT scheduling is $(3-\frac{2}{k})$-competitive when optimizing max-flow on *disjoint* intervals of size k. However, we show that the competitive ratio of EFT is at least $m-k+1$ when these intervals overlap, even when unit tasks are considered. We compare these two replication strategies in simulations and assess their efficiency when popularity biases are introduced, i.e., when some machines are accessed more frequently than others because they hold popular data.

This work has been accepted at IPDPS 2022 [19].

7.2.8 Memory-Aware Scheduling of Tasks Sharing Data on Multiple GPUs with Dynamic Runtime Systems.

Participants: Maxime Gonthier, Loris Marchal, Samuel Thibault *(Inria Bordeaux)*.

The use of accelerators such as GPUs has become mainstream to achieve high performance on modern computing systems. GPUs come with their own (limited) memory and are connected to the main memory of the machine through a bus (with limited bandwidth). When a computation is started on a

GPU, the corresponding data needs to be transferred to the GPU before the computation starts. Such data movements may become a bottleneck for performance, especially when several GPUs have to share the communication bus. Task-based runtime schedulers have emerged as a convenient and efficient way to use such heterogeneous platforms. When processing an application, the scheduler has the knowledge of all tasks available for processing on a GPU, as well as their input data dependencies. Hence, it is able to choose which task to allocate to which GPU and to reorder tasks so as to minimize data movements. We focus on this problem of partitioning and ordering tasks that share some of their input data. We present a novel dynamic strategy based on data selection to efficiently allocate tasks to GPUs and a custom eviction policy, and compare them to existing strategies using either a well-known graph partitioner or standard scheduling techniques in runtime systems. We also improved an offline scheduler recently proposed for a single GPU, by adding load balancing and task stealing capabilities. All strategies have been implemented on top of the STARPU runtime, and we show that our dynamic strategy achieves better performance when scheduling tasks on multiple GPU s with limited memory.

This work has been accepted at IPDPS 2022 [20].

7.3 Sparse direct solvers and sparsity in computing

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. We worked on combinatorial problems arising in sparse matrix and tensors computations. The computations involved direct methods for solving sparse linear systems and tensor factorizations. The combinatorial problems were based on matchings on bipartite graphs, partitionings, and hyperedge queries.

7.3.1 Trading Performance for Memory in Sparse Direct Solvers using Low-rank Compression.

Participants: Grégoire Pichon, Loris Marchal, Thibault Marette *(ENS Lyon)*, Frédéric Vivien.

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 at what point in the algorithm the 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. Approach 1 reaches a very small memory footprint generally at the expense of a greater execution time. Approach 2 achieves a smaller execution time but requires more memory. The objective of the proposed approach 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 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.

This work has been published in FGCS in 2022 [15].

7.3.2 An Efficient Parallel Implementation of a Perfect Hashing Method for Hypergraphs

Participants: Somesh Singh, Bora Uçar.

Querying the existence of an edge in a given graph or hypergraph is a building block in several algorithms. Hashing-based methods can be used for this purpose, where the given edges are stored in a hash table in a preprocessing step, and then the queries are answered using the lookup operations. While the general hashing methods have fast lookup times in the average case, the worst case run time is much higher. Perfect hashing methods take advantage of the fact that the items to be stored are all available and construct a collision free hash function for the given input, resulting in an optimal lookup time even in the worst case. We investigate an efficient shared-memory parallel implementation of a recently proposed perfect hashing method for hypergraphs. We experimentally compare the resulting parallel algorithms with the state-of-the-art and demonstrate better run time and scalability on a set of hypergraphs corresponding to real-life sparse tensors. This work was published at a worksop of IPDPS22 [22].

7.3.3 Scaling matrices and counting the perfect matchings in graphs

Participants: Fanny Dufossé *(Inria Grenoble)*, Kamer Kaya *(Sabanci Uni., Turkey)*, Ioannis Panagiotas, Bora Uçar.

We investigate efficient randomized methods for approximating the number of perfect matchings in bipartite graphs and general undirected graphs. Our approach is based on assigning probabilities to edges, randomly selecting an edge to be in a perfect matching, and discarding edges that cannot be put in a perfect matching. The probabilities are chosen according to the entries in the doubly stochastically scaled version of the adjacency matrix of the given graph. The experimental analysis on random and real-life graphs shows improvements in the approximation over previous and similar methods from the literature. This work appeared in a journal [12].

7.3.4 Algorithms and Data Structures for Hyperedge Queries

Participants: Jules Bertrand *(ENS de Lyon)*, Fanny Dufossé *(Inria Grenoble)*, Somesh Singh, Bora Uçar.

We consider the problem of querying the existence of hyperedges in hypergraphs. More formally, given a hypergraph, 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 runtime complexity of the proposed approach and experimentally compare it with the state-of-the-art hashing-based solutions. Experiments demonstrate the efficiency of the proposed approach with respect to the state-of-the-art. This work was first published a research report [24], the updated version of which is published in a journal [9].

8 Partnerships and cooperations

8.1 International initiatives

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 parts 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.

8.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program

ChalResil

Title: Challenges in resilience at scale

Duration: 2022 ->

Coordinator: Thomas Herault (herault@icl.utk.edu)

Partners:

- University of Tennessee, Knoxville (États-Unis)

Inria contact: Yves Robert

Summary: The associate team between ROMA and the DISCO group at ICL-UTK addresses major challenges that currently prevent the design of efficient resilient algorithms for large-scale scientific applications. Our results will facilitate the deployment of a wide range of applications and improve the utilization of the largest supercomputers in the world, thereby benefiting the entire JLESC laboratory, and beyond.

8.1.2 Inria associate team not involved in an IIL or an international program

PEACHTREE

Title: Shared memory sparse tensors computations: Combinatorial tools, scheduling, and numerical algorithms

Duration: 2020 ->

Coordinator: UMIT V. CATALYUREK (umit@gatech.edu)

Partners:

- GeorgiaTech Atlanta (États-Unis)

Inria contact: Bora Uçar

Summary: The PeachTree associated team between ROMA and TDAlab at GaTech addresses needs of sparse tensor computations on shared memory parallel systems. It investigates the building blocks of numerical parallel tensor computation algorithms, and designs a set of scheduling and combinatorial tools for achieving efficiency. The outcome will be an efficient library containing the numerical algorithms, scheduling and combinatorial tools for sparse tensor computations. URL of the project

8.1.3 Participation in other International Programs

FACCTS University of Chicago

Participants: Anne Benoit, Lucas Perotin, Yves Robert.

Title: Foundational Models and Efficient Algorithms for Scheduling with Variable Capacity Resources.

Partner Institution(s):
- University of Chicago (Andrew Chien, Chaojie Zhang)

Date/Duration: 2021–2024

Additionnal info/keywords: This is a FACCTS project. (CNRS - U. Chicago). We also received another grant to organize two workshops (one in March 2023 and one in March 2024) at the Paris Center of U. Chicago.

Homeland

Participants: Bora Uçar.

Title: Heidelberg & Lyon do machine learning for graph decomposition

Partner Institution(s):
- Heidelberg University, Germany

Date/Duration: 2022–2023

Additionnal info/keywords: This is a PHC Procope project. Homeland project's goal is to combine graph/hypergraph clustering with machine learning and obtain clustering algorithms with general objective functions to be used in varying applications. URL of the project

PIKS

Participants: Bora Uçar.

Title: Parallel Implementation of Karp–Sipser heuristic

Partner Institution(s):
- Simula, Norway

Date/Duration: 2020–2022 (due to the worldwide pandemic, the project was extended).

Additionnal info/keywords: This is a PHC Aurora project. 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. URL of the project

8.2 International research visitors

8.2.1 Visits of international scientists

Inria International Chair

Participants: Julien Langou *(University of Denver (USA))*.

Julien Langou has been granted an Inria International Chair to visit the team. He will start visiting the team during year 2023.

8.3 National initiatives

8.3.1 ANR Project SOLHARIS (2019-2023), 4 years.

Participants: Maxime Gonthier, Grégoire Pichon, Loris Marchal, Bora Uçar.

The ANR Project SOLHARIS 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.

8.3.2 ANR Project SPARTACLUS (2023-2027), 4 years.

Participants: Loris Marchal, Grégoire Pichon, Bora Uçar, Frédéric Vivien.

The ANR Project SPARTACLUS was launched in January 2023 for a duration of 48 months. This is a JCJC project lead by Grégoire Pichon and including other participants of the ROMA team. This project aims at building new ordering strategies to enhance the behavior of sparse direct solvers using low-rank compression.

The objective of this project is to end up with a common tool to perform the ordering and the clustering for sparse direct solvers when using low-rank compression. We will provide statistics that are currently missing and that will help understanding the compressibility of each block. The objective is to enhance sparse direct solvers, in particular targeting larger problems. The benefits will directly apply to academic or industrial applications using sparse direct solvers.

9 Dissemination

9.1 Promoting scientific activities

9.1.1 Scientific events: organisation

- Bora Uçar has organized the first SIAM ACDA Workshop in Aussois (workshop webpage)
- Grégoire Pichon and Loris Marchal organized the “15th Scheduling for Large Scale Systems Workshop” (workshop webpage)

General chair, scientific chair

- Anne Benoit is the general co-chair of IEEE IPDPS'22 (36th IEEE International Parallel & Distributed Processing Symposium).

Member of the organizing committees

- Anne Benoit is a member of the organizing committee of SIAM ACDA'23.

9.1.2 Scientific events: selection

Chair of conference program committees

- Yves Robert and Bora Uçar are the program co-chairs of IEEE IPDPS'22.
- Bora Uçar is the program vice-chair of SEA 2022 (20th Symposium on Experimental Algorithms).
- Yves Robert is the ACM Posters vice-chair of SC'22.
- Anne Benoit is the ACM SRC Graduate Posters chair of SC'22, and the Program area co-chair (parallel and distributed algorithms for computational science) of IPDPS'23.
- Frédéric Vivien is Research Posters vice-chair for SC'22.

Member of the conference program committees

- Anne Benoit was a member of the program committees of SC'22 and PPAM'22. She is a member of the program committee of EuroPar'23 and ACDA'23.
- Suraj Kumar was a member of the program committee of ICPP 2022.
- Loris Marchal was a member of the program committee of IPDPS 2022 and ICPP 2022.
- Grégoire Pichon was a member of the program committee of research posters for SC'22; Compas 2022; HiPC 2022 and ICPP 2022.
- Yves Robert was a member of the program committees of FTXS'22, SCALA'22, PMBS'22, SuperCheck'22 (co-located with SC'22) and Resilience (co-located with Euro-Par'22).
- Bora Uçar was a member of the Proceedings Paper Committee of the 20th SIAM Conference on Parallel Processing for Scientific Computing; Algorithms and Applications Track of the 2022 IEEE Cluster Conference; PPAM 2022 (14th International Conference on Parallel Processing and Applied Mathematics); ISC High Performance 2022 (Birds of a Feather Committee).
- Frédéric Vivien was a "Special Committee Member" of the program committee os IPDPS'22, and a member of the program committees of BigData 2022, IPDPS'23.

Reviewer

- Loris Marchal has reviewed papers for the conference: Mathematical Foundations of Computer Science.
- Bora Uçar has reviewed papers for conferences PPoPP2023, Principles and Practice of Parallel Programming; MFCS 2022, 47th International Symposium on Mathematical Foundations of Computer Science.

9.1.3 Journal

Member of the editorial boards

- Anne Benoit is Associate-Editor-in-Chief of JPDC (Journal of Parallel and Distributed Computing), and Associate Editor (in Chief) of the journal of Parallel Computing: Systems and Applications (ParCo).
- 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. He is also acting as a guest editor for a special issue of Journal of Parallel and Distributed Computing (on IPDPS22), and anso ACM JEA (on SEA2022).
- Yves Robert is a member of the editorial board of the International Journal of High Performance Computing (IJHPCA) and the Journal of Computational Science (JOCS).
- Frédéric Vivien is a member of the editorial board of Journal of Parallel and Distributed Computing and of the ACM Transactions on Parallel Computing.

Reviewer - reviewing activities

- Suraj Kumar has reviewed manuscripts for the journals: SIAM Journal on Scientific Computing, Transactions on Parallel and Distributed Systems, Journal of Parallel and Distributed Computing.
- Loris Marchal has reviewed manuscripts for the journals: IEEE Transactions on Emerging Topics in Computing, Parallel Computing and Journal of Combinatorial Optimization.
- Grégoire Pichon has reviewed manuscripts for the journals: Transactions on Parallel and Distributed Systems.
- Bora Uçar has reviewed manuscripts for the journals: Turkish Journal Of Electrical Engineering & Computer Sciences, Concurrency and Computation: Practice and Experience.

9.1.4 Invited talks

- Anne Benoit has given a keynote talk at the IEEE 34th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Bordeaux, France, November 2022.
- Loris Marchal has given a keynote talk at the Compas 2022 conference.
- Yves Robert has given a keynote talk at *Journées Scientifiques Inria (JSI)*, November 2022.

9.1.5 Leadership within the scientific community

- Anne Benoit is the Chair of IEEE Technical Community on Parallel Processing (TCPP).
- Yves Robert serves in the steering committee of IPDPS and HCW.
- Bora Uçar was the Secretary of the SIAM Activity Group on Applied and Computational Discrete Algorithms (ACDA), for the period 1 January 2021 – 31 December 2022.
- Bora Uçar was elected as the Program director of the SIAM Activity Group on Applied and Computational Discrete Algorithms (ACDA) for the period 1 January 2022 – 31 December 2023.

9.1.6 Scientific expertise

- Anne Benoit is a member of the selection committee for the IEEE CS TCHPC early career researchers award for excellence in HPC in 2022.
- Anne Benoit is a member of the organizing committee of ACDA online seminar series since 2022.
- Anne Benoit is a member of IEEE Future of Conferences Ad Hoc Committee, formed by IEEE CS president in 2022, to identify and recommend future models for conferences.
- Yves Robert was a member of the 2022 IEEE Fellow Committee.
- Yves Robert was the Chair of the 2022 IEEE Charles Babbage Award Committee.
- Bora Uçar has evaluated a project proposal for Vienna Science and Technology Fund (WWTF); he is also a member of the advisory board of The SparCity project webpage, which is funded by EuroHPC JU under the 2019 call of Extreme Scale Computing and Data Driven Technologies for research and innovation actions (project no 956213).
- 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.

9.2 Teaching - Supervision - Juries

9.2.1 Teaching

- Anne Benoit, Chair of the Computer Science department at ENS Lyon, France, since September 2022
- Licence: Anne Benoit, Responsible of the L3 students at ENS Lyon, France until August 2022
- 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: Grégoire Pichon, Resource optimization for linear system solvers, 12h, M2, ENS Lyon, France
- Master: Grégoire Pichon, Compilation / traduction des programmes, 22.5h, M1, Univ. Lyon 1, France
- Master: Grégoire Pichon, Systèmes avancés, 21h, M1, Univ. Lyon 1, France
- Master: Grégoire Pichon, Réseaux, 12h, M1, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Programmation concurrente, 27h, L3, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Réseaux, 40h, L3, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Système d'exploitation, 25.5h, L2, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Introduction aux réseaux et au web, 18h, L1, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Référent pédagogique, 30h, L1/L2/L3, Univ. Lyon 1, France
- Master: Grégoire Pichon, Bora Uçar, and Frédéric Vivien, Resource optimization for linear system solvers, 10h each, M2, ENS Lyon, France.
- Licence: Yves Robert, Probabilités et algorithmes randomisés, 48h, L3, ENS Lyon, France
- Agrégation Informatique: Yves Robert, Algorithmique, NP-complétude et algorithmes d'approximation, probabilités, graphes, structures de données, 75h, ENS Lyon, France

9.2.2 Supervision

- PhD in progress: Brian Bantsoukissa, "Ordering sparse matrices to enhance low-rank compressibility in the context of sparse direct solvers", funding: Inria, advisors: Grégoire Pichon and Bora Uçar.
- 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: Redouane Elghazi, "Stochastic Scheduling for HPC Systems", started in September 2020, funding: Région Franche-Comté, advisors: Anne Benoit, Louis-Claude Canon and Pierre-Cyrille Héam.
- 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). The PhD was terminated on August 31, 2022, by ENS de Lyon due to the applicant incapacity to meet languages requirements.
- PhD defended: Yishu Du, "Resilient algorithms and scheduling techniques for numerical algorithms", started in December 2019, funding: China Scholarship Council, advisors: Loris Marchal and Yves Robert, defended in December 2022.
- PhD in progress: Anthony Dugois "Scheduling for key value stores", started in October 2020, funding: Inria, advisors: Loris Marchal and Louis-Claude Canon (Univ. Besançon).
- PhD in progress: Maxime Gonthier "Memory-Aware scheduling for task-based runtime systems", started in October 2020, funding: Inria, advisors: Loris Marchal and Samuel Thibault (Univ. Bordeaux).

9.2.3 Juries

- Anne Benoit was a reviewer for the PhD thesis of Etienne Mauffret, Université Savoie Mont Blanc, France, June 20, 2022. Title: Placement des réplicas dans un système de gestion de données distribué à large échelle à protocole de cohérence adaptable.
- Loris Marchal is a responsible of the competitive selection of ENS Lyon students for Computer Science, and is a member of the jury of this competitive exam.
- Grégoire Pichon was a member of the PhD dissertation examination committee of Esragul Korkmaz, Inria Bordeaux, France, September 21, 2022. Title: Improving the memory and time overhead of low-rank parallel linear sparse direct solvers.
- Bora Uçar was a member of the PhD dissertation examination committee of Nabil F. Abubaker, Bilkent University, Ankara, Turkey July 6, 2022. Title: Novel Algorithms and Models for Scaling Parallel Sparse Tensor and Matrix Factorizations.

9.3 Popularization

9.3.1 Articles and contents

- Yves Robert, together with George Bosilca, Aurélien Bouteiller and Thomas Herault, gave a full-day tutorial at SC'22 on *Fault-tolerant techniques for HPC and Big Data: theory and practice*.
- Bora Uçar has co-authored a SIAM News article on the SIAM Activity Group on Applied and Computational Discrete Algorithms (SIAG/ACDA) workshop that was held in Aussois available online.

10 Scientific production

10.1 Major publications

[1] A. Benoit, T. Hérault, 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. Hérault and Y. Robert. 'Fault Tolerance Techniques for High-Performance Computing'. In: *Fault-Tolerance Techniques for High-Performance Computing*. Ed. by T. Hérault 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. 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. 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. URL: https://hal.inria.fr/hal-02025521.

10.2 Publications of the year

International journals

[7] A. Benoit, L.-C. Canon, R. Elghazi and P.-C. Heam. 'List and shelf schedules for independent parallel tasks to minimize the energy consumption with discrete or continuous speeds'. In: *Journal of Parallel and Distributed Computing* (2022). URL: https://hal.inria.fr/hal-03920697.

[8] A. Benoit, L. Perotin, Y. Robert and H. Sun. 'Checkpointing Workflows à la Young/Daly Is Not Good Enough'. In: *ACM Transactions on Parallel Computing* 9.4 (31st Dec. 2022), pp. 1–25. DOI: 10.1145/3548607. URL: https://hal.inria.fr/hal-03920329.

[9] J. Bertrand, F. Dufossé, S. Singh and B. Uçar. 'Algorithms and Data Structures for Hyperedge Queries'. In: *ACM Journal of Experimental Algorithmics* 27 (31st Dec. 2022), pp. 1–23. DOI: 10.1145/3568421. URL: https://hal.inria.fr/hal-03905905.

[10] G. Bosilca, A. Bouteiller, T. Hérault, V. Le Fèvre, Y. Robert and J. J. Dongarra. 'Comparing Distributed Termination Detection Algorithms for Task-Based Runtime Systems on HPC platforms'. In: *International Journal of Networking and Computing* 12.1 (2022). URL: https://hal.inria.fr/hal-03920388.

[11] Y. Du, L. Marchal, G. Pallez and Y. Robert. 'Optimal Checkpointing Strategies for Iterative Applications'. In: *IEEE Transactions on Parallel and Distributed Systems* 33.3 (1st Mar. 2022), pp. 507–522. DOI: 10.1109/TPDS.2021.3099440. URL: https://hal.inria.fr/hal-03338278.

[12] F. Dufossé, K. Kaya, I. Panagiotas and B. Uçar. 'Scaling matrices and counting the perfect matchings in graphs'. In: *Discrete Applied Mathematics* 308 (Feb. 2022), pp. 130–146. DOI: 10.1016/j.dam.2020.07.016. URL: https://hal.inria.fr/hal-01743802.

[13] Y. Gao, G. Pallez, Y. Robert and F. Vivien. 'Dynamic Scheduling Strategies for Firm Semi-Periodic Real-Time Tasks'. In: *IEEE Transactions on Computers* 72.1 (1st Jan. 2023), pp. 55–68. DOI: 10.1109/TC.2022.3208203. URL: https://hal.inria.fr/hal-03778357.

[14] C. Gou, A. Benoit, M. Chen, L. Marchal and T. Wei. 'Mapping series-parallel streaming applications on hierarchical platforms with reliability and energy constraints'. In: *Journal of Parallel and Distributed Computing* 163 (May 2022), pp. 45–61. DOI: 10.1016/j.jpdc.2022.01.016. URL: https://hal.inria.fr/hal-03863951.

[15] L. Marchal, T. Marette, G. Pichon and F. Vivien. 'Trading Performance for Memory in Sparse Direct Solvers using Low-rank Compression'. In: *Future Generation Computer Systems* 130 (May 2022), pp. 307–320. DOI: 10.1016/j.future.2021.12.018. URL: https://hal.inria.fr/hal-03517124.

[16] L. Marchal, S. McCauley, B. Simon and F. Vivien. 'Minimizing I/Os in Out-of-Core Task Tree Scheduling'. In: *International Journal of Foundations of Computer Science* (22nd July 2022), pp. 1–30. DOI: 10.1142/s0129054122500186. URL: https://hal.archives-ouvertes.fr/hal-03758021.

International peer-reviewed conferences

[17] A. Benoit, Y. Du, T. Herault, L. Marchal, G. Pallez, L. Perotin, Y. Robert, H. Sun and F. Vivien. 'Checkpointing à la Young/Daly: An Overview'. In: IC3 2022 - 2022 Fourteenth International Conference on Contemporary Computing. Noida, India: ACM, 4th Aug. 2022, pp. 701–710. DOI: 10.1145/3549206.3549328. URL: https://hal.inria.fr/hal-03830322.

[18] A. Benoit, L. Perotin, Y. Robert and H. Sun. 'Online Scheduling of Moldable Task Graphs under Common Speedup Models'. In: ICPP 2022 - 51st International Conference on Parallel Processing. Bordeaux, France, 29th Aug. 2022. URL: https://hal.inria.fr/hal-03778405.

[19] L.-C. Canon, A. Dugois and L. Marchal. 'Bounding the Flow Time in Online Scheduling with Structured Processing Sets'. In: IPDPS 2022 - 36th IEEE International Parallel & Distributed Processing Symposium. Lyon, France: IEEE, 30th May 2022, pp. 1–11. URL: https://hal.archives-ouvertes.fr/hal-03561018.

[20] M. Gonthier, L. Marchal and S. Thibault. 'Memory-Aware Scheduling of Tasks Sharing Data on Multiple GPUs with Dynamic Runtime Systems'. In: IPDPS 2022 - 36th IEEE International Parallel & Distributed Processing Symposium. Lyon, France: IEEE, 30th May 2022, pp. 1–11. DOI: 10.1109/IPDPS53621.2022.00073. URL: https://hal.inria.fr/hal-03552243.

[21] S. Kulagina, H. Meyerhenke and A. Benoit. 'Mapping Tree-shaped Workflows on Memory-heterogeneous Architectures'. In: 20th Int. Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms (HeteroPar). Glasgow, United Kingdom, 23rd Aug. 2022. URL: https://hal.inria.fr/hal-03921445.

[22] S. Singh and B. Uçar. 'An Efficient Parallel Implementation of a Perfect Hashing Method for Hypergraphs'. In: GrAPL 2022 - Workshop on Graphs, Architectures, Programming, and Learning. Lyon, France, 2022, pp. 265–274. URL: https://hal.inria.fr/hal-03612360.

Reports & preprints

[23] A. Benoit, L. Perotin, Y. Robert and F. Vivien. *Checkpointing strategies to protect parallel jobs from non-memoryless fail-stop errors*. RR-9465. Inria - Research Centre Grenoble – Rhône-Alpes, Mar. 2022, p. 42. URL: https://hal.inria.fr/hal-03610883.

[24] J. Bertrand, F. Dufossé, S. Singh and B. Uçar. *Algorithms and data structures for hyperedge queries*. RR-9390. Inria Grenoble Rhône-Alpes, 29th Apr. 2022, p. 28. URL: https://hal.inria.fr/hal-03127673.

[25] L.-C. Canon, A. Dugois and L. Marchal. *Bounding the Flow Time in Online Scheduling with Structured Processing Sets (extended version)*. RR-9446. INRIA, Jan. 2022, pp. 1–35. URL: https://hal.archives-ouvertes.fr/hal-03558600.

[26] Y. Du, L. Marchal, G. Pallez and Y. Robert. *Doing better for jobs that failed: node stealing from a batch scheduler's perspective*. 15th Apr. 2022. URL: https://hal.inria.fr/hal-03643403.

[27] Y. Gao, G. Pallez, Y. Robert and F. Vivien. *Scheduling Strategies for Overloaded Real-Time Systems*. RR-9455. Inria - Research Centre Grenoble – Rhône-Alpes, Feb. 2022, pp. 1–48. URL: https://hal.inria.fr/hal-03580853.

[28] M. Gonthier, L. Marchal and S. Thibault. *Taming data locality for GPU task scheduling in runtime systems*. 29th Mar. 2022. URL: https://hal.inria.fr/hal-03623220.

[29] E. Korkmaz, M. Faverge, G. Pichon and P. Ramet. *Reaching the Quality of SVD for Low-Rank Compression Through QR Variants*. RR-9476. Inria Bordeaux - Sud Ouest, July 2022, p. 43. URL: https://hal.inria.fr/hal-03718312.

[30] S. Kulagina, H. Meyerhenke and A. Benoit. *Mapping Tree-shaped Workflows on Memory-heterogeneous Architectures*. RR-9458. Inria Grenoble Rhône-Alpes, Feb. 2022, pp. 1–20. URL: https://hal.inria.fr/hal-03581418.

[31] Z. Wu, L. Han, J. Liu, Y. Robert and F. Vivien. *Energy-aware mapping and scheduling strategies for real-time workflows under reliability constraints*. RR-9469. INRIA, Apr. 2022, pp. 1–23. URL: https://hal.inria.fr/hal-03641039.