IN PARTNERSHIP WITH:
CNRS

Université Claude Bernard (Lyon 1)

Ecole normale supérieure de Lyon

Activity Report 2012

Team ROMA

Optimisation des ressources : modèles, algorithmes et ordonnancement

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

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Distributed and High Performance Computing

Table of contents

Team ROMA

Keywords: Scheduling, Parallel And Distributed Algorithms, Combinatorial Optimization, Exascale Systems, Fault Tolerance

The ROMA *team is common to CNRS, ENS Lyon, UCBL, and Inria. This team is part of the Laboratoire de l'Informatique du Parallélisme (LIP), UMR ENS Lyon/CNRS/Inria/UCBL 5668. The team is located at the École normale supérieure de Lyon. The external collaborators are members of the APO team of the IRIT laboratory (UMR 5505), located at the ENSEEIHT site of IRIT.*

Creation of the Team: February 01, 2012 .

1. Members

Research Scientists
Jean-Yves L'Excellent [Junior Researcher (CR), HdR]
Loris Marchal [Junior Researcher (CR)]
Bora Uçar [Junior Researcher (CR)]
Frédéric Vivien [Team Leader, Senior Researcher (DR), HdR]

Faculty Members
Anne Benoit [Associate Professor (MCF), IUF junior member, HdR]
Yves Robert [Professor, IUF senior member, HdR]

External Collaborators
Patrick Amestoy [Professor, HdR]
Alfredo Buttari [Junior Researcher (CR)]
François-Henry Rouet [PhD student, until November 30, 2012]
Clément Weisbecker [PhD student]

PhD Students
Guillaume Aupy [ENS grant]
Paul Renaud-Goud [MENRT grant, until September 30, 2012]
Mohamed Sid-Lakhdar [MENRT grant]
Julien Herrmann [ENS grant, HdR]

Post-Doctoral Fellows
Amina Guermouche [Until December 11, 2012]
Johannes Langguth [Until September 30, 2012]
Fanny Dufossé [Until August 31, 2012]

Visiting Scientist
Oliver Sinnen [April-June, 2012]

Administrative Assistant
Evelyne Blesle [Inria, 50% on the team]

2. Overall Objectives

2.1. Overall Objectives

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

Modern computing platforms provide huge amounts of computational power—the top supercomputers contain more than 100,000 cores, and volunteer computing grids gather millions of processors. Squeezing the most out of these platforms could enable scientists to solve problems that remain currently beyond reach. However, to reach such a goal, all platform resources must be efficiently used: computational units, communication capabilities, memory hierarchies, energy, etc. Such resource optimizations are quite difficult because modern platforms have new, and hard to manage, characteristics: they contain multicore processors and sometimes specialized processors such as GPGPUs (General Purpose - Graphical Processing Units); they may be distributed on a very large scale, which can significantly impact communications; they may be volatile and even unreliable; and their usage may be subject to conflicting objectives from the platform owner(s) and users. Therefore, harnessing the full power of modern computing platforms requires careful theoretical algorithmic studies of resource optimization problems. The goal of the ROMA project is to perform such studies and to design efficient practical scheduling strategies and resource allocation algorithms.

Historically, the ROMA team comes from the merging of two of the three groups that composed the GRAAL project-team: (i) the group focusing on fundamental research on scheduling strategies and algorithm design for heterogeneous platforms; and (ii) the group working on direct solvers for sparse linear systems. The ROMA project is organized around two main research themes —that are relevant to the focus of the former groups— and four transverse topics.

The two main research themes are:

- Static algorithms for dynamic environments
- Direct solvers for sparse linear systems

The four transverse topics are:

- Memory-aware algorithms
- Linear algebra on post-petascale multicore platforms
- Multi-criteria optimization
- Combinatorial scientific computing

3. Application Domains

3.1. Application of sparse direct solvers

Sparse direct (multifrontal) solvers in distributed-memory environments have a wide range of applications as they are used at the heart of many numerical methods in simulation: 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 linear system of equations involving sparse matrices. There are therefore a number of application fields, among which some of the ones cited by the users of our sparse direct solver MUMPS (see Section 4.1) are: structural mechanics, biomechanics, medical image processing, tomography, geophysics, ad-hoc networking modeling (e.g., Markovian processes), electromagnetics, fluid dynamics, econometric models, oil reservoir simulation, magneto-hydro-dynamics, chemistry, acoustics, glaciology, astrophysics, circuit simulation.

4. Software

4.1. MUMPS

Participants: Patrick Amestoy, Alfredo Buttari, Jean-Yves L'Excellent [correspondent], Mohamed Sid-Lakhdar, François-Henry Rouet, Bora Uçar, Clément Weisbecker.

MUMPS (for *MUltifrontal Massively Parallel Solver*, see http://graal.ens-lyon.fr/MUMPS) is a software package for the solution of large sparse systems of linear equations. The development of MUMPS was initiated by the European project PARASOL (Esprit 4, LTR project 20160, 1996-1999), whose results and developments were public domain. Since then, MUMPS has been supported by CERFACS, CNRS, ENS Lyon, INPT(ENSEEIHT)-IRIT (main contributor), Inria, and University of Bordeaux. In the context of an ADT project (Action of Technological Development), Maurice Brémond ("SED" service) also works part-time on MUMPS.

MUMPS implements a direct method, the multifrontal method; it is a parallel code capable of exploiting distributed-memory computers; its main originalities are its numerical robustness and the wide range of functionalities available.

The latest release is MUMPS 4.10.0 (May 2011).

More information on MUMPS is available at http://graal.ens-lyon.fr/MUMPS/ and http://mumps.enseeiht.fr.

5. New Results

5.1. Unified model for assessing checkpointing protocols at extreme-scale

In this work [38], we defined a unified model for several well-known checkpoint/restart protocols. The proposed model is generic enough to encompass both extremes of the checkpoint/restart space, from coordinated approaches to a variety of uncoordinated checkpoint strategies (with message logging). We identified a set of crucial parameters, instantiated them and compared the expected efficiency of the fault tolerant protocols, for a given application/platform pair. We then proposed a detailed analysis of several scenarios, including some of the most powerful currently available HPC platforms, as well as anticipated Exascale designs. The results of this analytical comparison are corroborated by a comprehensive set of simulations. Altogether, they outlined comparative behaviors of checkpoint strategies at very large scale, thereby providing insight that is hardly accessible to direct experimentation.

5.2. Impact of fault prediction on checkpointing strategies

We dealt [34] with the impact of fault prediction techniques on checkpointing strategies. We extended the classical analysis of Young and Daly in the presence of a fault prediction system, which is characterized by its recall and its precision, and which provides either exact or window-based time predictions. We succeeded in deriving the optimal value of the checkpointing period (thereby minimizing the waste of resource usage due to checkpoint overhead) in all scenarios. These results allow to analytically assess the key parameters that impact the performance of fault predictors at very large scale. In addition, the results of this analytical evaluation were nicely corroborated by a comprehensive set of simulations, thereby demonstrating the validity of the model and the accuracy of the results.

5.3. Combining process replication and checkpointing for resilience on exascale systems

Processor failures in post-petascale settings are common occurrences. The traditional fault-tolerance solution, checkpoint-rollback, severely limits parallel efficiency. One solution is to replicate application processes so that a processor failure does not necessarily imply an application failure. Process replication, combined with checkpoint-rollback, has been recently advocated by Ferreira et al. [52]. We first identified [41] an incorrect analogy made in their work between process replication and the birthday problem, and derived correct values for the Mean Number of Failures To Interruption and Mean Time To Interruption for exponentially distributed failures. We then extended these results to arbitrary failure distributions, including closed-form solutions for Weibull distributions. Finally, we evaluated process replication using both synthetic and real-world failure traces. Our main findings are: (i) replication is less beneficial than claimed by Ferreira et al.; (ii) although the choice of the checkpointing period can have a high impact on application execution in the no-replication case, with process replication this choice is no longer critical.

5.4. On the complexity of scheduling checkpoints for computational workflows

This work [22] dealt with the complexity of scheduling computational workflows in the presence of Exponential failures. When such a failure occurs, rollback and recovery is used so that the execution can resume from the last checkpointed state. The goal is to minimize the expected execution time, and we have to decide in which order to execute the tasks, and whether to checkpoint or not after the completion of each given task. We showed that this scheduling problem is strongly NP-complete, and proposed a (polynomial-time) dynamic programming algorithm for the case where the application graph is a linear chain. These results laid the theoretical foundations of the problem, and constituted a prerequisite before discussing scheduling strategies for arbitrary DAGS of moldable tasks subject to general failure distributions.

5.5. Scheduling tree-shaped task graphs to minimize memory and makespan

We [44] investigated the execution of tree-shaped task graphs using multiple processors. Each edge of such a tree represents a large IO file. A task can only be executed if all input and output files fit into memory, and a file can only be removed from memory after it has been consumed. Such trees arise, for instance, in the multifrontal method of sparse matrix factorization. The maximum amount of memory needed depends on the execution order of the tasks. With one processor the objective of the tree traversal is to minimize the required memory. This problem was well studied and optimal polynomial algorithms were proposed. We extended the problem by considering multiple processors, which is of obvious interest in the application area of matrix factorization. With the multiple processors comes the additional objective to minimize the time needed to traverse the tree, i.e., to minimize the makespan. Not surprisingly, this problem proves to be much harder than the sequential one. We studied the computational complexity of this problem and provided an inapproximability result even for unit weight trees. We proposed several heuristics, each with a different optimization focus, and we analyzed them in an extensive experimental evaluation using realistic trees.

5.6. Memory allocation for different classes of DAGs

In this work, we studied the complexity of traversing workflows whose tasks require large I/O files. Such workflows arise in many scientific fields, such as image processing, genomics or geophysical simulations. They usually exhibit some regularity, and most of them can be modeled as Series-Parallel Graph. We target a classical two-level memory system, where the main memory is faster but smaller than the secondary memory. A task in the workflow can be processed if all its predecessors have been processed, and if its input and output files fit in the currently available main memory. The amount of available memory at a given time depends upon the ordering in which the tasks are executed. We focus on the problem of minimizing the amount of main memory needed to process the whole DAG.

We first concentrate on the parallel composition of task chains, or fork-join graphs. We adapt an algorithm designed for trees by Liu [54]. We prove that an optimal schedule for fork-join can be split in two optimal tree schedules, which are obtained using Liu's algorithm. We then move to Series-Parallel graphs and propose a recursive adaptation of the previous algorithm, which consists in serializing every parallel compositions, starting from the innermost, using the fork-join algorithm. Simulations show that this algorithm always reach the optimal performance, and we provide a sketch of the optimality proof. We also study compositions of complete bipartite graphs, which are another important class of DAGs arising in scientific workflows. We propose an optimal algorithm for a class of compositions which we name tower of complete bipartite graphs.

5.7. Scheduling non-linear divisible loads

Divisible Load Theory (DLT) has received a lot of attention in the past decade. A divisible load is a perfect parallel task, that can be split arbitrarily and executed in parallel on a set of possibly heterogeneous resources. The success of DLT is strongly related to the existence of many optimal resource allocation and scheduling algorithms, what strongly differs from general scheduling theory. Moreover, recently, close relationships have been underlined between DLT, that provides a fruitful theoretical framework for scheduling jobs on heterogeneous platforms, and MapReduce, that provides a simple and efficient programming framework to deploy applications on large scale distributed platforms.

The success of both have suggested to extend their framework to non-linear complexity tasks. We show [35] that both DLT and MapReduce are better suited to workloads with linear complexity. In particular, we prove that divisible load theory cannot directly be applied to quadratic workloads, such as it has been proposed recently. We precisely state the limits for classical DLT studies and we review and propose solutions based on a careful preparation of the dataset and clever data partitioning algorithms. In particular, through simulations, we show the possible impact of this approach on the volume of communications generated by MapReduce, in the context of Matrix Multiplication and Outer Product algorithms.

5.8. Energy-aware scheduling under reliability and makespan constraints

We consider [13] a task graph mapped on a set of homogeneous processors. We aim at minimizing the energy consumption while enforcing two constraints: a prescribed bound on the execution time (or makespan), and a reliability threshold. Dynamic voltage and frequency scaling (DVFS) is an approach frequently used to reduce the energy consumption of a schedule, but slowing down the execution of a task to save energy is decreasing the reliability of the execution.

In this work, to improve the reliability of a schedule while reducing the energy consumption, we allow for the re-execution of some tasks. We assess the complexity of the tri-criteria scheduling problem (makespan, reliability, energy) of deciding which task to re-execute, and at which speed each execution of a task should be done, with two different speed models: either processors can have arbitrary speeds (continuous model), or a processor can run at a finite number of different speeds and change its speed during a computation (VDD model). We propose several novel tri-criteria scheduling heuristics under the continuous speed model, and we evaluate them through a set of simulations. The two best heuristics turn out to be very efficient and complementary.

5.9. Approximation algorithms for energy, reliability and makespan optimization problems

We consider [32] the problem of scheduling an application on a parallel computational platform. The application is a particular task graph, either a linear chain of tasks, or a set of independent tasks. The platform is made of identical processors, whose speed can be dynamically modified. It is also subject to failures: if a processor is slowed down to decrease the energy consumption, it has a higher chance to fail. Therefore, the scheduling problem requires to re-execute or replicate tasks (i.e., execute twice a same task, either on the same processor, or on two distinct processors), in order to increase the reliability. It is a tri-criteria problem: the goal is to minimize the energy consumption, while enforcing a bound on the total execution time (the makespan), and a constraint on the reliability of each task.

Our main contribution is to propose approximation algorithms for these particular classes of task graphs. For linear chains, we design a fully polynomial time approximation scheme. However, we show that there exists no constant factor approximation algorithm for independent tasks, unless P=NP, and we are able in this case to propose an approximation algorithm with a relaxation on the makespan constraint.

5.10. Optimal algorithms and approximation algorithms for replica placement with distance constraints in tree networks

We study [16] the problem of replica placement in tree networks subject to server capacity and distance constraints. The client requests are known beforehand, while the number and location of the servers are to be determined. The Single policy enforces that all requests of a client are served by a single server in the tree, while in the Multiple policy, the requests of a given client can be processed by multiple servers, thus distributing the processing of requests over the platform. For the Single policy, we prove that all instances of the problem are NP-hard, and we propose approximation algorithms. The problem with the Multiple policy was known to be NP-hard with distance constraints, but we provide a polynomial time optimal algorithm to solve the problem in the particular case of binary trees when no request exceeds the server capacity.

5.11. Throughput optimization for pipeline workflow scheduling with setup times

We tackle [15] pipeline workflow applications that are executed on a distributed platform with setup times. In such applications, several computation stages are interconnected as a linear application graph, and each stage holds a buffer of limited size where intermediate results are stored and a processor setup time occurs when passing from one stage to another. The considered stage/processor mapping strategy is based on interval mappings, where an interval of consecutive stages is performed by the same processor and the objective is the throughput optimization. Typical examples for this kind of applications are streaming applications such as audio and video coding or decoding, image processing using co-processing devices as FPGA. Even when neglecting setup times, the problem is NP-hard on heterogeneous platforms and we therefore restrict to homogeneous resources. We provide an optimal algorithm for constellations with identical buffer capacities. When buffer sizes are not fixed, we deal with the problem of allocating the buffers in shared memory and present a $b/(b+1)$-approximation algorithm.

5.12. Semi-matching algorithms for scheduling parallel tasks under resource constraints

We study [37] the problem of minimum makespan scheduling when tasks are restricted to subsets of the processors (resource constraints), and require either one or multiple distinct processors to be executed (parallel tasks). This problem is related to the minimum makespan scheduling problem on unrelated machines, as well as to the concurrent job shop problem, and it amounts to finding a semi-matching in bipartite graphs or hypergraphs. While the problem was known to be NP-complete for bipartite graphs, but solvable in polynomial time for unweighted graphs (i.e., unit tasks), we prove that the problem is NP-complete for hypergraphs even in the unweighted case. We design several greedy algorithms of low complexity to solve two versions of the problem, and assess their performance through a set of exhaustive simulations. Even though there is no approximation guarantee on these linear algorithms, they return solutions close to the optimal (or a known lower bound) in average.

5.13. A Symmetry preserving algorithm for matrix scaling

We present an iterative algorithm which asymptotically scales the ∞-norm of each row and each column of a matrix to one. This scaling algorithm preserves symmetry of the original matrix and shows fast linear convergence with an asymptotic rate of $1/2$. We discuss extensions of the algorithm to the one-norm, and by inference to other norms. For the 1-norm case, we show again that convergence is linear, with the rate dependent on the spectrum of the scaled matrix. We demonstrate experimentally that the scaling algorithm improves the conditioning of the matrix and that it helps direct solvers by reducing the need for pivoting. In particular, for symmetric matrices the theoretical and experimental results highlight the potential of the proposed algorithm over existing alternatives. This work resulted in an improved version [43] of an earlier technical report [55].

5.14. On shared-memory parallelization of a sparse matrix scaling algorithm

We discuss [25] efficient shared memory parallelization of sparse matrix computations whose main traits resemble to those of the sparse matrix-vector multiply operation. Such computations are difficult to parallelize because of the relatively small computational granularity characterized by small number of operations per each data access. Our main application is a sparse matrix scaling algorithm which is more memory bound than the sparse matrix vector multiplication operation. We take the application and parallelize it using the standard OpenMP programming principles. Apart from the common race condition avoiding constructs, we do not reorganize the algorithm. Rather, we identify associated performance metrics and describe models to optimize them. By using these models, we implement parallel matrix scaling algorithms for two well-known sparse matrix storage formats. Experimental results show that simple parallelization attempts which leave data/work

partitioning to the runtime scheduler can suffer from the overhead of avoiding race conditions especially when the number of threads increases. The proposed algorithms perform better than these algorithms by optimizing the identified performance metrics and reducing the overhead.

5.15. Investigations on push-relabel based algorithms for the maximum transversal problem

In a technical report [42], we investigate the push-relabel algorithm for solving the problem of finding a maximum cardinality matching in a bipartite graph in the context of the maximum transversal problem. We describe in detail an optimized yet easy-to-implement version of the algorithm and fine-tune its parameters. We also introduce new performance-enhancing techniques. On a wide range of real-world instances, we compare the push-relabel algorithm with state-of-the-art augmenting path-based algorithms and the recently proposed pseudoflow approach. We conclude that a carefully tuned push-relabel algorithm is competitive with all known augmenting path-based algorithms, and superior to the pseudoflow-based ones. We finalized this work by reporting the most important results in a journal article [9].

5.16. On optimal and balanced sparse matrix partitioning problems

We investigate [20] one dimensional partitioning of sparse matrices under a given ordering of the rows/columns. The partitioning constraint is to have load balance across processors when different parts are assigned to different processors. The load is defined as the number of rows, or columns, or the nonzeros assigned to a processor. The partitioning objective is to optimize different functions, including the well-known total communication volume arising in a distributed memory implementation of parallel sparse matrix-vector multiplication operations. The difference between our problem in this work and the general sparse matrix partitioning problem is that the parts should correspond to disjoint intervals of the given order. Whereas the partitioning problem without the interval constraint corresponds to the NP-complete hypergraph partitioning problem, the restricted problem corresponds to a polynomial-time solvable variant of the hypergraph partitioning problem. We adapt an existing dynamic programming algorithm designed for graphs to solve two related partitioning problems in graphs. We then propose graph models for a given hypergraph and a partitioning objective function so that the standard cutsize definition in the graph model exactly corresponds to the hypergraph partitioning objective function. In extensive experiments, we show that our proposed algorithm is helpful in practice. It even demonstrates performance superior to the standard hypergraph partitioners when the number of parts is high.

5.17. Constructing elimination trees for sparse unsymmetric matrices

The elimination tree model for sparse unsymmetric matrices and an algorithm for constructing it have been recently proposed [50], [51]. The construction algorithm has a worst-case time complexity of $\Theta(mn)$ for an $n \times n$ unsymmetric matrix having m off-diagonal nonzeros. We proposed [53] another algorithm that has a worst-case time complexity of $\mathcal{O}(m \log n)$. During this reporting period, we compared the two algorithms experimentally and showed that both algorithms are efficient in general. The known algorithm [51] is faster in many practical cases, yet there are instances in which there is a significant difference between the running time of the two algorithms in favor of the proposed one.

5.18. Introduction of shared memory parallelism in a distributed-memory sparse multifrontal solver

We study the adaptation of a parallel distributed-memory solver, MUMPS, into a shared-memory code, targetting multicore architectures. An advantage of adapting the code rather than starting with a new design is to fully benefit from its numerical kernels and functionalities. We show how one can take advantage of OpenMP directives and of existing libraries optimized for shared-memory environments, in our case BLAS libraries [48]. We have also started to study approaches that take advantage of the specificities of NUMA architectures.

5.19. Improving multifrontal methods by means of low-Rank representations

Matrices coming from elliptic PDEs have been shown to have a low-rank property. Although the dense internal datastructures involved in a multifrontal method, the so-called frontal matrices or fronts, are full-rank, their off-diagonal blocks can then be approximated by low-rank products. We have studied a low-rank format called Block Low Rank and explained how it can be used to reduce the memory footprint and complexity of both the factorization and solve phases, depending on the way variables are grouped. The proposed approach can be used either to accelerate the factorization and solution phases or to build a preconditioner [47]. We have started the development of a version of MUMPS that exploits such properties. This work is in collaboration with EDF (contract funding for the Ph.D. thesis of C. Weisbecker at INPT) and C. Ashcraft (LSTC).

5.20. Parallel computation of inverse entries of a sparse matrix

We have worked on the parallel computation of several entries [31] of the inverse of a large sparse matrix. We assume that the matrix has already been factorized by a direct method and that the factors are distributed. Entries are efficiently computed by exploiting sparsity of the right-hand sides and the solution vectors in the triangular solution phase. We demonstrate that in this setting, parallelism and computational efficiency are two contrasting objectives. We develop an efficient approach and show its efficacy by runs using the MUMPS code that implements a parallel multifrontal method.

5.21. Robust memory-aware mappings for parallel multifrontal factorization

We have studied the memory scalability of the parallel multifrontal factorization of sparse matrices. In particular, we are interested in controlling the active memory specific to the multifrontal factorization. We illustrate why commonly used mapping strategies (e.g. proportional mapping) cannot achieve a high memory efficiency. We propose a class of "memory-aware" algorithms that aim at maximizing performance under given memory constraints, and explain why they provide reliable memory estimates, thus a more robust solver. We study these issues in the context of the MUMPS solver, in which new experimental static scheduling strategies have been implemented and experimented on large matrices [46].

6. Partnerships and Cooperations

6.1. National Initiatives

6.1.1. ANR

ANR White Project RESCUE (2010-2014), 4 years. The ANR White Project RESCUE was launched in November 2010, for a duration of 48 months. It gathers three Inria partners (ROMA, Grand-Large and Hiepacs) and is led by ROMA. The main objective of the project is to develop new algorithmic techniques and software tools to solve the *exascale resilience problem*. Solving this problem implies a departure from current approaches, and calls for yet-to-be-discovered algorithms, protocols and software tools.

This proposed research follows three main research thrusts. The first thrust deals with novel *checkpoint protocols*. The second thrust entails the development of novel *execution models*, i.e., accurate stochastic models to predict (and, in turn, optimize) the expected performance (execution time or throughput) of large-scale parallel scientific applications. In the third thrust, we will develop novel *parallel algorithms* for scientific numerical kernels.

6.2. International Initiatives

6.2.1. Inria Associate Teams

The ALOHA associate-team is a joint project of the ROMA team and of the Information and Computer science Department of the University of Hawai'i (UH) at Mānoa, Honolulu, USA. Building on a vast array of theoretical techniques and expertise developed in the field of parallel and distributed computing, and more particularly application *scheduling*, we tackle database questions from a fresh perspective. To this end, this proposal includes:

- a group that specializes in database systems research and who has both industrial and academic experience, the group of Lipyeow Lim (UH);
- a group that specializes in practical aspects of scheduling problems and in simulation for emerging platforms and applications, and who has a long experience of multidisciplinary research, the group of Henri Casanova (UH);
- a group that specializes in the theoretical aspects of scheduling problems and resource management (the ROMA team).

The research work focuses on the following three thrusts:

1. Online, multi-criteria query optimization
2. Fault-Tolerance for distributed databases
3. Query scheduling for distributed databases

6.3. International Research Visitors

6.3.1. Visits of International Scientists

Oliver Sinnen, senior lecturer at the Department of Electrical and Computer Engineering (ECE) of the University of Auckland, New Zealand, visited the ROMA team for three months (April-June, 2012). He worked with Loris Marchal and Frédéric Vivien on scheduling tree-shaped task graphs to minimize both the peak memory usage and the makespan (see Section 5.5).

6.3.2. 7th Scheduling for large scale systems workshop

The University of Pittsburgh (Rami Melhem), the ROMA team (Yves Robert and Frédéric Vivien) and the University of Hawai'i at Manoa (Henri Casanova) have organized a workshop in Pittsburgh, on June 28-30, 2012. The workshop focused on scheduling and algorithms for large-scale systems. This was the seventh edition of this workshop series, after Aussois in August 2004, San Diego in November 2005, Aussois in May 2008, Knoxville in May 2009, Aussois in May 2010, and Aussois in May 2011. The next workshop will be held in Schloss Dagstuhl in September 2013.

7. Dissemination

7.1. Scientific Animation

Anne Benoit is an associate editor of the *Journal of Parallel and Distributed Computing (JPDC)*. She was program vice co-chair of IEEE Cluster 2012; program vice-chair of IEEE AINA 2012; member of the organizing committee of SIAM PP 2012, and organizer of a mini-symposium in SIAM PP12. She is workshops co-chair of ICPP 2013. She is or was a member of the program committees of of the following conferences and workshops: CCGrid 2012, HPDC 2012, IPDPS 2012, IPCE 2013, CCGrid 2013, IPDPS 2013, CLOSER 2013, HCW 2013, IGCC 2013.

Jean-Yves L'Excellent was a member of the program committees of Vecpar'12, Renpar'13.

Loris Marchal was a member of the program committees of IPDPS'2012, ICPP 2012 and IPDPS'2013.

Yves Robert is an associate editor of *IJHPCA*, *IJGUC* and *JOCS*. He is Program Chair of ICPP 2013 (Int. Conference on Parallel Processing) and of HiPC 2013 (Int. Conference on High Performance Computing). He was Program vice-chair of HiPC 2012. He is a Steering committee member of IPDPS and HCW. He is or was a member of the program committees of of the following conferences and workshops: EduPar 2012, FTXS 2012, ISC 2012, ISCIS 2012, EduPar 2013, FTXS 2013, ICCS 2013, IGCC 2013, ISC 2013 and SC 2013.

Bora Uçar was a member of the program committee for six conferences/workshops in 2012 (HiPC12, IEEE Cluster 12, EuroPart 2012, PCO'12, PMAA2012, TCPP PhD Forum). He was an organizer of a mini-symposium in SIAM PP12. He is in the program committee of IPDPS 2013.

Frédéric Vivien is an associate editor of *Parallel Computing*. Frédéric Vivien is program vice-chair, for the algorithms track, of IPDPS 2013. He is or was a member of the program committee of the following conferences and workshops: EuroPDP 2013, RenPar 2013, ROADEF 2013, SC 13, CCGrid 2012, Cluster 2012, ROADEF 2012, AHPAA 2012.

7.2. Teaching - Supervision - Juries

7.2.1. Teaching

Loris Marchal, Ordonnancement, 36h, M2, École normale supérieure de Lyon, France.

Bora Uçar has given a mini course in Parallel Computing Group at the University of Murcia, 28 and 29 November 2011.

Frédéric Vivien, Algorithmique et Programmation Parallèles, 36 h, M1, École normale supérieure de Lyon, France.

Frédéric Vivien, Ordonnancement, 3 h, M2, École normale supérieure de Lyon, France.

7.2.2. Supervision

PhD: Paul Renaud-Goud, Energy-aware scheduling: complexity and algorithms, École Normale Supérieure de Lyon, July 5, 2012, Anne Benoit and Yves Robert.

PhD in progress: Guillaume Aupy, Multi-criteria scheduling on volatile platforms, September 1, 2011, Anne Benoit and Yves Robert.

PhD in progress: Dounia Zaidouni, Performance and execution models for exascale applications in failure-prone environments, October 1, 2011, Frédéric Vivien and Yves Robert.

PhD in progress: Mohamed Sid-Lakhdar, Exploitation of multicore architectures in the resolution of sparse linear systems by multifrontal methods, October 1, 2011, Jean-Yves L'Excellent et Frédéric Vivien.

PhD in progress: Julien Herrmann, Numerical algorithms for large-scale platforms, September 1, 2012, Loris Marchal and Yves Robert.

8. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] J.-Y. L'EXCELLENT. *Multifrontal Methods: Parallelism, Memory Usage and Numerical Aspects*, Ecole normale supérieure de Lyon - ENS LYON, September 2012, Habilitation à Diriger des Recherches, http://hal.inria.fr/tel-00737751.

[2] P. RENAUD-GOUD. *Energy-aware scheduling : complexity and algorithms*, Ecole normale supérieure de Lyon - ENS LYON, July 2012, http://hal.inria.fr/tel-00744247.

Articles in International Peer-Reviewed Journals

[3] K. AGRAWAL, A. BENOIT, F. DUFOSSÉ, Y. ROBERT. *Mapping filtering streaming applications*, in "Algorithmica", 2012, vol. 62, n^{o} 1, p. 258-308.

[4] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT, Y. ROBERT, F.-H. ROUET, B. UÇAR. *On computing inverse entries of a sparse matrix in an out-of-core environment*, in "SIAM Journal on Scientific Computing (SISC)", 2012, vol. 34, n^{o} 4, p. A1975-A1999.

[5] G. AUPY, A. BENOIT, F. DUFOSSÉ, Y. ROBERT. *Reclaiming the energy of a schedule: models and algorithms*, in "Concurrency and Computation: Practice and Experience", 2012, To appear. Available on-line at the journal website.

[6] A. BENOIT, L.-C. CANON, E. JEANNOT, Y. ROBERT. *Reliability of task graph schedules with transient and fail-stop failures: complexity and algorithms*, in "Journal of Scheduling", 2012, vol. 15, n^{o} 5, p. 615-627, http://dx.doi.org/10.1007/s10951-011-0236-y.

[7] A. BENOIT, U. CATALYUREK, Y. ROBERT, E. SAULE. *A survey of pipelined workflow scheduling: models and algorithms*, in "ACM Computing Surveys", 2012, To appear.

[8] A. BENOIT, Y. ROBERT, A. L. ROSENBERG, F. VIVIEN. *Static strategies for worksharing with unrecoverable interruption*, in "Theory of Computing Systems", 2012, To appear., http://dx.doi.org/10.1007/s00224-012-9426-z.

[9] K. KAYA, J. LANGGUTH, F. MANNE, B. UÇAR. *Push-relabel based algorithms for the maximum transversal problem*, in "Computers & Operations Research", 2012, To appear.

[10] S. K. PRASAD, A. GUPTA, K. KANT, A. LUMSDAINE, D. A. PADUA, Y. ROBERT, A. L. ROSENBERG, A. SUSSMAN, C. C. WEEMS. *Literacy for all in parallel and distributed computing: guidelines for an undergraduate core curriculum*, in "CSI Journal of Computing", 2012, To appear.

[11] S. K. PRASAD, A. GUPTA, K. KANT, A. LUMSDAINE, D. A. PADUA, Y. ROBERT, A. L. ROSENBERG, A. SUSSMAN, C. C. WEEMS. *Toward a core undergraduates curriculum in parallel and distributed computing*, in "Computer Education (China)", 2012, p. 76-90.

[12] M. L. STILLWELL, F. VIVIEN, H. CASANOVA. *Dynamic Fractional Resource Scheduling versus Batch Scheduling*, in "Parallel and Distributed Systems, IEEE Transactions on", 2012, vol. 23, n^{o} 3, http://dx.doi.org/10.1109/TPDS.2011.183.

International Conferences with Proceedings

[13] G. AUPY, A. BENOIT, Y. ROBERT. *Energy-aware scheduling under reliability and makespan constraints*, in "International Conference on High Performance Computing (HiPC'2012)", IEEE Computer Society Press, 2012.

[14] O. BEAUMONT, N. BONICHON, L. EYRAUD-DUBOIS, L. MARCHAL. *Minimizing weighted mean completion time for malleable tasks scheduling*, in "Proceedings of IPDPS 2012", IEEE, 2012.

[15] A. BENOIT, M. COQBLIN, J.-M. NICOD, L. PHILIPPE, V. REHN-SONIGO. *Throughput optimization for pipeline workflow scheduling with setup times*, in "Proceedings of CGWS 2012, the CoreGRID/ERCIM Workshop on Grids, Clouds and P2P Computing, in conjunction with EuroPar 2012", Rhodes Island, Greece, August 2012, Also available as Inria Research report RR-7886, Version 2, June 2012.

[16] A. BENOIT, H. LARCHEVÊQUE, P. RENAUD-GOUD. *Optimal algorithms and approximation algorithms for replica placement with distance constraints in tree networks*, in "Proceedings of IPDPS 2012", IEEE, 2012.

[17] A. BENOIT, R. MELHEM, P. RENAUD-GOUD, Y. ROBERT. *Power-aware Manhattan routing on chip multiprocessors*, in "IPDPS'2012, the 26th IEEE International Parallel and Distributed Processing Symposium", IEEE Computer Society Press, 2012.

[18] H. CASANOVA, F. DUFOSSÉ, Y. ROBERT, F. VIVIEN. *Mapping tightly-coupled applications on volatile resources*, in "PDP'2013, the 21st Euromicro Int. Conf. on Parallel, Distributed, and Network-Based Processing", IEEE Computer Society Press, 2013.

[19] J. DONGARRA, M. FAVERGE, T. HÉRAULT, J. LANGOU, Y. ROBERT. *Hierarchical QR factorization algorithms for multi-core cluster systems*, in "IPDPS'2012, the 26th IEEE International Parallel and Distributed Processing Symposium", IEEE Computer Society Press, 2012.

[20] A. GRANDJEAN, J. LANGGUTH, B. UÇAR. *On optimal and balanced sparse matrix partitioning problems*, in "2012 IEEE International Conference on Cluster Computing", Los Alamitos, CA, USA, IEEE Computer Society, 2012, p. 257–265.

[21] K. KAYA, F.-H. ROUET, B. UÇAR. *On partitioning problems with complex objectives*, in "Euro-Par 2011: Parallel Processing Workshops", M. ALEXANDER, P. D'AMBRA, A. BELLOUM, G. BOSILCA, M. CANNATARO, M. DANELUTTO, B. DI MARTINO, M. GERNDT, E. JEANNOT, R. NAMYST, J. ROMAN, S. SCOTT, J. TRAFF, G. VALLÉE, J. WEIDENDORFER (editors), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, vol. 7155, p. 334-344.

[22] Y. ROBERT, F. VIVIEN, D. ZAIDOUNI. *On the complexity of scheduling checkpoints for computational workflowss*, in "FTXS'2012, the Workshop on Fault-Tolerance for HPC at Extreme Scale, in conjunction with the 42nd Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN 2012)", IEEE Computer Society Press, 2012, http://dx.doi.org/10.1109/DSNW.2012.6264675.

[23] M. L. STILLWELL, F. VIVIEN, H. CASANOVA. *Virtual Machine Resource Allocation for Service Hosting on Heterogeneous Distributed Platforms*, in "proceedings of IPDPS 2012", IEEE, 2012, http://dx.doi.org/10.1109/IPDPS.2012.75.

[24] Ü. V. ÇATALYÜREK, M. DEVECI, K. KAYA, B. UÇAR. *Multithreaded clustering for multi-level hypergraph partitioning*, in "26th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2012,", Shanghai, China, IEEE Computer Society, 2012, p. 848–859.

[25] Ü. V. ÇATALYÜREK, K. KAYA, B. UÇAR. *On shared-memory parallelization of a sparse matrix scaling algorithm*, in "2012 41st International Conference on Parallel Processing", Los Alamitos, CA, USA, IEEE Computer Society, September 2012, p. 68–77.

Scientific Books (or Scientific Book chapters)

[26] A. BENOIT, L. MARCHAL, Y. ROBERT, B. UÇAR, F. VIVIEN. *Scheduling for large-scale systems*, in "The Computing Handbook Set, vol. 1", Chapman and Hall/CRC Press, 2013, To appear.

[27] I. S. DUFF, B. UÇAR. *Combinatorial problems in solving linear systems*, in "Combinatorial Scientific Computing", U. NAUMANN, O. SCHENK (editors), CRC Press, 2012, chap. 2, p. 21–68.

[28] M. SATHE, O. SCHENK, B. UÇAR, A. SAMEH. *A scalable hybrid linear solver based on combinatorial algorithms*, in "Combinatorial Scientific Computing", U. NAUMANN, O. SCHENK (editors), CRC Press, 2012, chap. 4, p. 95–127.

[29] Ü. V. ÇATALYÜREK, K. KAYA, J. LANGGUTH, B. UÇAR. *A Partitioning-based divisive clustering technique for maximizing the modularity*, in "Graph Partitioning and Graph Clustering", D. A. BADER, H. MEYERHENKE, P. SANDERS, D. WAGNER (editors), Contemporary Mathematics, AMS, 2012, to appear.

[30] Ü. V. ÇATALYÜREK, M. DEVECI, K. KAYA, B. UÇAR. *UMPA: A Multi-objective, multi-level partitioner for communication minimization*, in "Graph Partitioning and Graph Clustering", D. A. BADER, H. MEYERHENKE, P. SANDERS, D. WAGNER (editors), Contemporary Mathematics, AMS, 2012, to appear.

Research Reports

[31] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT, F.-H. ROUET. *Parallel computation of entries of A-1*, Inria, December 2012, n^{o} RR-8142, http://hal.inria.fr/hal-00759556.

[32] G. AUPY, A. BENOIT. *Approximation algorithms for energy, reliability and makespan optimization problems*, Inria, October 2012, n^{o} RR-8107, 32, http://hal.inria.fr/hal-00742754.

[33] G. AUPY, A. BENOIT, Y. ROBERT. *Energy-aware scheduling under reliability and makespan constraints*, Inria, February 2012, n^{o} RR-7757, 25, http://hal.inria.fr/inria-00630721.

[34] G. AUPY, Y. ROBERT, F. VIVIEN, D. ZAIDOUNI. *Impact of fault prediction on checkpointing strategies*, Inria, October 2012, n^{o} RR-8023, http://hal.inria.fr/hal-00720401.

[35] O. BEAUMONT, H. LARCHEVÊQUE, L. MARCHAL. *Non-Linear divisible loads: There is no free lunch*, Inria, December 2012, n^{o} RR-8170, 20, http://hal.inria.fr/hal-00762008.

[36] A. BENOIT, M. COQBLIN, J.-M. NICOD, L. PHILIPPE, V. REHN-SONIGO. *Throughput optimization for pipeline workflow scheduling with setup times*, Inria, February 2012, n^{o} RR-7886, 29, http://hal.inria.fr/hal-00674057.

[37] A. BENOIT, J. LANGGUTH, B. UÇAR. *Semi-matching algorithms for scheduling parallel tasks under resource constraints*, Inria, October 2012, n^{o} RR-8089, 30, http://hal.inria.fr/hal-00738393.

[38] G. BOSILCA, A. BOUTEILLER, É. BRUNET, F. CAPPELLO, J. DONGARRA, A. GUERMOUCHE, T. HÉRAULT, Y. ROBERT, F. VIVIEN, D. ZAIDOUNI. *Unified Model for Assessing Checkpointing Protocols at Extreme-Scale*, Inria, October 2012, n^{o} RR-7950, http://hal.inria.fr/hal-00696154.

[39] M. BOUGERET, H. CASANOVA, Y. ROBERT, F. VIVIEN, D. ZAIDOUNI. *Using group replication for resilience on exascale systems*, Inria, February 2012, n^{o} RR-7876, http://hal.inria.fr/hal-00668016.

[40] H. CASANOVA, F. DUFOSSÉ, Y. ROBERT, F. VIVIEN. *Mapping Tightly-Coupled Applications on Volatile Resources*, May 2012, http://hal.inria.fr/ensl-00697621.

[41] H. CASANOVA, Y. ROBERT, F. VIVIEN, D. ZAIDOUNI. *Combining Process Replication and Checkpointing for Resilience on Exascale Systems*, Inria, May 2012, n° RR-7951, http://hal.inria.fr/hal-00697180.

[42] K. KAYA, J. LANGGUTH, F. MANNE, B. UÇAR. *Investigations on push-relabel based algorithms for the maximum transversal problem*, Inria, October 2012, n° RR-8093, 27, http://hal.inria.fr/hal-00739360.

[43] P. A. KNIGHT, D. RUIZ, B. UÇAR. *A Symmetry preserving algorithm for matrix scaling*, Inria, October 2012, n° revision of RR-7552, http://hal.inria.fr/inria-00569250.

[44] L. MARCHAL, O. SINNEN, F. VIVIEN. *Scheduling tree-shaped task graphs to minimize memory and makespan*, Inria, October 2012, n° RR-8082, 21, http://hal.inria.fr/hal-00740105.

[45] Y. ROBERT, F. VIVIEN, D. ZAIDOUNI. *On the complexity of scheduling checkpoints for computational workflows*, Inria, March 2012, n° RR-7907, http://hal.inria.fr/hal-00680386.

Other Publications

[46] E. AGULLO, P. R. AMESTOY, A. BUTTARI, A. GUERMOUCHE, J.-Y. L'EXCELLENT, F.-H. ROUET. *Robust memory-aware mappings for parallel multifrontal factorizations*, February 2012, Presentation at the 15th SIAM conference on Parallel Processing for Scientific Computing (PP12), Savannah, GA, USA.

[47] P. R. AMESTOY, C. ASHCRAFT, A. BUTTARI, J.-Y. L'EXCELLENT, C. WEISBECKER. *Improving Multifrontal methods by means of Low-Rank Approximations techniques*, June 2012, Presentation at the 2012 SIAM conference on Applied Linear Algebra, Valencia, Spain.

[48] P. R. AMESTOY, A. BUTTARI, A. GUERMOUCHE, J.-Y. L'EXCELLENT, M. SID-LAKHDAR. *Exploiting Multithreaded Tree Parallelism for Multicore Systems in a Parallel Multifrontal Solver*, February 2012, Presentation at the 15th SIAM conference on Parallel Processing for Scientific Computing (PP12), Savannah, GA, USA.

[49] B. UÇAR. *Partitioning problems on trees and simple meshes*, February 2012, Presentation at 15th SIAM Conference on Parallel Processing for Scientific Computing (PP12), Savannah, Georgia, USA.

References in notes

[50] S. C. EISENSTAT, J. W. H. LIU. *The theory of elimination trees for sparse unsymmetric matrices*, in "SIAM Journal on Matrix Analysis and Applications", 2005, vol. 26, n° 3, p. 686–705.

[51] S. C. EISENSTAT, J. W. H. LIU. *Algorithmic aspects of elimination trees for sparse unsymmetric matrices*, in "SIAM Journal on Matrix Analysis and Applications", 2008, vol. 29, n° 4, p. 1363–1381.

[52] K. FERREIRA, J. STEARLEY, J. H. I. LAROS, R. OLDFIELD, K. PEDRETTI, R. BRIGHTWELL, R. RIESEN, P. G. BRIDGES, D. ARNOLD. *Evaluating the viability of process replication reliability for exascale systems*, in "Proc. 2011 Int. Conf. High Performance Computing, Networking, Storage and Analysis SC '11", ACM Press, 2011.

[53] K. KAYA, B. UÇAR. *Constructing elimination trees for sparse unsymmetric matrices*, Inria, February 2011, n^{o} RR-7549.

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

[55] D. RUIZ, B. UÇAR. *A symmetry preserving algorithm for matrix scaling*, Inria, 2011, n^{o} RR-7552.