RESEARCH CENTRE

Inria Lyon Centre

IN PARTNERSHIP WITH:

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

2024
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

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

- Frédéric Vivien [Team leader, INRIA, Senior Researcher, Team leader starting from 31/07/2024]
- Suraj Kumar [INRIA, ISFP]
- Loris Marchal [CNRS, Senior Researcher, Team leader until 30/07/2024]
- Bora Uçar [CNRS, Senior Researcher]

Faculty Members

- Anne Benoît [ENS DE LYON, Associate Professor]
- Redouane Elghazi [ENS DE LYON, ATER, until Sep 2024]
- Grégoire Pichon [UNIV LYON I, Associate Professor, until Aug 2024]
- Yves Robert [ENS DE LYON, Professor]

Post-Doctoral Fellows

- Antoine Jego [SORBONNE UNIVERSITE]
- Maher Mallem [INRIA, Post-Doctoral Fellow, from Nov 2024]
- Somesh Singh [UDL, Post-Doctoral Fellow, until Aug 2024]

PhD Student

- Joachim Cendrier [CNRS]

Interns and Apprentices

- Mathis Lamiroy [ENS DE LYON, until Feb 2024]
- Damien Lesens [ENS DE LYON, Intern, from Sep 2024]
- Damien Lesens [ENS DE LYON, Intern, from Feb 2024 until Jul 2024]
- Adrien Obrecht [ENS DE LYON, Intern, from Sep 2024]
- Adrien Obrecht [ENS DE LYON, Intern, from Feb 2024 until Jul 2024]
- Alix Tremodeux [ENS DE LYON, Intern, from Feb 2024 until Jul 2024]
- Felix Wirth-Bonne [INRIA, Intern, from May 2024 until Nov 2024]

Administrative Assistant

- Chrystelle Mouton [INRIA]

Visiting Scientists

- Rob Bisseling [UTRECHT UNIVERSITY, from Feb 2024 until May 2024]
- Julien Langou [UNIV COLORADO, From April until July 2024, and from Nov 2024]

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 several hundreds of thousands of cores, and edge servers represent many millions of resources. 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.

Combinatorial scientific computing (CSC) is a term 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 Social and environmental responsibility

5.1 Impact of research results

Within the framework of our collaboration with the University of Chicago (see Section 10.1.4) we explore novel scheduling algorithms that are able to adapt to dynamic power changes, to reduce carbon emissions, and to give priority to using green energy sources.

6 Highlights of the year

6.1 Awards

Bora Uçar received TPDS Award for Editorial Excellence for his service in 2023; established in 2019, this award recognizes exceptional contributions by members of the TPDS editorial board.

7 New software, platforms, open data

7.1 New software

7.1.1 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: Bora Uçar, Kamer Kaya, Johannes Langguth

7.1.2 PaStiX

Name: Parallel Sparse matriX package

Keywords: Direct solvers, Parallel numerical solvers, Linear Systems Solver

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

Publications: inria-00346017, inria-00346018, hal-01485507, hal-01824275, hal-03361299

Contact: Pierre Ramet

Participants: Alycia Lisito, Grégoire Pichon, Mathieu Faverge, Pierre Ramet

8 New results

8.1 Resilience for very large scale platforms

The ROMA team has been working on resilience problems for several years. In 2024, we have focused on several problems.

8.1.1 Minimizing Energy Consumption for Real-Time Tasks on Heterogeneous Platforms Under Deadline and Reliability Constraints

Participants: Yiqin Gao *(Shanghai Jiao Tong University)*, Li Han *(ECNU - East China Normal University)*, Jing Liu *(ECNU - East China Normal University)*, Yves Robert, Frédéric Vivien.

As real-time systems are safety critical, guaranteeing a high reliability threshold is as important as meeting all deadlines. Periodic tasks are replicated to mitigate the negative impact of transient faults, which leads to redundancy and high energy consumption. On the other hand, energy saving is widely identified as increasingly relevant issues in real-time systems. In this work, we formalize this challenging tri-criteria optimization problem, i.e., minimizing the expected energy consumption while enforcing the reliability threshold and meeting all task deadlines, and propose several mapping and scheduling heuristics to solve it. Specifically, a novel approach is designed to (i) map an arbitrary number of replicas onto processors, (ii) schedule each replica of each task instance on its assigned processor with less temporal overlap. The platform is composed of processing units with different characteristics, including speed profile, energy cost and fault rate. The heterogeneity of the computing platform makes the problem more complicated, because different mappings achieve different levels of reliability and consume different amounts of energy. Moreover, scheduling plays an important role in energy saving, as the expected energy consumption is the average over Minimizing energy consumption for real-time tasks all failure scenarios. Once a task replica is successful, the other replicas of that task instance can be canceled, which calls for minimizing the overlap between any replica pair. Finally, to quantitatively analyze our methods, we derive a theoretical lower-bound for the expected energy consumption. Comprehensive experiments are conducted on a large set of execution scenarios and parameters. The comparison results reveal that our strategies perform better than the random baseline under almost all settings, with an average gain in energy consumption of more than 40%, and our best heuristic achieves an excellent performance: its energy saving is only 2% less than the lower-bound on average.

This work has been published in the journal Algorithmica [15].

8.1.2 A Survey on Checkpointing Strategies

Participants: Anne Benoit, Leonardo Bautista-Gomez *(Barcelona Supercomputing Center, Spain)*, Sheng Di *(Argonne National Laboratory, USA)*, Thomas Herault *(University of Tennessee, Knoxville, USA)*, Yves Robert, Hongyang Sun *(University of Kansas, USA)*.

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 an invited publication at IC3 2022, we had provided some background and various application scenarios to assess the usefulness and limitations of the formula. In 2023, we have considerably extended the scope of our survey and we have published this contribution to the special issue of FGCS focusing on JLESC collaboration results [8].

8.1.3 Checkpointing strategies for a fixed-length execution

Participants: Anne Benoit, Lucas Perotin, Yves Robert, Frédéric Vivien.

This work considers checkpointing strategies for a parallel application executing on a large-scale platform whose nodes are subject to failures. The application executes for a fixed duration, namely the length of the reservation that it has been granted. We start with small examples that show the difficulty of the problem: it turns out that the optimal checkpointing strategy neither always uses periodic checkpoints nor always takes its last checkpoint exactly at the end of the reservation. Then, we introduce a dynamic heuristic that is periodic and decides for the checkpointing frequency based upon thresholds for the time left; we determine threshold times Tn such that it is best to plan for exactly n checkpoints if the time left (or initially the length of the reservation) is between Tn and Tn+1. Next, we use time discretization and design a (complicated) dynamic programming algorithm that computes the optimal solution, without any restriction on the checkpointing strategy. Finally, we report the results of an extensive simulation campaign that shows that the optimal solution is far more efficient than the Young/Daly periodic approach for short or mid-size reservations.

This work has been published in FTXS'2024, a workshop co-located with SC'2024 [16].

8.1.4 Checkpointing strategies to tolerate non-memoryless failures on HPC platforms

Participants: Anne Benoit, Lucas Perotin, Yves Robert, Frédéric Vivien.

This work studies checkpointing strategies for parallel applications subject to failures. The optimal strategy to minimize total execution time, or makespan, is well known when failure inter-arrival times obey an Exponential distribution, but it is unknown for non-memoryless failure distributions. We explain why the latter fact is misunderstood in recent literature. We propose a general strategy that maximizes the expected efficiency until the next failure, and we show that this strategy achieves an asymptotically optimal makespan, thereby establishing the first optimality result for arbitrary failure distributions. Through extensive simulations, we show that the new strategy is always at least as good as the Young/Daly strategy for various failure distributions. For distributions with a high infant mortality (such as LogNormal with shape parameter $k = 2.51$ or Weibull with shape parameter 0.5), the execution time is divided by a factor 1.9 on average, and up to a factor 4.2 for recently deployed platforms.

This work has been published in the ACM TOPC journal [11].

8.1.5 Improving Batch Schedulers with Node Stealing for Failed Jobs

Participants: Yishu Du, Loris Marchal, Guillaume Pallez *(Inria Rennes)*, Yves Robert.

After a machine failure, batch schedulers typically re-schedule the job that failed with a high priority. This is fair for the failed job but still requires that job to re-enter the submission queue and to wait for enough resources to become available. The waiting time can be very long when the job is large and the platform highly loaded, as is the case with typical HPC platforms. We propose another strategy: when a job J fails, if no platform node is available, we steal one node from another job J', and use it to continue the execution of J despite the failure. In this work, we give a detailed assessment of this node stealing strategy using traces from the Mira supercomputer at Argonne National Laboratory. The main conclusion is that node stealing improves the utilization of the platform and dramatically reduces the flow of large jobs, at the price of slightly increasing the flow of small jobs.

This work has been published in the CCPE journal [14].

8.2 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.

8.2.1 Revisiting I/O bandwidth-sharing strategies for HPC applications

Participants: Anne Benoit, Thomas Herault *(University of Tennessee, Knoxville, USA)*, Lucas Perotin, Yves Robert, Frédéric Vivien.

This work revisits I/O bandwidth-sharing strategies for HPC applications. When several applications post concurrent I/O operations, well-known approaches include serializing these operations (FCFS) or fair-sharing the bandwidth across them (FairShare). Another recent approach, I/O-Sets, assigns priorities to the applications, which are classified into different sets based upon the average length of their iterations. We introduce several new bandwidth-sharing strategies, some of them simple greedy algorithms, and some of them more complicated to implement, and we compare them with existing ones. Our new strategies do not rely on any a-priori knowledge of the behavior of the applications, such as the length of work phases, the volume of I/O operations, or some expected periodicity. We introduce a rigorous framework, namely steady-state windows, which enables to derive bounds on the competitive ratio of all bandwidth-sharing strategies for three different objectives: minimum yield, platform utilization, and global efficiency. To the best of our knowledge, this work is the first to provide a quantitative assessment of the online competitiveness of any bandwidth-sharing strategy. This theory-oriented assessment is complemented by a comprehensive set of simulations, based upon both synthetic and realistic traces. The main conclusion is that two of our simple and low-complexity greedy strategies significantly outperform FCFS, FairShare and I/O-Sets, and we recommend that the I/O community would implement them for further assessment.

This work has been published in [10].

8.2.2 Concealing compression-accelerated I/O for HPC applications through In Situ task scheduling

Participants: Franck Cappello *(Argonne National Laboratory)*, Sheng Di *(Argonne National Laboratory)*, Sian Jin *(Indiana University)*, Yves Robert, Dingwen Tao *(Indiana University)*, Frédéric Vivien, Daoce Wang *(Indiana University)*.

Lossy compression and asynchronous I/O are two of the most effective solutions for reducing storage overhead and enhancing I/O performance in large-scale high-performance computing (HPC) applications. However, current approaches have limitations that prevent them from fully leveraging lossy compression, and they may also result in task collisions, which restrict the overall performance of HPC applications. To address these issues, we propose an optimization approach for the task scheduling problem that encompasses computation, compression, and I/O. Our algorithm adaptively selects the optimal compression and I/O queue to minimize the performance degradation of the computation. We also introduce an intra-node I/O workload balancing mechanism that evenly distributes the workload across different processes. Additionally, we design a framework that incorporates fine-grained compression, a compressed data buffer, and a shared Huffman tree to fully benefit from our proposed task scheduling. Experimental results with up to 16 nodes and 64 GPUs from ORNL Summit, as well as real-world HPC applications, demonstrate that our solution reduces I/O overhead by up to 3.8×× and 2.6×× compared to non-compression and asynchronous I/O solutions, respectively.

This work has been published in EuroSys'2024 [20]. A summary of this work is also included in a survey paper [12].

8.2.3 Mapping Large Memory-constrained Workflows onto Heterogeneous Platforms

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

Scientific workflows are often represented as directed acyclic graphs (DAGs), where vertices correspond to tasks and edges represent the dependencies between them. Since these graphs are often large in both the number of tasks and their resource requirements, it is important to schedule them efficiently on parallel or distributed compute systems. Typically, each task requires a certain amount of memory to be executed and needs to communicate data to its successor tasks. The goal is thus to execute the workflow as fast as possible (i.e., to minimize its makespan) while satisfying the memory constraints.

Hence, we investigate the partitioning and mapping of DAG-shaped workflows onto heterogeneous platforms where each processor can have a different speed and a different memory size. We first propose a baseline algorithm in the absence of existing memory-aware solutions. As our main contribution, we then present a four-step heuristic. Its first step is to partition the input DAG into smaller blocks with an existing DAG partitioner. The next two steps adapt the resulting blocks of the DAG to fit the processor memories and optimize for the overall makespan by further splitting and merging these blocks. Finally, we use local search via block swaps to further improve the makespan. Our experimental evaluation on real-world and simulated workflows with up to 30,000 tasks shows that exploiting the heterogeneity with the four-step heuristic reduces the makespan by a factor of 2.44 on average (even more on large workflows), compared to the baseline that ignores heterogeneity.

This is an extension of the work done on tree-shaped workflow the previous year, that was published in CCPE. The current work focuses on general DAG, and it was published in ICPP'2024 [21].

8.2.4 Scheduling requests in replicated key-value stores

Participants: Sonia Ben Mokhtar *(LIRIS, UCBL)*, Louis-Claude Canon *(FEMTO-ST, UFC)*, Anthony Dugois *(FEMTO-ST, UFC)*, Loris Marchal, Étienne Rivière *(ICTEAM, UCL)*.

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 that highlight which properties enable limiting tail latency: for instance, the EarliestFinishTime strategy—which uses the earliest next 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.

This work was published in the Journal of Scheduling [9].

8.2.5 Solving the Restricted Assignment Problem to Schedule Multi-Get Requests in Key-Value Stores

Participants: Louis-Claude Canon *(FEMTO-ST, UFC)*, Anthony Dugois *(FEMTO-ST, UFC)*, Loris Marchal.

Modern distributed key-value stores, such as Apache Cassandra, enhance performance through multi-get requests, minimizing network round-trips between the client and the database. However, partitioning these requests for appropriate storage server distribution is non-trivial and may result in imbalances. This study addresses this optimization challenge as the Restricted Assignment problem on Intervals (RAI). We propose an efficient $(2-1/m)$-approximation algorithm, where m is the number of machines. Then, we generalize the problem to the Restricted Assignment problem on Circular Intervals (RACI), matching key-value store implementations, and we present an optimal $O(nlogn)$ algorithm for RACI with fixed machines and unitary jobs. Additionally, we obtain a $(4-2/m)$-approximation for arbitrary jobs and introduce new heuristics, whose solutions are very close to the optimal in practice. Finally, we show that optimizing multi-get requests individually also leads to global improvements, increasing achieved throughput by 27%-34% in realistic cases compared to state-of-the-art strategy.

This work was presented at the EuroPar 2024 conference [17].

8.2.6 Data-Driven Locality-Aware Batch Scheduling

Participants: Maxime Gonthier *(University of Chicago)*, Elisabeth Larsson *(Uppsala Universitet)*, Loris Marchal, Carl Nettelblad *(Uppsala Universitet)*, Samuel Thibault *(Inria Bordeaux)*.

Clusters employ workload schedulers such as the Slurm Workload Manager to allocate computing jobs onto nodes. These schedulers usually aim at a good trade-off between in- creasing resource utilization and user satisfaction (decreasing job waiting time). However, these schedulers are typically unaware of jobs sharing large input files, which may happen in data intensive scenarios. The same input files may end up being loaded several times, leading to a waste of resources. We study how to design a data-aware job scheduler that is able to keep large input files on the computing nodes, without impacting other memory needs, and can benefit from previously- loaded files to decrease data transfers in order to reduce the waiting times of jobs. We present three schedulers capable of distributing the load between the computing nodes as well as re-using input files already loaded in the memory of some node as much as possible. We perform simulations with single node jobs using traces of real HPC-cluster usage, to compare them to classical job schedulers. The results show that keeping data in local memory between successive jobs and using data-locality information to schedule jobs improves performance compared to a widely-used scheduler (FCFS, with and without backfilling): a reduction in job waiting time (a 7.5% improvement in stretch), and a decrease in the amount of data transfers (7%).

This work has been presented at the APDCM 2024 workshop [19].

8.3 Sparse direct solvers and sparsity in computing

We continued our work sparse tensors by looking at the tensor contraction operation, which is a higher order analogue of the sparse matrix sparse matrix multiplication operation. We worked on combinatorial problems arising in sparse tensor models. Further work on graph algorithms and Birkhoff–von Neumann decomposition using methods from the signal processing domain were also conducted. We have also investigated communication lower bounds and algorithms achieving those bounds for certain (dense) matrix and tensor kernels.

8.3.1 Engineering edge orientation algorithms

Participants: Henrik Reinstädtler *(Heidelberg Univ.)*, Christian Schulz *(Heidelberg Univ.)*, Bora Uçar.

Given an undirected graph G, the edge orientation problem asks for assigning a direction to each edge to convert G into a directed graph. The aim is to minimize the maximum out degree of a vertex in the resulting directed graph. This problem, which is solvable in polynomial time, arises in many applications. An ongoing challenge in edge orientation algorithms is their scalability, particularly in handling large-scale networks with millions or billions of edges efficiently. We propose a novel algorithmic framework based on finding and manipulating simple paths to face this challenge. Our framework is based on an existing algorithm and allows many algorithmic choices. By carefully exploring these choices and engineering the underlying algorithms, we obtain an implementation which is more efficient and scalable than the current state-of-the-art. Our experiments demonstrate significant performance improvements compared to state-of-the-art solvers. On average our algorithm is 6.59 times faster when compared to the state-of-the-art.

This work is published at a conference [24], and the codes are made available online with an MIT license.

8.3.2 Orthogonal matching pursuit-based algorithms for the Birkhoff-von Neumann decomposition

Participants: Damien Lesens, Jérémy E. Cohen *(Myriad team, CREATIS)*, Bora Uçar.

Birkhoff-von Neumann (BvN) decomposition writes a doubly stochastic matrix as a convex combination of permutation matrices. For a given doubly stochastic matrix, the decomposition in general is not unique. In many applications a sparsest decomposition, that is with the smallest number of permutation matrices is of interest. This problem is known to be NP-complete, and heuristics are used to obtain sparse solutions. We propose heuristics based on the well-known orthogonal matching pursuit for sparse BvN decomposition. We experimentally compare our heuristics with the state of the art from the literature and show how our methods advance the known heuristics.

This work is published at a conference [22].

8.3.3 Efficient sparse tensor contraction

Participants: Somesh Singh, Bora Uçar.

We investigate the performance of algorithms for sparse tensor-sparse tensor multiplication (SpGeTT). This operation, also called sparse tensor contraction, is a higher order analogue of the sparse matrix-sparse matrix multiplication (SpGeMM) operation. Therefore, SpGeTT can be performed by first converting the input tensors into matrices, then invoking high performance variants of SpGeMM, and finally reconverting the resultant matrix into a tensor. Alternatively, one can carry out the scalar operations underlying SpGeTT in the realm of tensors without matrix formulation. We discuss the building blocks in both approaches and formulate a hashing-based method to avoid costly search or redirection operations. We present performance results with the current state-of-the-art SpGeMM-based approaches, existing SpGeTT approaches, and a carefully implemented SpGeTT approach with a new fine-tuned hashing method, proposed in this paper. We evaluate the methods on real world tensors, contracting a tensor with itself along various dimensions. Our proposed hashing-based method for SpGETT consistently outperforms the state-of-the-art method, achieving a 25% reduction in sequential execution time on average and a 21% reduction in parallel execution time on average across a variety of input instances.

This work is explained in a technical report [31].

8.3.4 Connectivity of a random directed graph model

Participants: Anne Benoit, Kamer Kaya *(Sabanci Univ., Türkiye)*, Bora Uçar.

We study the strong connectivity of directed graphs belonging to a random model with three parameters n, d, p. The parameter n defines the number of vertices. Each d-tuple of vertices is picked independently with probability p and if picked, $d-1$ directed edges from the vertex in the first position in the picked tuple to all others are added to the edge set. For $d = 2$, the model thus reduces down to the well-known Erdös–Renyi random directed graphs. The higher order case $d > 2$ arises in the spectral analysis of sparse tensors. We first investigate the threshold phenomenon for the strong connectivity of the directed random graphs from this model. Then, we conduct a series of experiments aimed at gaining a deeper understanding of these directed random graphs.

This work is explained in a technical report [25].

8.3.5 Communication Lower Bounds and Optimal Algorithms for Multiple Tensor-Times-Matrix Computation

Participants: Hussam Al Daas *(Rutherford Appleton Laboratory, UK)*, Grey Ballard *(Wake Forest University, USA)*, Laura Grigori *(EPFL, Switzerland)*, Suraj Kumar, Kathryn Rouse *(Inmar Intelligence, USA)*.

Multiple Tensor-Times-Matrix (Multi-TTM) is a key computation in algorithms for computing and operating with the Tucker tensor decomposition, which is frequently used in multidimensional data analysis. We establish communication lower bounds that determine how much data movement is required (under mild conditions) to perform the Multi-TTM computation in parallel. The crux of the proof relies on analytically solving a constrained, nonlinear optimization problem. We also present a parallel algorithm to perform this computation that organizes the processors into a logical grid with twice as many modes as the input tensor. We show that with correct choices of grid dimensions, the communication cost of the algorithm attains the lower bounds and is therefore communication optimal. Finally, we show that our algorithm can significantly reduce communication compared to the straightforward approach of expressing the computation as a sequence of tensor-times-matrix operations when the input and output tensors vary greatly in size.

This work has been published in the SIMAX journal [13].

8.3.6 Communication Lower Bounds and Optimal Algorithms for Symmetric Matrix Computations

Participants: Hussam Al Daas *(Rutherford Appleton Laboratory, UK)*, Grey Ballard *(Wake Forest University, USA)*, Laura Grigori *(EPFL, Switzerland)*, Suraj Kumar, Kathryn Rouse *(Inmar Intelligence, USA)*, Mathieu Vérité *(EPFL, Switzerland)*.

In this work, we focus on the communication costs of three symmetric matrix computations: i) multiplying a matrix with its transpose, known as a symmetric rank-k update (SYRK) ii) adding the result of the multiplication of a matrix with the transpose of another matrix and the transpose of that result, known as a symmetric rank-2k update (SYR2K) iii) performing matrix multiplication with a symmetric input matrix (SYMM). All three computations appear in the Level 3 Basic Linear Algebra Subroutines (BLAS) and have wide use in applications involving symmetric matrices. We establish communication lower bounds for these kernels using sequential and distributed-memory parallel computational models, and we show that our bounds are tight by presenting communication-optimal algorithms for each setting. Our lower bound proofs rely on applying a geometric inequality for symmetric computations and analytically solving constrained nonlinear optimization problems. The symmetric matrix and its corresponding computations are accessed and performed according to a triangular block partitioning scheme in the optimal algorithms.

This work has been submitted in the TOPC journal (Aug 2024) [28].

8.3.7 Tightening I/O Lower Bounds through the Hourglass Dependency Pattern

Participants: Lionel Eyraud-Dubois *(TOPAL)*, Guillaume Iooss *(CORSE)*, Julien Langou, Fabrice Rastello *(CORSE)*.

When designing an algorithm, one cares about arithmetic/computational complexity, but data movement (I/O) complexity plays an increasingly important role that highly impacts performance and energy consumption. For a given algorithm and a given I/O model, scheduling strategies such as loop tiling can reduce the required I/O down to a limit, called the I/O complexity, inherent to the algorithm itself. The objective of I/O complexity analysis is to compute, for a given program, its minimal I/O requirement among all valid schedules. We consider a sequential execution model with two memories, an infinite one, and a small one of size S on which the computations retrieve and produce data. The I/O is the number of reads and writes between the two memories. We identify a common "hourglass pattern" in the dependency graphs of several common linear algebra kernels. Using the properties of this pattern, we mathematically prove tighter lower bounds on their I/O complexity, which improves the previous state-of-the-art bound by a parametric ratio. This proof was integrated inside the IOLB automatic lower bound derivation tool.

This work was presented at the SPAA 2024 conference [18].

8.3.8 Enhancing sparse direct solver scalability through runtime system automatic data partition

Participants: Alycia Lisito *(TOPAL)*, Mathieu Faverge *(TOPAL)*, Grégoire Pichon, Pierre Ramet *(TOPAL)*.

With the ever-growing number of cores per node, it is critical for runtime systems and applications to adapt the task granularity to scale on recent architectures. Among applications, sparse direct solvers are a time-consuming step and the task granularity is rarely adapted to large many-core systems. In this work, we investigate the use of runtime systems to automatically partition tasks in order to achieve more parallelism and refine the task granularity. Experiments are conducted on the new version of the PaStiX solver, which has been completely rewritten to better integrate modern task-based runtime systems. The results demonstrate the increase in scalability achieved by the solver thanks to the adaptive task granularity provided by the StarPU runtime system.

This work was presented at the WAMTA 2024 workshop [23].

9 Bilateral contracts and grants with industry

9.1 Bilateral contracts with industry

Participants: Loris Marchal, Félix Wirth-Bonne.

- Contrat de collaboration entre la société Kog et l'équipe-projet ROMA pour le co-encadrement du stage de Félix Wirth (2200€).

10 Partnerships and cooperations

10.1 International initiatives

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

2024 was the thrid and last year of the CHALRESIL associate team. CHALRESIL stands for *Challenges in resilience at scale* and is operated between ROMA (PI Yves Robert) and the Innovative Computing Laboratory of the University of Tennessee Knoxville, USA (PI Thomas Herault). Many fundamental challenges in the resilience field have yet to be addressed, and CHALRESIL focused on some critical ones.

In 2024, we have focused on checkpointing strategies for fixed-size reservations on failure-prone platforms. Scheduling a job onto a computing platform typically involves making a series of reservations of the required resources. Long running applications, or applications whose total run-time are hard to predict, usually split their reservation in multiple smaller reservations and use checkpoint-restart to save intermediate steps of computation. There are multiple advantages to this approach, but the main one is that it lowers the wait-time of the application, as the job scheduler can easily place a smaller reservation. For each actual reservation, the job needs to be checkpointed before the reservation time has elapsed, otherwise the progress of the execution during the reservation will be lost. The checkpoint duration is a stochastic random variable that obeys some well-known (arbitrary) probability distribution law. This collaboration had led to a publication in the FTXS'23 workshop last year. We have extended the study to failure-prone platforms. Indeed, deciding when to checkpoint to protect from failures within a fixed-length reservation was an opne problem open, even for memoryless failure distributions. This problem is very important in practice, but it is much harder than the standard problem with fixed work instead of fixed time. This is because the optimal strategy will not be periodic. In other words, the fixed-work checkpointing problem, namely the optimal checkpoint strategy to minimize the expected time to execute a given amount of work is well-known, but the dual fixed-time checkpointing problem of maximizing the amount of work achieved within a given period of time seems to be very challenging. A survey of our results has been submitted for publication to the IJHPCA journal.

MODS

Title: Match and Order: improving direct solvers for cardiac simulations

Duration: 2023 ->

Coordinator: Johannes Langguth (langguth@simula.no)

Partners:

- Simula Research Laboratory (Norvège)

Inria contact: Grégoire Pichon until August 2024, and then Bora Uçar

Summary: The goal of the MODS project is to enhance robustness, scalability, and performance of sparse direct solvers by developing novel parallel matching and ordering algorithms. The results will be tested on and applied to simulations of cardiac electrophysiology developed by Simula. The partners Johannes Langguth, James Trotter, and Luk Burchard visited Lyon 15–19 January 2024; full time working at the offices in ENS de Lyon on 16–18 January. During this time, we have interfaced the approximate matching algorithm of the first year into iterative linear solvers within Matlab to see if as a preconditioners they were useful.

10.1.2 Participation in other International Programs

Collaboration with U. Chicago

Participants: Anne Benoit, Joachim Cendrier, Loris Marchal, Yves Robert, Frédéric Vivien.

- 2022–2024: FACCTS research collaboration with A. Chien at U. Chicago: Foundational Models and Efficient Algorithms for Scheduling with Variable Capacity Resources, funded by the France Chicago Center (see website).
- 2023–2024: Additional funding as part of our collaboration with A. Chien to organize two workshops on the topic Bridging Communities — Scheduling Variable Capacity Resources for Sustainability, in the U. Chicago center in Paris (2023-24). These workshops are funded by the International Institute of Research in Paris. The first workshop was organized in March 2023, see website.
- 2023–2026: U. Chicago-CNRS collaboration on efficient and environment-friendly scheduling and resource management algorithms at the edge: Funding secured for the PhD thesis of Joachim Cendrier.
- 2024–2026: FACCTS research collaboration with Ian Foster on Smart Scheduling in Serverless Environments: Unveiling the Role of Data in Performance and Resilience, funded by the France Chicago Center (see website).

10.2 International research visitors

10.2.1 Visits of international scientists

Inria International Chair

Participants: Julien Langou.

Julien Langou, professor at the University Denver (USA) has been awarded an Inria International Chair to visit the ROMA team in the period 2023–2026. He spent 4.5 months in the team in May-July 2024 and starting mid-November 2024 to start collaborations with researchers in ROMA.

Other international visits to the team

Rob Bisseling

Status Visiting research scientist.

Institution of origin: Utrecht University.

Country: The Netherlands.

Dates: February 2024–May 2024.

Context of the visit: R. Bisseling and B. Uçar have collaborated on optimal partitioning of finite element meshes. A theoretical work is conducted, resulting in certificates for the optimality of a previous heuristic by Uçar and colleagues on special instances. The theoretical analysis also led to heuristics to speed-up integer linear programs for the problem at hand, reducing the run time from about 4 weeks to under a minute in the initial problem.

Mobility program/type of mobility: LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, ENS de Lyon, and Guest Research Fellowship at the Collegium – Institut d’études avancées de Lyon.

Henrik Reinstädtler

Status Visiting PhD student.

Institution of origin: Heidelberg University.

Country: Germany.

Dates: 9 September – 4 October 2024.

Context of the visit: H. Reinstädtler pursues a PhD degree at the Heidelberg University, in Prof. Dr. C. Schulz group. We are working on b-matching problems in hypergraphs.

Mobility program/type of mobility: Funding from Prof. Dr. C. Schulz.

10.2.2 Visits to international teams

Research stays abroad

Loris Marchal

Visited institution: École Supérieure de Technologie, Montréal

Country: Canada

Dates: 08–12/2024

Context of the visit: temporary assignement at the ILLS laboratory (CNRS, McGill, ETS, MILA).

Mobility program/type of mobility: sabbatical

10.3 National initiatives

10.3.1 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.

10.4 Regional initiatives

- Bora Uçar has received the support for inter-laboratories projects program of the Fédération Informatique de Lyon (FIL) 2024 for the project Gabi, with Jérémy E. Cohen of the Myriads Team, CREATIS.

11 Dissemination

11.1 Promoting scientific activities

17th Workshop on Scheduling for Large Scale Systems: Anne Benoit and Yves Robert have organized the 17th Workshop on Scheduling for Large Scale Systems in Aussois in June 2024. Further details can be found on the workshop webpage.

11.1.1 Scientific events: selection

Chair of conference program committees

- Anne Benoit was co-chair for Track 2: "Scheduling, Resource Management, Cloud, Edge Computing, and Workflows" of the 30th Int. European Conf. on Parallel and Distributed Computing (EuroPar 2024), Madrid, Spain, August 26-30, 2024.

Member of conference program committees

- Anne Benoit was a member of the program committees of IEEE BigData'24, PPAM'24, SC'24 BoFs, and SC'24 Workshops.
- Suraj Kumar was a member of the program committee of SC'24.
- Loris Marchal was a member of the program committee of EuroPar'24.
- Yves Robert was a member of the program committees of FTXS'24, SCALA'24, SuperCheck'24 (all co-located with SC'24) and Resilience (co-located with Euro-Par'24).
- Bora Uçar was a member of the programme committees of HiPC2024, BASARIM 2024, IPDPS 2024, SIAM PP 2024, and HPC Asia 2024.
- Frédéric Vivien was a member of the program committees of EuroPar'24, HPDC 2024, ICPP 2024, and IPDPS'24.

Reviewer

- Bora Uçar has reviewed papers for ALENEX2025 and SODA2024.

11.1.2 Journal

Member of the editorial boards

- Anne Benoit is Editor in Chief of ParCo, the journal of Parallel Computing: Systems and Applications, and she is also a member of the editorial board of ACM TOPC (Transactions on Parallel Computing).
- 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).
- Bora Uçar is a member of the editorial board of IEEE Transactions on Parallel and Distributed System, the journal of Parallel Computing, SIAM Journal on Scientific Computing (SISC), and SIAM Journal on Matrix Analysis and Applications (SIMAX).
- Frédéric Vivien is a member of the editorial board of the Journal of Parallel and Distributed Computing.

Reviewer - reviewing activities

- Suraj Kumar has reviewed manuscripts for IEEE Transactions on Parallel and Distributed Systems journal.

11.1.3 Leadership within the scientific community

- Anne Benoit was the chair of IEEE Technical Community on Parallel Processing (TCPP) until June 2024.
- Bora Uçar was the program director of SIAM Activity Group on Applied and Discrete Algorithms (2023–2024).
- Bora Uçar was the Chair of the 2024 IEEE TCPP Outstanding Service and Contributions Award Committee.
- Yves Robert was the Chair of the 2024 IEE Charles Babbage Award Committee.

11.1.4 Scientific expertise

- Frédéric Vivien was an elected member of the scientific council of the École normale supérieure de Lyon until July 2024.
- Frédéric Vivien is an elected member of INRIA *commision d'évaluation*.
- Frédéric Vivien is a member of the scientific council of the IRMIA labex.

11.2 Teaching - Supervision - Juries

11.2.1 Teaching

- Yves Robert, Chair of the Computer Science department at ENS Lyon, France, from Oct 1, 2023 up to Sep 30, 2024
- Master: Anne Benoit, Parallel and Distributed Algorithms and Programs, 42h, M1, ENS Lyon, France
- Master: Suraj Kumar, Data-aware algorithms for matrix and tensor computations, 30h, M2, ENS Lyon, France.
- Master: Grégoire Pichon, Bibliographie, étude de cas, projet, certifications, 12h, M2, Univ. Lyon 1, France
- Master: Grégoire Pichon, Compilation / traduction des programmes, 22.5h, M1, Univ. Lyon 1, France
- Master: Grégoire Pichon, Systèmes avancés, 19.5h, M1, Univ. Lyon 1, France
- Master: Grégoire Pichon, Réseaux, 12h, M1, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Programmation concurrente, 33h, L3, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Réseaux, 36h, L3, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Système d'exploitation, 27h, L2, Univ. Lyon 1, France
- Licence: Grégoire Pichon, Référent pédagogique, 30h, L1/L2/L3, Univ. Lyon 1, France
- Agrégation Informatique: Yves Robert, Algorithmique, NP-complétude et algorithmes d'approximation, probabilités, graphes, structures de données, 75h, ENS Lyon, France
- Master: Frédéric Vivien, Parallel algorithms, 10h, M1/M2, ECNU, Shanghaï, Chine, 2023 (remote teaching).

11.2.2 Supervision

- Anne Benoit and Frédéric Vivien are co-supervising the thesis of Joachim Cendrier. Joachim works on efficient and environment-friendly scheduling and resource management algorithms at the edge, in collaboration with Andrew Chien from U. Chicago. He is funded by a CNRS - U. Chicago project.
- Anne Benoit is co-supervising the thesis of Svetlana Kulagina with Henning Meyerhenke (Humboldt-Universitat zu Berlin, Germany), as part of the FONDA project, on the execution of large workflows in heterogeneous execution environments.

11.2.3 Juries

- Anne Benoit was a member of the jury for hiring an Associate Professor at Toulouse INP.
- Loris Marchal has reviewed applications for the discovery grants of the Natural Sciences and Engineering Research Council of Canada.
- Bora Uçar was a member of the prize committee of 2024 SIAM Student Paper Prize.
- Bora Uçar was a member of the prize committee of 2024 SIAM's George Polya Prize in Applied Combinatorics.
- Frédéric Vivien was a member of the jury for hiring INRIA CRCN and ISFP researchers for the Centre Inria de l'Université de Lille.
- Frédéric Vivien was a member of the jury for hiring INRIA CRCN-TH researchers.

11.3 Popularization

11.3.1 Participation in Live events

Anne Benoit and Nathalie Revol went twice to the primary school Guilloux (Saint-Genis-Laval), during 2 hours, to introduce computer science (programming and robotics) for two classes of 9-10 years old pupils.

12 Scientific production

12.1 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] A. Benoit, L. Perotin, Y. Robert and F. Vivien. 'Checkpointing strategies to tolerate non-memoryless failures on HPC platforms'. In: *ACM Transactions on Parallel Computing* (Sept. 2023). DOI: 10.1145/3624560. URL: https://inria.hal.science/hal-04215283.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

12.2 Publications of the year

International journals

[8] L. Bautista-Gomez, A. Benoit, S. Di, T. Herault, Y. Robert and H. Sun. ‘A survey on checkpointing strategies: Should we always checkpoint à la Young/Daly?’ In: *Future Generation Computer Systems* 161 (Dec. 2024), pp. 315–328. DOI: 10.1016/j.future.2024.07.022. URL: https://inria.hal.science/hal-04767137 (cit. on p. 8).

[9] S. Ben Mokhtar, L.-C. Canon, A. Dugois, L. Marchal and E. Rivière. ‘A scheduling framework for distributed key-value stores and its application to tail latency minimization’. In: *Journal of Scheduling* (26th Feb. 2024). DOI: 10.1007/s10951-023-00803-8. URL: https://hal.science/hal-04501444 (cit. on p. 11).

[10] A. Benoit, T. Herault, L. Perotin, Y. Robert and F. Vivien. ‘Revisiting I/O bandwidth-sharing strategies for HPC applications’. In: *Journal of Parallel and Distributed Computing* 188 (June 2024). DOI: 10.1016/j.jpdc.2024.104863. URL: https://inria.hal.science/hal-04714098 (cit. on p. 9).

[11] A. Benoit, L. Perotin, Y. Robert and F. Vivien. ‘Checkpointing strategies to tolerate non-memoryless failures on HPC platforms’. In: *ACM Transactions on Parallel Computing* 11.1 (Mar. 2024), pp. 1–26. DOI: 10.1145/3624560. URL: https://inria.hal.science/hal-04215283 (cit. on p. 8).

[12] F. Cappello, S. Di, R. Underwood, D. Tao, J. Calhoun, Y. Kazutomo, K. Sato, A. Singh, L. Giraud, E. Agullo, X. Yepes, M. Acosta, S. Jin, J. Tian, F. Vivien, B. Zhang, K. Sano, T. Ueno, T. Grützmacher and H. Anzt. ‘Multifacets of lossy compression for scientific data in the Joint-Laboratory of Extreme Scale Computing’. In: *Future Generation Computer Systems* (13th June 2024). DOI: 10.1016/j.future.2024.05.022. URL: https://inria.hal.science/hal-04618060 (cit. on p. 10).

[13] H. A. Daas, G. Ballard, L. Grigori, S. Kumar and K. Rouse. ‘Communication Lower Bounds and Optimal Algorithms for Multiple Tensor-Times-Matrix Computation’. In: *SIAM Journal on Matrix Analysis and Applications* 45.1 (6th Feb. 2024), pp. 450–477. DOI: 10.1137/22M1510443. URL: https://inria.hal.science/hal-03950359 (cit. on p. 13).

[14] Y. Du, L. Marchal, G. Pallez and Y. Robert. ‘Improving Batch Schedulers with Node Stealing for Failed Jobs’. In: *Concurrency and Computation: Practice and Experience* 36.12 (2024), pp. 1–36. DOI: 10.1002/cpe.8043. URL: https://inria.hal.science/hal-03643403 (cit. on p. 9).

[15] Y. Gao, L. Han, J. Liu, Y. Robert and F. Vivien. ‘Minimizing Energy Consumption for Real-Time Tasks on Heterogeneous Platforms Under Deadline and Reliability Constraints’. In: *Algorithmica* 86.10 (22nd July 2024), pp. 3079–3114. DOI: 10.1007/s00453-024-01253-0. URL: https://inria.hal.science/hal-04715054 (cit. on p. 7).

International peer-reviewed conferences

[16] A. Benoit, L. Perotin, Y. Robert and F. Vivien. ‘Checkpointing strategies for a fixed-length execution’. In: 14th Workshop on Fault-Tolerance for HPC at eXtreme Scale (FTXS 2024). Atlanta, United States, 22nd Nov. 2024. URL: https://inria.hal.science/hal-04714372 (cit. on p. 8).

[17] L.-C. Canon, A. Dugois and L. Marchal. ‘Solving the Restricted Assignment Problem to Schedule Multi-get Requests in Key-Value Stores’. In: 30th European Conference on Parallel and Distributed Processing. Vol. 14801. Lecture Notes in Computer Science. Madrid, Spain: Springer Nature Switzerland, 26th Aug. 2024, pp. 195–209. DOI: 10.1007/978-3-031-69577-3_14. URL: https://hal.science/hal-04784268 (cit. on p. 11).

[18] L. Eyraud-Dubois, G. Iooss, J. Langou and F. Rastello. ‘Tightening I/O Lower Bounds through the Hourglass Dependency Pattern’. In: SPAA 2024 - 36th ACM Symposium on Parallelism in Algorithms and Architectures. Nantes, France, Apr. 2024, pp. 1–34. URL: https://inria.hal.science/hal-04555744 (cit. on p. 14).

[19] M. Gonthier, E. Larsson, L. Marchal, C. Nettelblad and S. Thibault. ‘Data-Driven Locality-Aware Batch Scheduling’. In: APDCM 2024 - 26th Workshop on Advances in Parallel and Distributed Computational Models. San Francisco, United States, 27th May 2024. URL: https://inria.hal.science/hal-04500281 (cit. on p. 11).

[20] S. Jin, S. Di, F. Vivien, D. Wang, Y. Robert, D. Tao and F. Cappello. 'Concealing Compression-accelerated I/O for HPC Applications through In Situ Task Scheduling'. In: EuroSys 2024. Athens, Greece, 22nd Apr. 2024. URL: https://inria.hal.science/hal-04225758 (cit. on p. 10).

[21] S. Kulagina, H. Meyerhenke and A. Benoit. 'Mapping Large Memory-constrained Workflows onto Heterogeneous Platforms'. In: ICPP 2024 - 53rd International Conference on Parallel Processing. Gotland, Sweden: ACM, 2024, pp. 305–316. DOI: 10.1145/3673038.3673068. URL: https://inria.hal.science/hal-04767107 (cit. on p. 10).

[22] D. Lesens, J. E. Cohen and B. Uçar. 'Orthogonal matching pursuit-based algorithms for the Birkhoff-von Neumann decomposition'. In: EUSIPCO 2024 - 24th European Signal Processing Conference. Lyon, France, Mar. 2024, p. 12. URL: https://hal.science/hal-04500014 (cit. on p. 12).

[23] A. Lisito, M. Faverge, G. Pichon and P. Ramet. 'Enhancing sparse direct solver scalability through runtime system automatic data partition'. In: WAMTA 2024 - Workshop on Asynchronous Many-Task Systems and Applications 2024. Vol. 14626. Lecture Notes in Computer Science. Knoxville, United States: Springer Nature Switzerland, 14th Mar. 2024, pp. 105–110. DOI: 10.1007/978-3-031-61763-8_10. URL: https://inria.hal.science/hal-04527103 (cit. on p. 14).

[24] H. Reinstädtler, C. Schulz and B. Uçar. 'Engineering Edge Orientation Algorithms'. In: *In 32nd Annual European Symposium on Algorithms (ESA 2024). Leibniz International Proceedings in Informatics (LIPIcs),* ESA 2024 - 32nd Annual European Symposium on Algorithms. Egham, United Kingdom: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 23rd Sept. 2024, pp. 1–21. DOI: 10.4230/LIPIcs.ESA.2024.97. URL: https://hal.science/hal-04555599 (cit. on p. 12).

Reports & preprints

[25] A. Benoit, K. Kaya and B. Uçar. *Connectivity of a random directed graph model.* RR-9540. Inria Lyon, Jan. 2024. URL: https://inria.hal.science/hal-04428417 (cit. on p. 13).

[26] A. Benoit, L. Perotin, Y. Robert and F. Vivien. *Checkpointing strategies for a fixed-length execution.* RR-9552. Inria, July 2024, p. 26. URL: https://inria.hal.science/hal-04668191.

[27] L.-C. Canon, A. Dugois and L. Marchal. *Solving the Restricted Assignment Problem to Schedule Multi-Get Requests in Key-Value Stores (extended version).* 22nd Mar. 2024. URL: https://hal.science/hal-04516752.

[28] H. A. Daas, G. Ballard, L. Grigori, S. Kumar, K. Rouse and M. Verite. *Communication Lower Bounds and Optimal Algorithms for Symmetric Matrix Computations.* 17th Sept. 2024. URL: https://inria.hal.science/hal-04701302 (cit. on p. 14).

[29] M. Gonthier, S. Thibault and L. Marchal. *A generic scheduler to foster data locality for GPU and out-of-core task-based applications.* 13th Sept. 2024. URL: https://inria.hal.science/hal-04146714.

[30] D. Lesens, J. E. Cohen and B. Uçar. *Algorithms for symmetric Birkhoff-von Neumann decomposition of symmetric doubly stochastic matrices.* RR-9566. Inria Lyon, Jan. 2025, p. 30. URL: https://inria.hal.science/hal-04877502.

[31] S. Singh and B. Uçar. *Efficient parallel sparse tensor contraction.* RR-9551. Inria Lyon, July 2024. URL: https://hal.science/hal-04659658 (cit. on p. 12).

[32] A. Tremodeux, E. Agullo, A. Benoit, L. Giraud, T. Herault and Y. Robert. *Fault-tolerant numerical iterative algorithms at scale.* RR-9567. Inria Lyon, Jan. 2025. URL: https://inria.hal.science/hal-04872041.