INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team GRAAL

Algorithms and Scheduling for Distributed Heterogeneous Platforms

Grenoble - Rhône-Alpes

THEME NUM

Activity Report

2008

Table of contents

The GRAAL *project-team is common to CNRS, ENS Lyon, and INRIA. This project-team is part of the Laboratoire de l'Informatique du Parallélisme (LIP), UMR ENS Lyon/CNRS/INRIA/UCBL 5668. This project-team is located in part at the École normale supérieure de Lyon and in part at the Université Claude Bernard – Lyon 1.*

1. Team

Research Scientist

Frédéric Desprez [Research Director (DR), HdR]
Gilles Fedak [Research Associate (CR), since September 1, 2008]
Jean-Yves L'Excellent [Research Associate (CR), Acting as Team Leader during Frédéric Vivien's sabbatical]
Loris Marchal [Research Associate (CR)]
Christian Pérez [Research Associate (CR), since October 1, 2008, HdR]
Frédéric Vivien [Team Leader, Research Associate (CR), HdR]

Faculty Member

Anne Benoît [Assistant Professor (MCF)]
Hinde Bouziane [ATER, since September 1, 2008]
Yves Caniou [Assistant Professor (MCF)]
Eddy Caron [Assistant Professor (MCF)]
Gaël Le Mahec [ATER, since September 1, 2008]
Bernard Tourancheau [Professor, HdR]
Yves Robert [Professor, HdR]

External Collaborator

Sékou Diakité [PhD student, MENRT grant]
Alexandru Dobrila [PhD student, MENRT grant]
Jean-Marc Nicod [Assistant Professor, HdR]
Laurent Philippe [Professor, HdR]

Technical Staff

Nicolas Bard [CNRS]
Aurélien Ceyden [ENS Lyon, 50% on the project]
Philippe Combes [CNRS, until December 15, 2008]
Haiwu He [INRIA, since September 1, 2008]
Benjamin Isnard [INRIA, since March 1, 2008]
David Loureiro [INRIA, until September 30, 2008]
Vincent Pichon [ENS Lyon]

PhD Student

Emmanuel Agullo [MENRT grant until September 30, INRIA grant until December 31]
Leila Ben Saad [MENRT grant, starting September 1, 2008]
Julien Bigot [MENRT grant, since September 1, 2008]
Raphaël Bolze [BDI CNRS until September 30; INRIA grant until October 31; ENS-AFM grant until January 31, 2009]
Ghislain Charrier [INRIA Cordi-S grant]
Benjamin Depardon [MENRT grant]
Fanny Dufossé [ENS Grant, starting September 1, 2008]
Matthieu Gallet [ENS grant]
Jean-Sébastien Gay [Rhône-Alpes region grant, on long term leave for health reasons]
Mathias Jacquelin [MENRT grant, starting October 1, 2008]
Jean-François Pineau [ENS grant, until August 31, 2008]
Veronika Rehn-Sonigo [MENRT grant]
Clément Rezvoy [MENRT grant]

Cédric Tedeschi [MENRT grant, until October 3, 2008]

Post-Doctoral Fellow

Alfredo Buttari [INRIA, until October 15, 2008]
Mourad Hakem [INRIA, until August 31, 2008]

Visiting Scientist

Franck Petit [On leave from University of Picardie since September 1, 2008]
Bing Trang [Wuhan University of technology, since October 28, 2008]

Administrative Assistant

Caroline Suter [INRIA, 50% on the project]

2. Overall Objectives

2.1. Introduction

Keywords: *Grid computing*, *algorithm design for heterogeneous systems*, *distributed application*, *library*, *programming environment*.

Parallel computing has spread into all fields of applications, from classical simulation of mechanical systems or weather forecast to databases, video-on-demand servers or search tools like Google. From the architectural point of view, parallel machines have evolved from large homogeneous machines to clusters of PCs (with sometime boards of several processors sharing a common memory, these boards being connected by high speed networks like Myrinet). However the need of computing or storage resources has continued to grow leading to the need of resource aggregation through Local Area Networks (LAN) or even Wide Area Networks (WAN). The recent progress of network technology has made it possible to use highly distributed platforms as a single parallel resource. This has been called Metacomputing or more recently Grid Computing [80]. An enormous amount of financing has recently been put on this important subject, leading to an exponential growth of the number of projects, most of them focusing on low level software detail. We believe that many of these projects failed to study fundamental problems such as problems and algorithms complexity, and scheduling heuristics. Also they usually have not validated their theoretical results on available software platforms.

From the architectural point of view, Grid Computing has different scales but is always highly heterogeneous and hierarchical. At a very large scale, tens of thousands of PCs connected through the Internet are aggregated to solve very large applications. This form of the Grid, usually called a Peer-to-Peer (P2P) system, has several incarnations, such as SETI@home, Gnutella or XtremWeb [93]. It is already used to solve large problems (or to share files) on PCs across the world. However, as today's network capacity is still low, the applications supported by such systems are usually embarrassingly parallel. Another large-scale example is TeraGRID which connects several supercomputing centers in the USA and reaches a peak performance of over 100 Teraflops. At a smaller scale but with a high bandwidth, one can mention the Grid'5000 project, which connects PC clusters spread in nine French university research centers. Many such projects exist over the world that connect a small set of machines through a fast network. Finally, at a research laboratory level, one can build an heterogeneous platform by connecting several clusters using a fast network such as Myrinet.

The common problem of all these platforms is not the hardware (these machines are already connected to the Internet) but the software (from the operating system to the algorithmic design). Indeed, the computers connected are usually highly heterogeneous (from clusters of SMPs to the Grid).

There are two main challenges for the widespread use of Grid platforms: the development of environments that will ease the use of the Grid (in a seamless way) and the design and evaluation of new algorithmic approaches for applications using such platforms. Environments used on the Grid include operating systems, languages, libraries, and middlewares [87], [78], [80]. Today's environments are based either on the adaptation of "classical" parallel environments or on the development of toolboxes based on Web Services.

Aims of the GRAAL project.

In the GRAAL project we work on the following research topics:

- algorithms and scheduling strategies for heterogeneous platforms and the Grid,
- environments and tools for the deployment of applications in a client-server mode.

The main keywords of the GRAAL project:
Algorithmic Design + Middleware/Libraries + Applications
over Heterogeneous Architectures and the Grid

2.2. Highlights of the year

- 2008 saw the launch of a start-up company around the DIET middleware and GRAAL's expertise on the deployment of large scale applications over dedicated grids.

3. Scientific Foundations

3.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Anne Benoît, Leila Ben Saad, Sékou Diakité, Alexandru Dobrila, Fanny Dufossé, Matthieu Gallet, Mourad Hakem, Mathias Jacquelin, Loris Marchal, Jean-Marc Nicod, Laurent Philippe, Jean-François Pineau, Veronika Rehn-Sonigo, Clément Rezvoy, Yves Robert, Bernard Tourancheau, Frédéric Vivien.

Scheduling sets of computational tasks on distributed platforms is a key issue but a difficult problem. Although a large number of scheduling techniques and heuristics have been presented in the literature, most of them target only homogeneous resources. However, future computing systems, such as the computational Grid, are most likely to be widely distributed and strongly heterogeneous. Therefore, we consider the impact of heterogeneity on the design and analysis of scheduling techniques: how to enhance these techniques to efficiently address heterogeneous distributed platforms?

The traditional objective of scheduling algorithms is the following: given a task graph and a set of computing resources, or *processors*, map the tasks onto the processors, and order the execution of the tasks so that: (i) the task precedence constraints are satisfied; (ii) the resource constraints are satisfied; and (iii) a minimum schedule length is achieved. Task graph scheduling is usually studied using the so-called *macro-dataflow* model, which is widely used in the scheduling literature: see the survey papers [79], [92], [101], [103] and the references therein. This model was introduced for homogeneous processors, and has been (straightforwardly) extended to heterogeneous computing resources. In a word, there is a limited number of computing resources, or processors, to execute the tasks. Communication delays are taken into account as follows: let task T be a predecessor of task T' in the task graph; if both tasks are assigned to the same processor, no communication overhead is incurred, the execution of T' can start immediately at the end of the execution of T; on the contrary, if T and T' are assigned to two different processors P_i and P_j, a communication delay is incurred. More precisely, if P_i completes the execution of T at time-step t, then P_j cannot start the execution of T' before time-step $t + \text{comm}(T, T', P_i, P_j)$, where $\text{comm}(T, T', P_i, P_j)$ is the communication delay, which depends upon both tasks T and T', and both processors P_i and P_j. Because memory accesses are typically several orders of magnitude cheaper than inter-processor communications, it is sensible to neglect them when T and T' are assigned to the same processor.

The major flaw of the macro-dataflow model is that communication resources are not limited in this model. Firstly, a processor can send (or receive) any number of messages in parallel, hence an unlimited number of communication ports is assumed (this explains the name *macro-dataflow* for the model). Secondly, the number of messages that can simultaneously circulate between processors is not bounded, hence an unlimited number of communications can simultaneously occur on a given link. In other words, the communication network is assumed to be contention-free, which of course is not realistic as soon as the number of processors exceeds a few units.

The general scheduling problem is far more complex than the traditional objective in the *macro-dataflow* model. Indeed, the nature of the scheduling problem depends on the type of tasks to be scheduled, on the platform architecture, and on the aim of the scheduling policy. The tasks may be independent (e.g., they represent jobs submitted by different users to a same system, or they represent occurrences of the same program run on independent inputs), or the tasks may be dependent (e.g., they represent the different phases of a same processing and they form a task graph). The platform may or may not have a hierarchical architecture (clusters of clusters vs. a single cluster), it may or may not be dedicated. Resources may be added to or may disappear from the platform at any time, or the platform may have a stable composition. The processing units may have the same characteristics (e.g., computational power, amount of memory, multi-port or only single-port communications support, etc.) or not. The communication links may have the same characteristics (e.g., bandwidths, latency, routing policy, etc.) or not. The aim of the scheduling policy can be to minimize the overall execution time (makespan minimization), the throughput of processed tasks, etc. Finally, the set of all tasks to be scheduled may be known from the beginning, or new tasks may arrive all along the execution of the system (on-line scheduling).

In the GRAAL project, we investigate scheduling problems that are of practical interest in the context of large-scale distributed platforms. We assess the impact of the heterogeneity and volatility of the resources onto the scheduling strategies.

3.2. Scheduling for Parallel Sparse Direct Solvers

Participants: Emmanuel Agullo, Alfredo Buttari, Jean-Yves L'Excellent.

The solution of sparse systems of linear equations (symmetric or unsymmetric, most often with an irregular structure) is at the heart of many scientific applications, most often related to numerical simulation: geophysics, chemistry, electromagnetism, structural optimization, computational fluid dynamics, etc. The importance and diversity of the fields of application are our main motivation to pursue research on sparse linear solvers. Furthermore, in order to deal with the larger and larger problems that result from increasing demands in simulation, special attention must be paid to both memory usage and execution time on the most powerful parallel platforms now available (whose usage is necessary because of the volume of data and amount of computation induced). This is done by specific algorithmic choices and scheduling techniques. From a complementary point of view, it is also necessary to be aware of the functionality requirements from the applications and from the users, so that robust solutions can be proposed for a large range of problems.

Because of their efficiency and robustness, direct methods (based on Gaussian factorization) are methods of choice to solve these types of problems. In this context, we are particularly interested in the multifrontal method [90], [91], for symmetric positive definite, general symmetric or unsymmetric problems, with numerical pivoting in order to ensure numerical stability. Note that numerical pivoting induces dynamic data structures that are unpredictable symbolically or from a static analysis.

The multifrontal method is based on an elimination tree [97] which results (i) from the graph structure corresponding to the nonzero pattern of the problem to be solved, and (ii) from the order in which variables are eliminated. This tree provides the dependency graph of the computations and is exploited to define tasks that may be executed in parallel. In this method, each node of the tree corresponds to a task (itself potentially parallel) that consists in the partial factorization of a dense matrix. This approach allows for a good locality and usage of cache memories.

In order to deal with numerical pivoting and keep an approach that can adapt to as many computer architectures as we can, we are especially interested in approaches that are intrinsically dynamic and asynchronous [1], [84]. In addition to their numerical robustness, the algorithms retained are based on a dynamic and distributed management of the computational tasks, not so far from today's peer-to-peer approaches: each process is responsible for providing work to some other processes and at the same time it acts as a slave for others. These algorithms are very interesting from the point of view of parallelism and in particular for the study of mapping and scheduling strategies for the following reasons:

- the associated task graphs are very irregular and can vary dynamically,

- these algorithms are currently used inside industrial applications, and
- the evolution of high performance platforms, more heterogeneous and less predictable, requires that applications adapt, using a mixture of dynamic and static approaches, as our approach allows.

Note that our research in this field is strongly linked to the software package MUMPS (see Section 5.2) which is our main platform to experiment and validate new ideas and research directions. Finally, note that we are facing new challenges for very large problems (tens to hundreds of millions of equations) that occur nowadays in various application fields: in that case, either parallel out-of-core approaches are required, or direct solvers should be combined with iterative schemes, leading to hybrid direct-iterative methods.

3.3. Providing Access to HPC Servers on the Grid

Participants: Nicolas Bard, Julien Bigot, Raphaël Bolze, Hinde Bouziane, Yves Caniou, Eddy Caron, Aurélien Ceyden, Ghislain Charrier, Benjamin Depardon, Frédéric Desprez, Gilles Fedak, Jean-Sébastien Gay, Haiwu He, David Loureiro, Christian Pérez, Vincent Pichon, Cédric Tedeschi, Bing Trang.

Resource management is one of the key issues for the development of efficient Grid environments. Several approaches co-exist in today's middleware platforms. The computation (or communication) grain and the dependences between the computations also have a great influence on the software choices.

A first approach provides the user with a uniform view of resources. This is the case of GLOBUS [1] which provides transparent MPI communications (with MPICH-G2) between distant nodes but does not manage load balancing issues between these nodes. It is the user's task to develop a code that will take into account the heterogeneity of the target architecture. Classical batch processing can also be used on the Grid with projects like Condor-G [2] or Sun GridEngine [3]. Finally, peer-to-peer [81] or Global computing [95] can be used for fine grain and loosely coupled applications.

A second approach provides a semi-transparent access to computing servers by submitting jobs to dedicated servers. This model is known as the Application Service Provider (ASP) model where providers offer, not necessarily for free, computing resources (hardware and software) to clients in the same way as Internet providers offer network resources to clients. The programming granularity of this model is rather coarse. One of the advantages of this approach is that end users do not need to be experts in parallel programming to benefit from high performance parallel programs and computers. This model is closely related to the classical Remote Procedure Call (RPC) paradigm. On a Grid platform, the RPC (or GridRPC [98], [99]) offers an easy access to available resources to a Web browser, a Problem Solving Environment, or a simple client program written in C, Fortran, or Java. It also provides more transparency by hiding the search and allocation of computing resources. We favor this second approach.

In a Grid context, this approach requires the implementation of middleware environments to facilitate the client access to remote resources. In the ASP approach, a common way for clients to ask for resources to solve their problem is to submit a request to the middleware. The middleware will find the most appropriate server that will solve the problem on behalf of the client using a specific software. Several environments, usually called Network Enabled Servers (NES), have developed such a paradigm: NetSolve [86], Ninf [100], NEOS [94], OmniRPC [102], and more recently DIET developed in the GRAAL project (see Section 5.1). A common feature of these environments is that they are built on top of five components: clients, servers, databases, monitors, and schedulers. Clients solve computational requests on servers found by the NES. The NES schedules the requests on the different servers using performance information obtained by monitors and stored in a database.

[1] http://www.globus.org/

[2] http://www.cs.wisc.edu/condor/condorg/

[3] http://wwws.sun.com/software/gridware/

To design such a NES we need to address issues related to several well-known research domains. In particular, we focus on:

- middleware and application platforms as a base to implement the necessary "glue" to broke clients requests, find the best server available, and then submit the problem and its data,
- online and offline scheduling of requests,
- link with data management,
- distributed algorithms to manage the requests and the dynamic behavior of the platform.

4. Application Domains

4.1. Applications of Sparse Direct Solvers

In the context of our activity on sparse direct (multifrontal) solvers in distributed environments, we develop, distribute, maintain and support competitive software. Our methods have a wide range of applications and they are 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 almost always ends up solving a system of equations involving sparse matrices. There are therefore a number of application fields, among which we list in the following some of the ones cited by the users of our sparse direct solver MUMPS (see Section 5.2): structural mechanical engineering (stress analysis, structural optimization, car bodies, ships, crankshaft segment, offshore platforms, computer assisted design, computer assisted engineering, rigidity of sphere packings), heat transfer analysis, thermomechanics in casting simulation, fracture mechanics, biomechanics, medical image processing, tomography, plasma physics (e.g., Maxwell's equations), critical physical phenomena, geophysics (e.g., seismic wave propagation, earthquake related problems), ad-hoc networking modeling (Markovian processes), modeling of the magnetic field inside machines, econometric models, soil-structure interaction problems, oil reservoir simulation, computational fluid dynamics (e.g., Navier-Stokes, ocean/atmospheric modeling with mixed Finite Elements Methods, fluvial hydrodynamics, viscoelastic flows), electromagnetics, magneto-hydro-dynamics, modeling the structure of the optic nerve head and of cancellous bone, modeling of the heart valve, modeling and simulation of crystal growth processes, chemistry (chemical process modeling), vibro-acoustics, aero-acoustics, aero-elasticity, optical fiber modal analysis, blast furnace modeling, glaciology (models of ice flow), optimization, optimal control theory, astrophysics (e.g., supernova, thermonuclear reaction networks, neutron diffusion equation, quantum chaos, quantum transport), research on domain decomposition (MUMPS can for example be used on each subdomain in an iterative framework), circuit simulations, etc.

4.2. Molecular Dynamics

LAMMPS is a classical molecular dynamics (MD) code created for simulating molecular and atomic systems such as proteins in solution, liquid-crystals, polymers, zeolites, or simple Lenard-Jonesium. It was designed for distributed-memory parallel computers and runs on any parallel platform that supports the MPI message-passing library or on single-processor workstations. LAMMPS is mainly written in F90.

LAMMPS was originally developed as part of a 5-way DoE-sponsored CRADA collaboration between 3 industrial partners (Cray Research, Bristol-Myers Squibb, and Dupont) and 2 DoE laboratories (Sandia and Livermore). The code is freely available under the terms of a simple license agreement that allows you to use it for your own purposes, but not to distribute it further.

We plan to provide the Grid benefit to LAMMPS with an integration of this application into our Problem Solving Environment, DIET. A computational server will be available from a DIET client and the choice of the best server will be taken by the DIET agent.

4.3. Biochemistry

Current progress in different areas of chemistry like organic chemistry, physical chemistry or biochemistry allows the construction of complex molecular assemblies with predetermined properties. In all these fields, theoretical chemistry plays a major role by helping to build various models which can greatly differ in terms of theoretical and computational complexity, and which allow the understanding and the prediction of chemical properties.

Among the various theoretical approaches available, quantum chemistry is at a central position as all modern chemistry relies on it. This scientific domain is quite complex and involves heavy computations. In order to fully apprehend a model, it is necessary to explore the whole potential energy surface described by the independent variation of all its degrees of freedom. This involves the computation of many points on this surface.

Our project is to couple DIET with a relational database in order to explore the potential energy surface of molecular systems using quantum chemistry: all molecular configurations to compute are stored in a database, the latter is queried, and all configurations that have not been computed yet are passed through DIET to computer servers which run quantum calculations, all results are then sent back to the database through DIET. At the end, the database will store a whole potential energy surface which can then be analyzed using proper quantum chemical analysis tools.

4.4. Bioinformatics

Genomics acquiring programs, such as full genomes sequencing projects, are producing larger and larger amounts of data. The analysis of these raw biological data require very large computing resources. In some cases, due to the lack of sufficient computing and storage resources, skilled staff or technical abilities, laboratories cannot afford such huge analyses. Grid computing may be a viable solution to the needs of the genomics research field: it can provide scientists with a transparent access to large computational and data management resources.

In this application domain, we are currently addressing two different problems. In the first one we tackle the clusterization into domain protein families of the sequences contained in international databanks. Our aim is to ensure, through the use of grids, the capacity over time to automatically build databases such as ProDom, when such a database is built from exponentially-fast growing protein databases.

In the second problem, we consider protein functional sites. Functional sites and signatures of proteins are very useful for analyzing raw biological data or for correlating different kinds of existing biological data. These methods are applied, for example, to the identification and characterization of the potential functions of new sequenced proteins. The sites and signatures of proteins can be expressed by using the syntax defined by the PROSITE databank, and written as a “protein regular expression”. Searching one such site in a sequence can be done with the criterion of the identity between the searched and the found patterns. Most of the time, this kind of analysis is quite fast. However, in order to identify non perfectly matching but biologically relevant sites, the user can accept a certain level of error between the searched and the matching patterns. Such an analysis can be very resource consuming.

4.5. Cosmological Simulations

Ramses [4] is a typical computational intensive application used by astrophysicists to study the formation of galaxies. *Ramses* is used, among other things, to simulate the evolution of a collisionless, self-gravitating fluid called “dark matter” through cosmic time. Individual trajectories of macro-particles are integrated using a state-of-the-art “N body solver”, coupled to a finite volume Euler solver, based on the Adaptive Mesh Refinement technique. The computational space is decomposed among the available processors using a *mesh partitioning* strategy based on the Peano-Hilbert cell ordering.

[4] http://www.projet-horizon.fr/

Cosmological simulations are usually divided into two main categories. Large scale periodic boxes requiring massively parallel computers are performed on a very long elapsed time (usually several months). The second category stands for much faster small scale "zoom simulations". One of the particularity of the HORIZON project is that it allows the re-simulation of some areas of interest for astronomers.

We designed a Grid version of *Ramses* through the DIET middleware. From Grid'5000 experiments we proved DIET is capable of handling long cosmological parallel simulations: mapping them on parallel resources of a Grid, executing and processing communication transfers. The overhead induced by the use of DIET is negligible compared to the execution time of the services. Thus DIET permits to explore new research axes in cosmological simulations (on various low resolutions initial conditions), with transparent access to the services and the data.

4.6. Ocean-Atmosphere Simulations

Climatologists have recourse to numerical simulation and particularly coupled models in several occasions: for example, to estimate natural variability (thousand of simulated years), for seasonal forecasting (only a few simulated months) or to study global warming characteristics (some simulated decades).

To take advantage of the Grid'5000 platform, we choose to launch parallel simulations (ensemble) on several nodes, approximatively 10 or more, according to the load of the platform. Scenario simulations, that simulate from present climate to 21st century, require huge computing power. Indeed, each simulation will differ from each other in physical parameterization of atmospheric model. Comparing them, we expect to better estimate global warming prediction sensibility in order to model parameterization.

Practically, a 150 year long scenario combines 1800 simulations of one month each, launched one after the other. This partitioning eases workflow and implements checkpointing. The initial state of simulation of month "n" is the ending state of the simulation of month "n-1".

Our goal regarding the climate forecasting application is to thoroughly analyze it in order to model its needs in terms of execution model, data access pattern, and computing needs. Once a proper model of the application has been derived, appropriate scheduling heuristics can be proposed, tested, and compared. We plan to extend this work to provide generic scheduling schemes for applications with similar dependence graphs.

4.7. Décrypthon

The Décrypthon project is built over a collaboration between CNRS, AFM (*Association Française contre les Myopathies*), and IBM. Its goal is to make computational and storage resources available to bioinformatic research teams in France. These resources, connected as a Grid through the Renater network, are installed in six universities and schools in France (Bordeaux, Jussieu, Lille, Lyon, Orsay, and Rouen). The Décrypthon project offers means necessary to use the Grid through financing of research teams and postdoc, and assistance on computer science problems (modeling, application development, data management, ...). The GRAAL research team is involved in this project as an expert for application gridification.

The Grid middleware used at the beginning of the project was GridMP from United Devices. In 2007, other software solutions were evaluated and among them DIET, developed within GRAAL, and g-Lite from the european EGEE project. DIET was finally chosen to be the Grid middleware of the Décrypthon Grid. It ensures the load-balancing of jobs over the 6 computation centers through the Renater network.

The Décrypthon Grid is built over several components: the DIET Grid middleware, a web portal to access Grid resources, and local batch schedulers in each university. The web portal is installed on a dedicated machine in Orsay. It runs a specific web application for each research project which allows submission of computation request over all Décrypthon resources. The web portal then sends requests to the DIET middleware deployed over the Grid to find appropriate resources and application over the network. The DIET middleware is deployed as follows. One ServerDeamon (SeD) is started on every server frontend. It is then connected to the Master Agent that runs in Orsay. SeDs collect information about the server loads and submit jobs to local batch schedulers (Loadleveler, PBS, OAR ...). Indeed, several improvements are now provided in the DIET

Grid middleware: they give Décrypthon contributors a powerful API to be able to launch transparently on their behalf their applications, in particular on AIX systems using the Loadleveler reservation batch system. Application can be parallel or not. No need to focus on the batch syntax, a user just has to write how to manage his data and how to call his program, and DIET creates the correct script accordingly to the batch directives, submits on the correct queue and manage the job on behalf of the user. Moreover, SeD take in charge te data movement between storage servers and computational servers.

This transfer of our middleware, first built for large scale experimentations of scheduling heuristics, in a production Grid is a real victory for our research team.

The Décrypthon have been presented at the SuperComputing 2008 exhibitor in Austin, Texas as a DIET use case of the INRIA booth.

4.8. Micro-Factories

Micro-factories are automated units designed to produce pieces composed of micro-metric elements. Today's micro-factories are composed of elementary modules or robots able to carry out basic operations. To perform more complex operations, few elementary modules may be grouped in a cell. The realization of one of these cells is still a scientific challenge but several research projects have already got significant results in this domain. These results show very promising functionalities like the ability to configure or reconfigure a cell, by changing a robot tool for instance. However, the set of operations carried out by a cell is still limited. The next generation of micro-factories will put several cells together and make them cooperate to produce complex assembled pieces, as we do for macroscopic productions. In this context, the cell control will evolve to become more cooperative and distributed.

Micro-factories may be modelled in a way that allows to reuse the results obtained in scheduling on heterogeneous platforms as Grids, in particular the results on steady-state scheduling. We develop scheduling strategies and algorithms adapted to this context and we optimize the deployment of cells based on the micro-product and the production specification. We are currently working on the evaluation and the adaptation of several scheduling algorithms in this context, taking small-to-medium batch of jobs into account.

At the micro-metric scale, the manipulation of the elements cannot be considered the same way as at macro-metric scale because the equilibrium of forces is modified. For instance, the electrostatic force becomes predominant on the gravity. This lead to uncontrolled behaviors and frequently generates faults. We are working on taking these faults into account into scheduling models and evaluating their performance depending on the fault characteristics.

5. Software

5.1. DIET

Participants: Nicolas Bard, Raphaël Bolze, Yves Caniou, Eddy Caron, Ghislain Charrier, Frédéric Desprez [correspondent], Jean-Sébastien Gay, Vincent Pichon, Cédric Tedeschi.

Huge problems can now be processed over the Internet thanks to Grid Computing Environments like Globus or Legion. Because most of the current applications are numerical, the use of libraries like BLAS, LAPACK, ScaLAPACK, or PETSc is mandatory. The integration of such libraries in high level applications using languages like Fortran or C is far from being easy. Moreover, the computational power and memory needs of such applications may of course not be available on every workstation. Thus, the RPC paradigm seems to be a good candidate to build Problem Solving Environments on the Grid as explained in Section 3.3. The aim of the DIET project (http://graal.ens-lyon.fr/DIET) is to develop a set of tools to build computational servers accessible through a GridRPC API.

Moreover, the aim of a NES environment such as DIET is to provide a transparent access to a pool of computational servers. DIET focuses on offering such a service at a very large scale. A client which has a problem to solve should be able to obtain a reference to the server that is best suited for it. DIET is designed to take into account the data location when scheduling jobs. Data are kept as long as possible on (or near to) the computational servers in order to minimize transfer times. This kind of optimization is mandatory when performing job scheduling on a wide-area network.

DIET is built upon *Server Daemons*. The scheduler is scattered across a hierarchy of *Local Agents* and *Master Agents*. Network Weather Service (NWS) [104] sensors are placed on each node of the hierarchy to collect resource availabilities, which are used by an application-centric performance prediction tool named FAST.

The different components of our scheduling architecture are the following. A **Client** is an application which uses DIET to solve problems. Many kinds of clients should be able to connect to DIET from a web page, a Problem Solving Environment such as Matlab or Scilab, or a compiled program. A **Master Agent (MA)** receives computation requests from clients. These requests refer to some DIET problems listed on a reference web page. Then the MA collects computational abilities from the servers and chooses the best one. The reference of the chosen server is returned to the client. A client can be connected to an MA by a specific name server or a web page which stores the various MA locations. Several MAs can be deployed on the network to balance the load among them. A **Local Agent (LA)** aims at transmitting requests and information between MAs and servers. The information stored on a LA is the list of requests and, for each of its subtrees, the number of servers that can solve a given problem and information about the data distributed in this subtree. Depending on the underlying network topology, a hierarchy of LAs may be deployed between an MA and the servers. No scheduling decision is made by a LA. A **Server Daemon (SeD)** encapsulates a computational server. For instance it can be located on the entry point of a parallel computer. The information stored on a SeD is a list of the data available on its server (with their distribution and the way to access them), the list of problems that can be solved on it, and all information concerning its load (available memory and resources, etc.). A SeD declares the problems it can solve to its parent LA. A SeD can give performance prediction for a given problem thanks to the CoRI module (Collector of Resource Information) [88].

Moreover applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via *plug-in schedulers* [88]. As the applications that are to be deployed on the Grid vary greatly in terms of performance demands, the DIET plug-in scheduler facility permits the application designer to express application needs and features in order that they be taken into account when application tasks are scheduled. These features are invoked at runtime after a user has submitted a service request to the MA, which broadcasts the request to its agent hierarchy.

Master Agents can then be connected over the net (Multi-MA version of DIET), either statically or dynamically.

Thanks to a collaboration between the GRAAL and PARIS projects, DIET can use *JuxMem*. *JuxMem* (Juxtaposed Memory) is a peer-to-peer architecture developed by the PARIS team which provides memory sharing services allowing peers to share memory data, and not only files. To illustrate how a *GridRPC* system can benefit from transparent access to data, we have implemented the proposed approach inside the DIET *GridRPC* middleware, using the *JuxMem* data-sharing service.

Tools have recently been developed to deploy the platform (GoDIET), to monitor its execution (LogService), and to visualize its behavior using Gantt graphs and statistics (VizDIET).

Seen from the user/developer point of view, the compiling and installation process of DIET should remain simple and robust. But DIET has to support this process for an increasing number of platforms (Hardware architecture, Operating System, C/C++ compilers). Additionally DIET also supports many functional extensions (sometimes concurrent) and many such extensions require the usage of one or a few external libraries. Thus the compilation and installation functionalities of DIET must handle a great number and variety of possible specific configurations. Up to the previous versions, DIET's privileged tool for such a task were the so-called GNU-autotools. DIET's autotools configuration files evolved to become fairly complicated and hard to maintain. Another important task for the packaging DIET is to assess that DIET can be properly compiled and installed at least for the most mainstream platforms and for a decent majority of all extension combinations. This quality assertion process should be realized with at least the frequency of the release. But, as clearly stated by the agile software development framework, the risk can be greatly reduced by developing software in short time-boxes (as short as a single cvs commit). For the above reasons, it was thus decided to move away from the GNU-autotools to cmake (refer http://www.cmake.org). Cmake offers a much simpler syntax for its configuration files (sometimes at the cost of semantics, but cmake remains an effective trade-off). Additionally, cmake integrates a scriptable regression test tool whose reports can be centralized on a so called dashboard server. The dashboard offers a synthetic view (see http://graal.ens-lyon.fr/DIET/dietdashboard.html) of the current state of DIET's code. This quality evaluation is partial (compilation and linking errors and warnings) but is automatically and constantly offered to the developers. Although the very nature of DIET makes it difficult to carry distributed regression tests, we still hope that the adoption of cmake will significantly improve DIET's robustness and general quality.

DIET has been validated on several applications. Some of them have been described in Sections 4.2 through 4.7.

5.1.1. Workflow support

Workflow-based applications are scientific, data intensive applications that consist of a set of tasks that need to be executed in a certain partial order. These applications are an important class of Grid applications and are used in various scientific domains like astronomy or bioinformatics.

We have developed a workflow engine in DIET to manage such applications and propose to the end-user and the developer a simple way either to use provided scheduling algorithms or to develop their own scheduling algorithm.

There are many Grid workflow frameworks that have been developed, but DIET is the first GridRPC middleware that provides an API for workflow execution. Moreover, existing tools have limited scheduling capabilities. One of our objectives is to provide an open system which provides several scheduling algorithms, but also that allows users to plug and use their own specific schedulers.

In our implementation, workflows are described using the XML language. Since no standard exists for scientific workflows, we have proposed our formalism. The DIET agent hierarchy has been extended with a new special agent, the *MA_DAG*. To be flexible we can execute workflows even if this special agent is not present in the platform. The use of the *MA_DAG* centralizes the scheduling decisions and thus can provide a better scheduling when the platform is shared by multiple clients. On the other hand, if the client bypasses the *MA_DAG*, a new scheduling algorithm can be used without affecting the DIET platform. The current implementation of DIET provides several schedulers (Round Robin, HEFT, random, Fairness on finish Time, etc.).

The DIET workflow runtime also includes a rescheduling mechanism. Most workflow scheduling algorithms are based on performance predictions that are not always accurate (erroneous prediction tool or resource load wrongly estimated). The rescheduling mechanism can trigger the application rescheduling when some conditions specified by the client are filled.

This year, more developments were performed to improve DIET workflow engine especially considering multi-workflows based applications. The previous support allows to manage multiple workflow submissions but each submitted workflow was scheduled alone. To study the behavior of workflow scheduling, another approach was used that consider the submitted workflow with all other waiting tasks of previous submitted workflows to compute a new scheduling. A first implementation was realized by using monitoring features and mechanisms to make different workflows respect new scheduling decisions, but it was insufficient in a real concurrent environment. The second implementation that corrects this drawback uses a centralized scheduler in the *MA_DAG* (like in the first implementation) but also a minimal runtime to execute active workflows. This minimalist runtime does not actually execute tasks (to minimize *MA_DAG load*) but it triggers the corresponding clients to execute them, so scheduling decisions can be respected since scheduling and task execution start are done in a centralized way.

In addition to these developments, graphical tools for workflows (Workflow designer and Workflow log service) were developed within the DIET DashBoard project.

5.1.2. Batch and parallel job management

Generally, the use of a parallel computing resource is done through a batch reservation system. Users wishing to submit parallel tasks have to write *scripts* which notably describe the number of required nodes and the walltime of the reservation. Once submitted, a script is processed by the batch scheduling algorithm: the user is answered the starting time of its job, and the batch system records the dedicated nodes (*the mapping*) allocated to the job.

In the Grid context, there is consequently a two-level scheduling: one at the batch level and the other one at the Grid middleware level. In order to efficiently exploit the resource (according to some metrics), the Grid middleware should map the computing tasks according to the local scheduler policy. This also supposes that the middleware integrates some mechanisms to submit to parallel resources, and that, during the submission, it provides information like the number of demanded resources, the job deadline, etc.

DIET servers are able to transparently submit tasks to parallel resources, via a batch system or not. For the moment, DIET servers can submit to OAR, OpenPBS and Loadleveler reservation systems, the latter being used in the Décrypthon project. Functions to access batch system information have also been implemented in order to use them both as scheduling metric and to tune parallel and moldable tasks.

5.1.3. DIET Data Management

DAGDA, designed during the PhD of Gaël Le Mahec, is a new data manager for the DIET middleware which allows data explicit or implicit replications and advanced data management on the grid. It was designed to be backward compatible with previously developed applications for DIET which benefit transparently of the data replications. It allows explicit or implicit data replications, file sharing between the nodes which can access to the same disk partition, the choice of a data replacement algorithm, and a high level configuration about the memory and disk space DIET should use for the data storage and transfers.

To transfer a data, DAGDA uses the pull model instead of the push model used by DTM. The data are not sent into the profile from the source to the destination, but they are downloaded by the destination from the source. DAGDA also chooses the best source for a given data.

DAGDA has also been used for the validation of our join replication and scheduling algorithms over DIET.

5.1.4. GridRPC Data Management API

Data Management is a challenging issue inside the OGF GridRPC standard, for performance reasons. Indeed some temporarily data do not need to be transfered once computed and can reside on servers for example. We can also imagine that data can be directly transferred from one server to another one, without being transferred to the client in accordance to the GridRPC paradigm behavior.

We have consequently worked on a Data Management API which has been presented to all OGF sessions since OGF'21. The new proposal involves asynchronous transfers and will be presented during the OGF'25, in 2009.

5.1.5. DIET Dashboard

When the purpose is to monitor a Grid, or deploy a Grid middleware on it, several tasks are involved in the process. Managing the resources of a Grid: allocating resources, deploying nodes with defined operating systems, etc. Monitoring the Grid: getting the status of the clusters (number of available nodes in each state, number and main properties of each job, Gantt chart of the jobs history), the status of the jobs (number, status, owner, walltime, scheduled start, ganglia information of the nodes) present in the platform, etc. Managing Grid middleware in Grid environment: designing hierarchies (manually or automatically by matching resources on patterns), deploying them directly or through workflows of applications, etc.

The DIET Dashboard provides tools trying to answer these needs with an environment dedicated to the GridRPC middleware DIET and it consists of a set of graphical tools that can be used separately or together.

These tools can be divided in three categories:

DIET tools including tools to design and deploy DIET applications. The DIET Designer allows users to graphically design a DIET hierarchy. The DIET Mapping tool allows users to map the allocated Grid'5000 resources to a DIET application. The mapping is done in an interactive way by selecting the site then DIET agents or SeDs. And the DIET Deploy tool is a graphical interface to GoDIET intended for the deployment of DIET hierarchies.

Workflow tools including workflow designer and workflow log service. **The Workflow designer** is dedicated to workflow applications written in DIET. It gives users an easy way to design and execute workflows. The user can compose the available services and link them by drag-and-drop or load a workflow description file in order to reuse it. Finally it can be directly executed online. **The Workflow LogService** can be used to monitor workflows execution by displaying the DAG nodes of each workflow and their states.

Grid tools (aka GRUDU). These tools are used to manage, monitor, and access user Grid resources. **Displaying the status of the platform**: this feature provides information about clusters, nodes and jobs. **Resource allocation**: this feature provides an easy way to allocate resources by selecting from a Grid'5000 map the number of required nodes and defining time. The allocated resources can be stored and used with DIET mapping tool. **Resource monitoring** through the use of the Ganglia plugin that provides low-level information on every machines of a site (instantaneous data) or on every machines of a job (history of the metrics). **Deployment management** with a GUI for KaDeploy simplifying its use. **A terminal emulator** for remote connections to Grid'5000 machines and a File transfer manager to send/receive files to/from Grid'5000 frontends.

As the Grid tools can be a powerful help for the Grid'5000 users, these have been extracted to create GRUDU (Grid'5000 Reservation Utility for Deployment Usage) which aims at simplifying the access and the management of Grid'5000.

5.1.6. Middleware Interoperability

For the requirements of the GridTLSE project, DIET has been extended with a protocol interoperability with the ITBL middleware which manages Japanese computing resources in the JAEA (Japan Atomic Energy Agency). A demo has been presented in the INRIA booth at SuperComputing'08.

5.2. MUMPS

Participants: Emmanuel Agullo, Alfredo Buttari, Philippe Combes, Jean-Yves L'Excellent [correspondent].

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, mainly in collaboration with ENSEEIHT-IRIT (Toulouse, France), lots of developments have been done, to enhance the software with more functionalities and integrate recent research work. Recent developments also involve the INRIA project ScAlAppliX since the recruitment of Abdou Guermouche as an assistant professor at LaBRI, while CERFACS contributes to some research work.

MUMPS implements a direct method, the multifrontal method, and is a parallel code for distributed memory computers; it is unique by the performance obtained and the number of functionalities available, among which we can cite:

- various types of systems: symmetric positive definite, general symmetric, or unsymmetric,
- several matrix input formats: assembled or expressed as a sum of elemental matrices, centralized on one processor or pre-distributed on the processors,
- detection of null pivots,
- preprocessing and scaling for symmetric and unsymmetric matrices,

- partial factorization and Schur complement matrix,
- dense or sparse right-hand sides, centralized or distributed solution,
- real or complex arithmetic, single or double precision,
- partial threshold pivoting,
- fully asynchronous approach with overlap of computation and communication,
- distributed dynamic scheduling of the computational tasks to allow for a good load balance in the presence of unexpected dynamic pivoting or in multi-user environments.

MUMPS is currently used by more than 1000 academic and industrial users, from a wide range of application fields (see Section 4.1). Notice that the MUMPS users include:

- students and academic users from all over the world;
- various developers of finite element software;
- companies such as Boeing, EADS, EDF, Free Field Technologies, or Samtech.

From a geographical point of view, 31% of our users come from North America, 39% are Europeans, and 19% are from Asia.

The latest release is MUMPS 4.8.4, available since December 2008 (see http://graal.ens-lyon.fr/MUMPS/). The most recent features available are: detection of null pivots and estimate of a null space basis, parallel scaling algorithms, reduction of the memory usage (smaller communication buffers), more statistics returned to the user, and out-of-core storage of the factors. A parallel analysis phase based on PT-Scotch [89] is under development.

6. New Results

6.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Keywords: *Algorithm design*, *bioinformatics*, *divisible loads*, *heterogeneous platforms*, *load balancing*, *online scheduling*, *scheduling strategies*, *steady-state scheduling*.

Participants: Anne Benoît, Leila Ben Saad, Sékou Diakité, Alexandru Dobrila, Fanny Dufossé, Matthieu Gallet, Mourad Hakem, Mathias Jacquelin, Loris Marchal, Jean-Marc Nicod, Laurent Philippe, Jean-François Pineau, Veronika Rehn-Sonigo, Clément Rezvoy, Yves Robert, Bernard Tourancheau, Frédéric Vivien.

6.1.1. Mapping simple workflow graphs

Mapping workflow applications onto parallel platforms is a challenging problem, that becomes even more difficult when platforms are heterogeneous —nowadays a standard assumption. A high-level approach to parallel programming not only eases the application developer's task, but it also provides additional information which can help realize an efficient mapping of the application. We focused on simple application graphs such as linear chains and fork patterns. Workflow applications are executed in a pipeline manner: a large set of data needs to be processed by all the different tasks of the application graph, thus inducing parallelism between the processing of different data sets. For such applications, several antagonist criteria should be optimized, such as throughput, latency, and failure probability.

We have considered the mapping of workflow applications onto different types of platforms: *Fully Homogeneous* platforms with identical processors and interconnection links; *Communication Homogeneous* platforms, with identical links but processors of different speeds; and finally, *Fully Heterogeneous* platforms.

For linear chain graphs, we have extensively studied the complexity of the mapping problem, for throughput and latency optimization criteria. Different mapping policies have been considered: the mapping can be required to be one-to-one (a processor is assigned at most one stage), or interval-based (a processor is assigned an interval of consecutive stages), or fully general. The most important new result this year is the NP-completeness of the latency minimization problem for interval-based mappings on Fully Heterogeneous platforms, which was left open in our previous study. Furthermore, we proved that this problem, together with the similar one-to-one mapping problem, cannot be approximated by any constant factor (unless P=NP).

Once again, this year we have focused mainly on pipeline graphs, and considered platforms in which processors are subject to failure during the execution of the application. We derived new theoretical results for bi-criteria mappings aiming at optimizing both the latency (*i.e.*, the response time) and the reliability (*i.e.*, the probability that the computation will be successful) of the application. Latency is minimized by using faster processors, while reliability is increased by replicating computations on a set of processors. However, replication increases latency (additional communications, slower processors). The application fails to be executed only if all the processors fail during execution. While simple polynomial algorithms can be found for fully homogeneous platforms, the problem becomes NP-hard when tackling heterogeneous platforms.

On the experimental side, we have designed and implemented several polynomial heuristics for different instances of our problems. Experiments have been conducted for pipeline application graphs, on Communication-Homogeneous platforms, since clusters made of different-speed processors interconnected by either plain Ethernet or a high-speed switch constitute the typical experimental platforms in most academic or industry research departments. We can express the problem of maximizing the throughput as the solution of an integer linear program, and thus we have been able to compare the heuristics with an optimal solution for small instances of the problem. For bi-criteria optimization problems, we have compared different heuristics through extensive simulations. Typical applications include digital image processing, where images are processed in steady-state mode. This year, we have thoroughly studied the mapping of a particular image processing application, the JPEG encoding. Mapping pipelined JPEG encoding onto parallel platforms is useful for instance for encoding Motion JPEG images. The performance of our bi-criteria heuristics has been validated on this application.

6.1.2. Mapping linear workflows with computation/communication overlap

In this joint work with Kunal Agrawal (MIT), we extend our work on linear pipelined workflows to a more realistic architectural model with bounded communication capabilities and full computation/communication overlap. This model is representative of current multi-threaded systems. We present several complexity results related to period and/or latency minimization.

To be precise, we prove that maximizing the period is NP-complete even for homogeneous platforms and minimizing the latency is NP-complete for heterogeneous platforms. Moreover, we present an approximation algorithm for throughput maximization for linear chain applications on homogeneous platforms, and an approximation algorithm for latency minimization for linear chain applications on all platforms where communication is homogeneous (the processor speeds can differ). In addition, we present algorithms for several important special cases for linear chain applications. Finally, we consider the implications of adding feedback to linear chain applications.

6.1.3. Energy-aware scheduling

We consider the problem of scheduling an application composed of independent tasks on a fully heterogeneous master-worker platform with communication costs. We introduce a bi-criteria approach aiming at maximizing the throughput of the application while minimizing the energy consumed by participating resources. Assuming arbitrary super-linear power consumption laws, we investigate different models for energy consumption, with and without start-up overheads. Building upon closed-form expressions for the uniprocessor case, we are able to derive optimal or asymptotically optimal solutions for both models.

6.1.4. Mapping filtering services

We explore the problem of mapping filtering services on large-scale heterogeneous platforms. Such applications can be viewed as regular workflow applications with arbitrary precedence graphs, with the additional property that each service (node) filters (shrinks or expands) its input data by a constant factor (its selectivity) to produce its output data. As always, period and/or latency minimization are the key objectives. For homogeneous platforms, the complexity of period minimization was already known; we derive an algorithm to solve the latency minimization problem in the general case with service precedence constraints; for independent services we also show that the bi-criteria problem (latency minimization without exceeding a prescribed value for the period) is of polynomial complexity. However, when adding heterogeneity to the platform, we prove that minimizing the period or the latency becomes NP-hard, and that these problems cannot be approximated by any constant factor (unless P=NP). The latter results hold true even for independent services. We provide an integer linear program to solve both problems in the heterogeneous case with independent services.

6.1.5. Resource allocation strategies for in-network stream processing

This year we studied the operator mapping problem for in-network stream processing applications. In-network stream processing consists in applying a tree of operators in steady-state to multiple data objects that are continually updated at various locations on a network. Examples of in-network stream processing include the processing of data in a sensor network, or of continuous queries on distributed relational databases. We studied the operator mapping problem in a “constructive” scenario, i.e., a scenario in which one builds a platform dedicated to the application by purchasing processing servers with various costs and capabilities. The objective is to minimize the cost of the platform while ensuring that the application achieves a minimum steady-state throughput. We have formalized a set of relevant operator-placement problems as linear programs, and proved that even simple versions of the problem are NP-complete. Also, we have designed several polynomial time heuristics, which are evaluated via extensive simulations and compared to theoretical bounds for optimal solutions.

6.1.6. Scheduling small to medium batches of identical jobs

Steady-state scheduling is optimal for an infinite number of jobs. It defines a schedule for a subset of jobs which are performed into a period. The global schedule is obtained by infinitely repeating this period. In the case of a finite number of jobs, this scheduling technique can however be used if the number of computed jobs is large. Three phases are distinguished in the schedule: an initialization phase which computes the tasks needed to enter the steady state, an optimal phase composed of several full periods and a termination phase which finishes the tasks remaining after the last period. With a finite number of jobs we must consider a different objective function, the makespan, instead of the throughput used in the steady-state case. We know that the steady-state phase of the schedule is optimal, thus we are interested in optimizing the initial and final phases.

We have worked on the improvement of the steady-state technique for a finite number of jobs. The main idea is to improve the scheduling of the sub-optimal phases: initialization and termination. By optimizing these two phases we reduce their weight in the global schedule and thus improve its performance. In the original algorithm the period is computed by a linear program. As a result, the period’s length can be quite large, resulting in a lot of temporary job instances. Each of these temporary job instances must be prepared in the initialization phase, and finished in the termination phase. To reduce initialization and termination phase, we propose to limit the period length. Another possible optimization is to better organize a given period to reduce the number of temporary instances, by transforming inter-period dependences into intra-period dependences when possible. Both propositions have been studied and implemented in the SimGrid toolkit and we are conducting experiences to evaluate their efficiency.

6.1.7. Steady-scheduling of task graph collections on heterogeneous resources

In this work, we focused on scheduling jobs on computing Grids. In our model, a Grid job is made of a large collection of input data sets, which must all be processed by the same task graph or *workflow*, thus resulting in a *collection of task graphs* problem. We are looking for a competitive scheduling algorithm not requiring complex control. We thus only consider single-allocation strategies. We present an algorithm based on mixed linear programming to find an optimal allocation, and this for different routing policies depending on how much latitude we have on routing communications. Then, using simulations, we compare our allocations to optimal multi-allocation schedules. Our results show that the single-allocation mixed-linear program approach almost always finds an allocation with a reasonably-good throughput, especially under communication-intensive scenarios.

In addition to the mixed linear programming approach, we present different heuristic schemes. Then, using simulations, we compare the performance of our different heuristics to the performance of a classical scheduling policy in Grids, HEFT. The results show that some of our static-scheduling policies take advantage of their platform and application knowledge and outperform HEFT, especially under communication-intensive scenarios. In particular, one of our heuristics, DELEGATE, almost always achieves the best performance while having lower running times than HEFT.

6.1.8. Fault-tolerant scheduling of precedence task graphs

Heterogeneous distributed systems are widely deployed for executing computationally intensive parallel applications with diverse computing needs. Such environments require effective scheduling strategies that take into account both algorithmic and architectural characteristics. Unfortunately, most of the scheduling algorithms developed for such systems rely on a simple platform model where communication contention is not taken into account. In addition, it is generally assumed that processors are completely safe. To schedule precedence graphs in a more realistic framework, we introduce first an efficient fault tolerant scheduling algorithm that is both contention-aware and capable of supporting an arbitrary number of fail-silent (fail-stop) processor failures. Next, we derive a more complex heuristic that departs from the main principle of the first algorithm. Instead of considering a single task (one with highest priority) and assigning all its replicas to the currently best available resources, we consider a chunk of ready tasks, and assign all their replicas in the same decision making procedure. This leads to a better load balance of processors and communication links. We focus on a bi-criteria approach, where we aim at minimizing the total execution time, or latency, given a fixed number of failures supported in the system. Our algorithms have a low time complexity, and drastically reduce the number of additional communications induced by the replication mechanism. Experimental results fully demonstrate the usefulness of the proposed algorithms, which lead to efficient execution schemes while guaranteeing a prescribed level of fault tolerance.

6.1.9. Static strategies for worksharing with unrecoverable interruptions

In this work, one has a large workload that is “divisible” and one has access to p remote computers that can assist in computing the workload. The problem is that the remote computers are subject to interruptions of known likelihood that kill all work in progress. One wishes to orchestrate sharing the workload with the remote computers in a way that maximizes the expected amount of work completed. Strategies for achieving this goal, by balancing the desire to checkpoint often, in order to decrease the amount of vulnerable work at any point, vs. the desire to avoid the context-switching required to checkpoint, are studied. Strategies are devised that provably maximize the expected amount of work when there is only one remote computer (the case $p = 1$). Results suggest the intractability of such maximization for higher values of p, which motivates the development of heuristic approaches. Heuristics are developed that replicate works on several remote computers, in the hope of thereby decreasing the impact of work-killing interruptions. The quality of these heuristics is assessed through exhaustive simulations.

6.1.10. Scheduling identical jobs with unreliable tasks

Depending on the context, the fault tolerance model may differ. We have studied the case where the fault probability depends on tasks instead of on the execution resource. The practical use case is a micro-factory where operations are performed on microscopic components. Due to the size of the components, some operations are not as well controlled as the others and thus the complexity of the task impacts on its reliability. In this context, we consider the schedule of a set of identical jobs composed of either linear chains or trees of tasks. Several objectives are studied depending on the available resources, in particular maximizing the throughput (number of components output per time unit), and minimizing the makespan (total time needed to output the required number of components). The resources in use are heterogeneous and general purpose but must be configured to execute a determined task type. For this reason, finding a good schedule turns into an assignment problem. The most simple instances of this problem can be solved in polynomial time whereas the other cases are NP-complete; for those cases, we designed polynomial heuristics to solve the problem.

6.1.11. Resource allocation using virtual clusters

We propose a novel approach for sharing cluster resources among competing jobs. The key advantage of our approach over current cluster sharing solutions is that it increases cluster utilization while optimizing a user-centric metric that captures both notions of performance and fairness. We motivate and formalize the corresponding resource allocation problem, determine its complexity, and propose several algorithms to solve it in the case of a static workload that consists of sequential jobs. Via extensive simulation experiments we identify an algorithm that runs quickly, that is always on par with, or better than, its competitors, and that produces resource allocations that are close to optimal. We find that the extension of our approach to workloads that comprise parallel jobs leads to similarly good results. Finally, we explain how to extend our work to handle dynamic workloads.

6.1.12. Parallelizing the construction of the ProDom database

ProDom is a protein domain family database automatically built from a comprehensive analysis of all known protein sequences. ProDom development is headed by Daniel Kahn (INRIA project-team HELIX). With the protein sequence databases increasing in size at an exponential pace, the parallelization of MkDom2, the algorithm used to build ProDom, has become mandatory (the original sequential version of MkDom2 took 15 months to build the 2006 version of ProDom and would have required at least twice that time to build the 2007 version).

The parallelization of MkDom2 is not a trivial one. The sequential MkDom2 algorithm is an iterative process and parallelizing it involves forecasting which of these iterations can be run in parallel and detecting and handling dependency breaks when they arise. We have moved forward to be able to efficiently handle larger databases. Such databases are prone to exhibit far larger variations in the processing time of query-sequences than was previously imagined. The collaboration with HELIX on ProDom continues today both on the computational aspects of the constructing of ProDom on distributed platforms, as well as on the biological aspects of evaluating the quality of the domains families defined by MkDom2, as well as the qualitative enhancement of ProDom.

6.1.13. MPI in a sensor network

We study the potential interest of using the MPI communication library in the distributed system made by the networked micro-controlers within a sensor network. We follow the IETF standardization groups dealing with IP for sensor networks. We are currently developing an IP stack for sensor networks with the 6LoWPAN specifications. Our design originality is modularity in order to be able to experiment with several routing modules. This is especially necessary for the test and validation of our multi-sink multi-position theoretical approaches where the route choices are scheduled depending on the sinks' locations in order to increase the lifespan of the overall sensors network. Our target assumptions and testbeds are real routing in buildings and urban environment where sinks' locations are limited to seldom powered and networked locations. Moreover, the sinks re-location frequency is very low because of the man made operation costs.

6.2. Providing access to HPC servers on the Grid

Keywords: *Grid computing*, *Numerical computing*, *computing server*, *performance forecasting*.

Participants: Nicolas Bard, Raphaël Bolze, Yves Caniou, Eddy Caron, Ghislain Charrier, Benjamin Depardon, Frédéric Desprez, Jean-Sébastien Gay, David Loureiro, Vincent Pichon, Cédric Tedeschi.

6.2.1. Workflow scheduling

In many scientific areas, such as high-energy physics, bioinformatics, astronomy, and others, we encounter applications involving numerous simpler components that process large data sets, execute scientific simulations, and share both data and computing resources. Such data intensive applications consist of multiple components (tasks) that may communicate and interact with each other. The tasks are often precedence-related. Data files generated by one task are needed to start another. This problem is known as workflow scheduling. Surprisingly the problem of scheduling multiple workflows online does not appear to be fully addressed. We study many heuristics based on list scheduling to solve this problem. We also implemented a simulator in order to classify the behaviors of these heuristics depending on the shape and size of the graphs. Some of these heuristics are implemented within DIET and tested with the bioinformatics applications involved in the Décrypthon program.

We also work on scheduling workflows when services involved in workflows are not necessarily present on all computing resources. In that case, there is a need to correctly schedule services in order not to see only short term performance: for example, a powerful resource may stay idle in order to later be available to process a job that can run on no other resource. Numerous heuristics have been designed, and we currently evaluate them before implementing them in the DIET Grid middleware.

Moreover during the collaboration with the associate team of the University of Hawai'i at Manoā, we worked on the integration of our grid middleware DIET with a bioinformatics project: ByOPortal. This project consists in providing to non computer scientists means of running computational tasks on DNA or protein databases. We developed a generic DIET client/server to abstract the communication level between the web portal and the computation resources. This client/server is able to deal with any kind of jobs described as a series of command lines which need to be executed either sequentially or in parallel with data dependences, thus describing a dataflow. The innovative part of this work is that it is able to dynamically manage the shape of the dataflow depending on the number of outputs each step provides (thus the parallelism level is not statically set when the workflow is submitted). This work is currently tested on a cluster in the University of Hawai'i.

6.2.2. Service Discovery in Peer-to-Peer environments

This work combines grid computing, P2P systems and falls within the field of their interactions for service discovery in grid computing. The DLPT (*Distributed Lexicographic Placement Table*) is a prefix tree structured overlay network, developed within the GRAAL project since 2005, that provides large scale discovery of computing services.

This year, on the theoretical side, we have concentrated on two aspects. First, we completed a solution to the problems of logical/physical association and load balancing within such a network using some mapping mechanisms and load balancing heuristics.

Second, we have developed new approaches for the problem of fault tolerance within such architectures. In collaboration with Ajoy K. Datta (University of Nevada, Las Vegas, USA), we have designed a self-stabilizing protocol in the realistic message-passing paradigm.

On the practical side, our DLPT software prototype, developed in collaboration with the INRIA project-team RESO, has been deployed on the Grid'5000 platform. Results of first experiments conducted on several clusters are promising.

6.2.3. Deployment for DIET: Software and Research

Concerning the deployment, we needed to provide a clustering among resources. Indeed, Grid environments, as well as mobile ad hoc networks are highly distributed, changeable and error prone environments. In order to improve communication efficiency within these platforms, i.e., minimize the communication costs, an often used approach is to group well connected nodes into clusters. Many centralized and distributed algorithms cluster the graphs according to a certain metric: the hop distance, or the weighted distance. We are interested more specifically in designing a k-clustering, that is to group the nodes into clusters such that within a cluster no node is further than k from a special node called clusterhead. In order to cope with errors and dynamicity of the platforms, designing self-stabilizing algorithms is a good approach. We designed the first self-stabilizing k-clustering algorithm on weighted graph, proved its correctness and complexity, and implemented a simulator to validate its efficiency. This work is a joint work with Ajoy K. Datta and Lawrence L. Larmore (University of Nevada, Las Vegas, USA).

6.2.4. Join Scheduling and Data Management

Usually, in existing Grid computing environments, data replication and scheduling are two independent tasks. In some cases, replication managers are requested to find best replicas in term of access costs. But the choice of the best replica has to be done at the same time as the schedule of computation requests. We first proposed an algorithm that computes at the same time the mapping of data and computational requests on these data using a linear program and a method to obtain a mixed solution, i.e., integer and rational numbers, of this program. However our results only held if the submitted requests precisely followed the usage frequencies given as an input for the static replication and scheduling algorithm. Due to the specificity of biological experiments, these frequencies may punctually change. To cope with those changes, we developed a dynamic algorithm and a set of heuristics that monitor the execution platform and take decision to move data and change scheduling of requests. The main goal of this algorithm is to balance the computation load between each server. Using the Optorsim simulator, we compared the results of the different heuristics. The conclusion of these simulations is that we have a set of heuristics that, in the case of our hypothesis, are able to reliably adapt the data placement and request scheduling to get an efficient usage of all computation resources.

In this previous work, we designed a scheduling strategy based on the hypothesis that, if you choose a large enough time interval, the proportion of a job using a given data is always the same. As observed in execution traces of bioinformatics clusters, this hypothesis seems to correspond to the way these clusters are generally used. However, this algorithm does not take into account the initial data distribution costs and, in its original version, the dynamicity of the submitted jobs proportions. We introduced algorithms that allow to get good performance as soon as the process starts and take care about the data redistribution when needed. We want to run a continuous stream of jobs, using linear-time algorithms that depend on the size of the data on which they are applied. Each job is submitted to a Resource Broker which chooses a Computing Element (CE) to queue the job on it. When a job is queued on a CE, it waits for the next worker node that can execute it, with a FIFO policy. These algorithms try to take into account the temporary changes in the usage of the platform and do not need to obtain dynamic information about the nodes (cpu load, free memory, etc.). The only information used to make the scheduling decisions is the frequency of each kind of job submitted. Thus, the only information needed by the scheduler is collected by the scheduler itself avoiding the use of complex platform monitoring services. In a next step, we will concentrate on the data redistribution process which is itself a non-trivial problem. We will study some redistribution strategies to improve the performance of the algorithms which dynamically choose where to replicate the data on the platform. Some large scale experiments have been already done on the Grid'5000 experimental platform using the DIET middleware. This work is done in collaboration with the PCSV team of the IN2P3 institute in Clermont-Ferrand.

6.2.5. Parallel Job Submission Management

We have used the DIET functionality to transparently submit parallel job to parallel resources in several experiments, some with *Ramses* (see Section 4.5) and others using the Décrypthon applications. A client/server for the LAMMPS software (see Section 4.2) is work in progress. Some mechanisms to tune moldable jobs have also been implemented and used with the Scotch sparse linear solver.

6.2.6. Scheduling and Deployment for Cosmological simulations

Cosmological simulations are parameter sweep applications, they require that the set of parameters is tested in order to find the best parameterization for the model. We are currently working on two particular softwares: GalaxyMaker and MoMaF, which purpose is to model the formation and evolution of galaxies. In order to be able to run such softwares on a grid environment, we developed a DIET client/server. It is able to dynamically spawn and delete services in order to improve the data management of the software: the amount of data produced in each intermediate step makes it hard to concurrently run many simulations, one needs to correctly manage the data migration and deletion. Deployment and scheduling algorithms for these workflows are currently being developed, but still need to be validated.

6.2.7. Grid Middleware Interoperability

In the context of the REDIMPS project, DIET has been extended with a protocol interoperability with the ITBL middleware which manages Japanese computing resources in the JAEA (Japan Atomic Energy Agency). A demo has been presented at the INRIA booth for Supercomputing'08.

6.3. Parallel Sparse Direct Solvers

Keywords: *direct solvers*, *graphs*, *memory usage*, *multifrontal method*, *out-of-core*, *scheduling*, *sparse matrices*.

Participants: Emmanuel Agullo, Alfredo Buttari, Philippe Combes, Jean-Yves L'Excellent.

6.3.1. Introduction

This year, we have pursued some work to add functionalities and improve the MUMPS software package, with strong interactions and informal collaborations with many users that provide challenging problems and help us validate and improve our algorithms: (i) industrial teams which experiment and validate our package, (ii) members of research teams with whom we discuss future functionalities wished, (iii) designers of finite element packages who integrate MUMPS as a solver for the internal linear systems, (iv) teams working on optimization, (v) physicists, chemists, etc., in various fields where robust and efficient solution methods are critical for their simulations. In all cases, we validate all our research and algorithms on large-scale industrial problems, either coming directly from MUMPS users, or from publicly available collections of sparse matrices (Davis collection, Rutherford-Boeing and PARASOL).

In the context of the Solstice project, funded by the ANR (French Research Agency), we have improved the algorithms for null pivot detection and we made them available in MUMPS 4.8.4. This is the object of collaborations with other partners of the project, including CERFACS and EDF, who is one of the main user of this functionality. We also worked closely with Bora Uçar (CERFACS) and Patrick Amestoy (ENSEEIHT-IRIT) on parallel scaling algorithms [85] and their integration into MUMPS. After feedback from several users on the version integrated in MUMPS 4.8.0 we have worked on an improved parallel algorithm that accelerates the scaling phase and we have tuned its default numerical behavior. The improved version is included in MUMPS 4.8.4. We collaborated with Luc Giraud (ENSEEIHT-IRIT) on hybrid direct-iterative solvers, providing a direct solver with the ability to return a Schur complement, that is then used within an iterative scheme based on domain decomposition. We also collaborate with ENSEEIHT-IRIT on an expertise site for sparse direct solvers, called GRID TLSE. The site has been used a lot to exchange test problems and sparse matrices with users of sparse direct solvers. The goal is to also provide scenarios allowing a user to experiment various combinations of algorithms and solvers on a user's typical test problems. More information can be obtained from http://gridtlse.org.

In the next two paragraphs, we give details on the work performed towards an efficient out-of-core factorization and a parallel analysis, two critical aspects when dealing with large sparse matrices on limited-memory computers.

6.3.2. Out-of-core Factorization

Although factorizing a sparse matrix is a robust way to solve large sparse systems of linear equations, such an approach is costly both in terms of computation and storage. When the storage required to process a matrix is greater than the amount of memory available on the platform, so-called out-of-core approaches have to be employed: disks extend the main memory to provide enough storage capacity. A first robust approach that stores factors on disk is now officially available within the MUMPS package. More research was done in the context of the PhD of Emmanuel Agullo, in which we have investigated both theoretical and practical aspects of such out-of-core factorizations. The MUMPS and SUPERLU software packages have been used to illustrate the difficulties on real-life matrices. First, we have proposed and studied various out-of-core models that aim at limiting the overhead due to data transfers between memory and disks on uniprocessor machines. To do so, we have revisited the algorithms to schedule the operations of the factorization and have proposed new memory management schemes to fit out-of-core constraints. Then we have focused on a particular factorization method, the multifrontal method, that we have pushed as far as possible in a parallel out-of-core context with a pragmatic approach. We have shown that out-of-core techniques allow to solve large sparse linear systems efficiently, and that a special attention must be paid to low-level I/O mechanisms; in particular we have shown that system I/O's have several drawbacks, that can be avoided by using direct I/O's together with an asynchronous approach at the application level.

When only the factors are stored on disks, a particular attention must be paid to temporary data, which remain in core memory. Therefore we started to rethink the whole schedule of the out-of-core parallel factorization with the objective to achieve a high scalability of the core memory usage.

This work was done in close collaboration with Abdou Guermouche (Université de Bordeaux and LaBRI). Work on supernodal methods and SuperLU was done in collaboration with Xiaoye S. Li (Lawrence Berkeley National Laboratory, Berkeley, USA).

6.3.3. Parallel Analysis

Although the analysis phase of a sparse direct solver only involves symbolic computations, it may still induce considerable computational and memory requirements. The parallelization of the analysis phase can thus provide significant benefits to the solution of large-scale problems and represents an essential feature on computer systems with limited memory capabilities. The core of the analysis phase consists of two main operations:

1. Elimination tree computation: this step provides a pivotal order that minimizes the fill-in generated at factorization time and identifies independent computational tasks that can be executed in parallel.
2. Symbolic factorization: simulates the actual factorization in order to estimate the memory that has to be allocated for the factorization phase.

In our approach we can use either PT-Scotch [89] or ParMETIS [96] for step 1; those packages return an index permutation and a separators tree which results from an ordering based on nested dissections. Based on this, we first select a number of subtrees that is equal to the number of working processors, and perform a symbolic factorization on each of these subtrees that is based on the usage of quotient graphs [83] to limit the memory consumption. Once every processor has finished with its subtree, the symbolic factorization of the unknowns associated to the top part of the tree is performed sequentially; because we use quotient graphs on entry, this still allows for a good overall memory scalability. We have observed that although PT-Scotch is slower than ParMETIS, the quality of the ordering it provides is considerably better and does not degrade with the degree of parallelism.

This work was presented in [82]. We now plan to include it in MUMPS and make it available in a future release of the package.

7. Contracts and Grants with Industry

7.1. Contract with SAMTECH, 2008-2010

INRIA and INPT-IRIT have signed a new contract with the company Samtech S.A. (Belgium). Samtech develops the European finite element software package SAMCEF, which uses our parallel sparse direct solver MUMPS as one of the internal solvers. The goal of this contract is to improve the memory usage of MUMPS, and to offer the possibility to address a larger amount of memory. We will also study how to use memory already allocated by SAMCEF instead of having the solver allocate its own memory. Finally we also plan to study how performance can be improved on Samtech problems by allowing the forward substitution step to be performed simultaneously with the matrix factorization. This last point is particularly interesting in the case of out-of-core executions.

The contract is 24-month long, and the new functionalities developed in MUMPS for this contract will be made available in a future public release of the package.

In Lyon, Emmanuel Agullo, Alfredo Buttari and Jean-Yves L'Excellent participate to this contract. Bora Uçar (arrival January 1, 2009) will also participate.

8. Other Grants and Activities

8.1. Regional Projects

8.1.1. Pôle Scientifique de Modélisation Numérique (PSMN)

This federation of laboratories aims at sharing the parallel machines from ENS Lyon/PSMN and experiences of parallelization of applications.

J.-Y. L'Excellent participates to this project.

8.1.2. MUSINE: Franche-Comté: conception, validation et pilotage de la micro-usine multi-cellulaire (2007-2009)

The aim of this project is to design the information model and management (scheduling) part of a micro-factory composed of cells. Each cell contains a set of micro-robots which manipulate micro-products (about 10^{-5} meters). The project is in collaboration with the LAB (Laboratoire d'Automatique de Besançon).

L. Philippe leads the MUSINE project and J.-M. Nicod participates to it.

8.1.3. Projet "Calcul Hautes Performances et Informatique Distribuée"

F. Desprez leads (with E. Blayo from LMC, Grenoble) the "Calcul Hautes Performances et Informatique Distribuée" project of the cluster "Informatique, Signal, Logiciels Embarqués". Together with several research laboratories from the Rhône-Alpes region, we initiate collaborations between application researchers and distributed computing experts. A Ph.D. thesis (J.-S. Gay) focuses on the scheduling problems for physics and bioinformatic applications.

Y. Caniou, E. Caron, F. Desprez, J.-Y. L'Excellent, J.-S. Gay, and F. Vivien participate to this project.

8.2. National Contracts and Projects

8.2.1. ANR grant: ALPAGE (ALgorithmique des Plates-formes À Grande Échelle), 3 years, 2005-2008

Please see the 2007 activity report for a description of the Alpage project. 2008 was the last year of the project, which has been quite successfully evaluated by ANR. A final workshop will take place in March 2009.

Yves Robert was leading the Rhône-Alpes site of this project. Anne Benoit, Loris Marchal and Frédéric Vivien were the permanent members who participated to this project.

8.2.2. ANR grant: Stochagrid (Scheduling algorithms and stochastic performance models for workflow applications on dynamic Grid platforms), 3 years, ANR-06-BLAN60192-01, 2007-2010

Please see the 2007 activity report for a description of the Stochagrid project. In the first 12 months, we have investigated timed-Petri nets to model the mapping of workflows with stage replication, and we have succeeded in deriving an optimal polynomial algorithm to compute the period in the bounded multi-port model with overlap. Quite interestingly, the period is no longer the bottleneck resource, the critical path becomes more complex. The latter work was conducted in collaboration with Bruno Gaujal (LIG Grenoble). We have also investigated several multi-criteria algorithms and heuristics. Anne Benoit and Yves Robert are leading this project.

The project is entirely conducted within the GRAAL team (Anne Benoit and Yves Robert are the permanent members involved).

8.2.3. ANR grant CICG-05-11: LEGO (League for Efficient Grid Operation), 3 years, 2006-2008

The aim of this project is to provide algorithmic and software solutions for large scale architectures; our focus is on performance issues. The software component provides a flexible programming model where resource management issues and performance optimizations are handled by the implementation. On the other hand, current component technology does not provide adequate data management facilities, needed for large data in widely distributed platforms, and does not deal efficiently with dynamic behaviors. We choose three applications: ocean-atmosphere numerical simulation, cosmological simulation, and sparse matrix solver. We propose to study the following topics: Parallel software component programming; Data sharing model; Network-based data migration solution; Co-scheduling of CPU, data movement and I/O bandwidth; High-perf. network support. The Grid'5000 platform provides the ideal environment for testing and validation of our approaches.

E. Caron is leading the project, which comprises six teams: GRAAL/LIP (Lyon), PARIS/IRISA (Rennes), RUNTIME/LaBRI (Bordeaux), ENSEEIHT/IRIT (Toulouse), CERFACS (Toulouse) and CRAL/ENS-Lyon (Lyon). A. Amar, R. Bolze, Y. Caniou, F. Desprez, J.-S. Gay and C. Tedeschi also participate to this project.

8.2.4. ANR grant ANR-06-CIS-010: SOLSTICE (SOlveurs et simulaTIon en Calcul Extrême), 3 years, 2007-2009

The objective of this project is to design and develop high-performance parallel linear solvers that will be efficient to solve complex multi-physics and multi-scale problems of very large size (10 to 100 millions of equations). To demonstrate the impact of our research, the work produced in the project will be integrated in real simulation codes to perform simulations that could not be considered with today's technologies. This project also comprises LaBRI (coordinator), CERFACS, INPT-IRIT, CEA-CESTA, EADS-CCR, EDF R&D, and CNRM. We are more particularly involved in tasks related to out-of-core factorization and solution, parallelization of the analysis phase of sparse direct solvers, rank detection, hybrid direct-iterative methods and expertise site for sparse linear algebra.

Emmanuel Agullo, Alfredo Buttari, Philippe Combes and Jean-Yves L'Excellent participate to this project. Bora Uçar (arrival January 1, 2009) will also participate.

8.2.5. ANR grant ANR-06-MDCA-009: Gwendia (Grid Workflow Efficient Enactment for Data Intensive Applications), 3 years, 2007-2009

The objective of the Gwendia[5] project is to design and develop workflow management systems for applications involving large amounts of data. It is a multidisciplinary project involving researchers in computer science (including GRAAL) and in life science (medical imaging and drug discovery). Our work consists in designing algorithms for the management of several workflows in distributed and heterogeneous platforms and to validate them within DIET on the Grid'5000 platform.

8.2.6. SEISCOPE Consortium (2006-2008)

The SEISCOPE project coordinated by Geosciences Azur focuses on wave propagation problems and seismic imaging. Our parallel solver MUMPS has been used extensively for finite-difference modeling of acoustic wave propagation. We also started using the large-scale test problems arising from this project to design and experiment our out-of-core approaches. The SEISCOPE project is supported by the ANR (French Research Agency), and by BP, CGG, SHELL and TOTAL.

Emmanuel Agullo and Jean-Yves L'Excellent participated to this collaboration.

[5] http://gwendia.polytech.unice.fr/doku.php

8.3. European Contracts and Projects

8.3.1. NoE CoreGRID (2004-2008)

The CoreGRID Network of Excellence aims at building a European-wide research laboratory that will achieve scientific and technological excellence in the domain of large-scale distributed, Grid, and Peer-to-Peer computing. The primary objective of the CoreGRID Network of Excellence is to build solid foundations for Grid and Peer-to-Peer computing both on a methodological basis and a technological basis. This will be achieved by structuring research in the area, leading to integrated research among experts from the relevant fields, more specifically distributed systems and middleware, programming models, knowledge discovery, intelligent tools, and environments.

GRAAL is involved in CoreGRID under the partner CNRS. The CNRS partnership involves Algorille in Nancy (E. Jeannot), MOAIS in Grenoble (G. Huard, D. Trystram), and the Graal project (A. Benoit, Y. Caniou, E. Caron, F. Desprez, Y. Robert, F. Vivien). F. Vivien is responsible for the CoreGRID contract within CNRS. He is responsible for managing the three teams involved in the partner CNRS, and for representing them in the CoreGRID Members General Assembly. F. Vivien is a member of the CoreGRID Integration Monitoring Committee. He is also responsible for a task in the scheduling workpackage.

The Coregrid project will continue through an ERCIM Working Group leaded by F. Desprez.

8.4. International Contracts and Projects

8.4.1. France-Berkeley Fund Award (2008-2009)

In the framework of the France-Berkeley Fund, we have been awarded a research grant to enable an exchange program involving both young and confirmed scientists. The project focuses on massively parallel solvers for large sparse matrices and will reinforce the collaboration initiated by Emmanuel Agullo. Emmanuel Agullo, Alfredo Buttari and Jean-Yves L'Excellent participate to this project. On the French side, this project also involves P. Amestoy (ENSEEIHT-IRIT), A. Guermouche (LaBRI), I. Duff (CERFACS), and Bora Uçar (CERFACS, joining GRAAL January 2009).

8.4.2. REDIMPS (2007-2009)

REDIMPS (Research and Development of International Matrix Prediction System) is a project funded by the Strategic Japanese-French Cooperative Program on "Information and Communications Technology including Computer Science" with the CNRS and the JST. The goal of this international collaboration is building an international sparse linear equation solver expert site. Among the objectives of the project, one resides in the cooperation of the TLSE partners and the JAEA in the testing, the validation and the promotion of the TLSE system that is currently released. JAEA, who is one of the leading institute and organization of Japanese HPC, is studying high-performance numerical simulation methods on novel supercomputers, and is expecting to find the best linear solver within this collaboration. By integrating knowledge and technology of JAEA and TLSE partners, it is expected that we will achieve the construction of an international expert system for sparse linear algebra on an international grid computing environment.

Thanks to additional funding from INRIA's "explorateur" program, Y. Caniou spent one month two times at the Japan Atomic Energy Agency in Tokyo, Japan. He worked on the AEGIS-DIET Grid system interoperability.

Yves Caniou, Eddy Caron, Frédéric Desprez, and Jean-Yves L'Excellent participate to this project.

8.4.3. CNRS-USA grant SchedLife, University of Hawai'i (2007-2009)

We have been awarded a CNRS grant in the framework of the CNRS/USA funding scheme, which runs for three years starting in 2007. The collaboration is done with the Concurrency Research Group (CoRG) of Henri Casanova, and the Bioinformatics Laboratory (BiL) of Guylaine Poisson of the Information and Computer Sciences Department, of the University of Hawai'i at Manoā, USA.

The SchedLife project targets the efficient scheduling of large-scale scientific applications on clusters and Grids. To provide context for this research, we focus on applications from the domain of bioinformatics, in particular comparative genomics and metagenomics applications, which are of interest to a large user community today. So far, applications (in bioinformatics or other fields) that have been successfully deployed at a large scale fall under the "independent task model": they consist of a large number of tasks that do not share data and that can be executed in any order. Furthermore, many of these application deployments rely on the fact that the application data for each task is "small", meaning that the cost of sending data over the network can be ignored in the face of long computation time. However, both previous assumptions are not valid for all applications, and in fact many crucial applications, such as the aforementioned bioinformatics applications, require computationally dependent tasks sharing very large data sets.

In our previous collaborations, we have tackled the issue of non-negligible network communication overheads and have made significant contributions. For instance, we have designed strategies that rely on the notions of steady-state scheduling (i.e., attempting to maximize the number of tasks that complete per time unit, in the long run) and/or divisible load scheduling (i.e., approximate the discrete workload that consists of individual tasks as a continuous workload). These strategies provide powerful means for rethinking the deployment and the scheduling of independent task applications when network communication can be a bottleneck. However, the target applications in this project cannot benefit from these strategies directly and will require fundamental advances. This project aims to build upon and go beyond our past collaborations, with two main research thrusts:

- Scheduling of applications with data requirements. We consider applications that require possibly multiple data files that need to be shared by multiple application tasks. These files may be extremely large (e.g., millions of genomic sequences) and may need to be updated frequently (e.g., when new sequences are identified). We must then ensure that file access is not a bottleneck.
- Scheduling of multiple concurrent applications. We also plan to study the scheduling for multiple applications, i.e., launched by different (most likely competing) users. We then aim to orchestrate computation and communication in order to have the best aggregate performance. This is a difficult problem, first in order to define a good performance metric, and then to maximize this performance metric in a tractable way.

A. Benoit, E. Caron, F. Desprez, Y. Robert and F. Vivien participate to this project.

8.4.4. Associated-team MetagenoGrid (2008-2010)

This associated-team involves the exact same persons, and covers the same subject, as the CNRS-USA grant SchedLife described above.

8.4.5. Marie Curie Action – IOF – MetagenoGrids

In the scope of the above associated-team, Frédéric Vivien is on sabbatical at the University of Hawai‘i at Manoā for one year, starting July 17, 2008. This sabbatical is in part funded by a Marie Curie Action – IOF from the European Commission.

9. Dissemination

9.1. Scientific Missions

Open Grid Forum. The objective of the Open Grid Forum working group on "Grid Remote Procedure Call" (GridRPC) is to define a standard for this way to use Grid resources. E. Caron is co-chair of this OGF working group. E. Caron and Y. Caniou participated to the elaboration of a GridRPC Data Management API. F. Desprez is also involved in this working group.

9.2. Edition and Program Committees

Anne Benoit co-organized the Fifth International Workshop on aPplications of declArative and object-oriented Parallel Programming (PAPP 2008), Krakow, Poland, June 2008; she is co-organizing the sixth edition of the workshop PAPP 2009 in Baton Rouge, Louisiana, USA, May 2009.

A. Benoit was a member of the program committee of ICCS 2008, SBAC-PAD 2008, IPDPS 2008. She is a member of the program committee of ICCS 2009, HPCC 2009 and ISPDC 2009.

Yves Caniou is a member of the program committee of Heterogeneous Computing Workshop 2008, of the Mardi Gras Conference 2008, and of the ICCSA'08 conference. He is also involved in the CCGRID'08 conference as Local arrangement chair.

Eddy Caron was a member of the program committee of RENPAR 2008, PDP 2008, HCW 2008, MGC 2008 and ISPA 2008. He was co-chair of Tutorial Sesion for CCGRID'2008, and is co-chair of Grid-RPC group in the OGF (Open Grid Forum). He was session co-chair during OGF'22 (February 2008) and OGF'23 (June 2008).

Frédéric Desprez is member of the EuroPar Advisory board, the editorial board of "Scalable Computing: Practice and Experience" (SCPE) and *Computing Letters* (COMPULETT).

F. Desprez participated to the program committees of ICCS'08, CLADE'08, ICCSA'08, Grid'08, VecPar'08, CCGRID'08, PCGrid (held in conjunction with IPDPS), 9th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC-08) et 9th IEEE International Conference on Computational Science and Engineering, Modern computer tools for the biosciences (within CCGRID'08), CoreGRID Symposium (within Europar'08).

F. Desprez was the vice-chair of the IEEE CCGRID conference[6] which was held in Lyon in may 2008.

F. Desprez gave an invited talk at CCGSC Workshop, Ashville, USA, September 2008 and a tutorial about data management for Grids and Clouds to the CDUR Workshop (within the Notere Conference held in Lyon in 2008).

Jean-Yves L'Excellent was a member of the program committee of IPDPS'08 (Miami, Florida). He was also a member of the program committee of CSE'08 (Sao Paulo, Brazil).

Gilles Fedak organized the XtremWeb Users Group Meeting in Orsay and co-chaired 2 workshops PCGRID'08 and GP2PC'08 associated respectively with IPDPS and CCGRID. He was a member of the program committees of the following conferences and workshops : SBAC-PAD'08, DAPSYS'08, CDUR'08, GP2PC'08, Renpar 18, Euromicro PDP'08.

Loris Marchal is a member of the program committee of ISPDC 2009. He organized the third workshop on Scheduling for Large Scale Platforms, in Aussois (France), on May 18-21 2008. This workshop was held in cooperation with École Normale Supérieure de Lyon, the University of California at San Diego and the University of Hawai'i at Manoā. Following this workshop, a special issue of the Parallel Computing journal will be edited. Frédéric Vivien and Loris Marchal are co-editors of this special issue.

Christian Pérez co-organized the third CompFrame workshop on Component-Based High Performance Computing, Karlruhe, (Germany) on October 16-17, 2008. He was a member of the program committee of AINA 2009 and CCGRID 2009. He is a member of the Steering Committee of CompFrame.

Yves Robert is a member of the editorial board of the *International Journal of High Performance Computing Applications* (Sage Press).

Y. Robert was program chair of IPDPS'08 (IEEE Int. Parallel and Distributed Processing Symposium, Miami). He also was vice-chair of the "Scheduling and load balancing workshop" of EuroPar'2008. Finally, he was vice-chair of ICPADS'09 ((IEEE Int. Conf. on Parallel and Distributed Systems, Melbourne), for the track *Parallel Algorithms and Applications*.

[6] http://www.ens-lyon.fr/LIP/RESO/ccgrid2008/

He will chair the IEEE TCPP PhD Forum in Rome (in conjunction with IPDPS'09). He will be program chair of ISPDC'09 that will take place in Lisbon in July 2009.

Y. Robert is a member of the Steering Committee of HCW (IEEE Workshop on Heterogeneity in Computing) and of HiPC (IEEE Int. Conf. on High Performance Computing).

Y. Robert gave an invited talk at:
- Journées IUF de Nancy, Mars 2008;
- Parallel Processing Workshop, Murcia, Spain, May 2008;
- High Performance Computing Conference in Cetraro, Italy, June 2008;
- CCGSC workshop, Asheville, USA, September 2008
Following the 35th (French) Spring school in theoretical computer science (EPIT) they organized in June 2007, Y. Robert and F. Vivien are editing a book on *Introduction to scheduling*, to be pubished in 2009 by Chapman and Hall/CRC Press.

Bernard Tourancheau serves as an expert for the "ministère de l'enseignement supérieur et de la recherche" (DGRI) for the projects PHC and PFCC, and also for the section ST2I of the CNRS for the PEPS projects. He co-organized the CCGSC'08 workshop in Ashville, and organized the thematic day on "IP and sensor networks" of the GDR-CNRS ResCom.

Frédéric Vivien is an associate editor of *Parallel Computing*.

F. Vivien was a member of the program committee of EuroPDP 2009, Weimar, Germany, February 2009, of ICPADS'08, Melbourne, Australia, December 8-10, 2008, of the 3rd CoreGRID Workshop on Grid Middleware, Barcelona, Spain, June 4th - 5th, 2008, and of Grid2008, Tsukuba, Japan, September 29 - October 1, 2008.

In 2008, F. Vivien evaluated a research project for *Wiener Wissenschafts-, Forschungs- und Technologiefonds*, and a *principal investigator program* for the *Science Foundation Ireland*.

Laurent Philippe is member of the program committee of CFSE 2008, the sixth French ACM Conference on Operating Systems, Fribourg, Switzerland, February 2008; DFMA, 2008 (4th International Conference on Distributed Frameworks and Applications), October 2008, Penang, Malaysia.

Jean-Marc Nicod is member of the program committee of DFMA 2008 (4th International Conference on Distributed Frameworks and Applications), October 2008, Penang, Malaysia.

9.3. Administrative and Teaching Responsibilities

9.3.1. Teaching Responsibilities

Master d'Informatique Fondamentale at ENS Lyon. Yves Robert is the head of the computer science teaching department at ENS Lyon. Most permanent members of the project participate in the Master d'Informatique Fondamentale at ENS Lyon and give advanced classes related to parallel algorithms, cluster computing.

Master in Computer Science at Université de Franche Comté. L. Philippe is the head of the Master in Computer Science of Université de Franche-Comté. J.-M. Nicod is in charge of the first year of the master (S2). L. Philippe and J.-M. Nicod participate to this master and give advanced classes related to distributed computing and distributed algorithms.

10. Bibliography

Major publications by the team in recent years

[1] P. R. AMESTOY, I. S. DUFF, J. KOSTER, J.-Y. L'EXCELLENT. *A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling*, in "SIAM Journal on Matrix Analysis and Applications", vol. 23, n^{o} 1, 2001, p. 15-41.

[2] C. BANINO, O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, Y. ROBERT. *Scheduling strategies for master-slave tasking on heterogeneous processor platforms*, in "IEEE Trans. Parallel Distributed Systems", vol. 15, n^{o} 4, 2004, p. 319-330.

[3] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Centralized versus distributed schedulers for multiple bag-of-task applications*, in "IEEE Trans. Parallel Distributed Systems", vol. 19, n^{o} 5, 2008, p. 698-709.

[4] O. BEAUMONT, H. CASANOVA, A. LEGRAND, Y. ROBERT, Y. YANG. *Scheduling divisible loads on star and tree networks: results and open problems*, in "IEEE Trans. Parallel Distributed Systems", vol. 16, n^{o} 3, 2005, p. 207-218.

[5] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Replica placement and access policies in tree networks*, in "IEEE Trans. Parallel Distributed Systems", vol. 19, n^{o} 12, 2008, p. 1614-1627.

[6] E. CARON, F. DESPREZ. *DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid*, in "International Journal of High Performance Computing Applications", vol. 20, n^{o} 3, 2006, p. 335-352.

[7] F. DESPREZ, J. DONGARRA, A. PETITET, C. RANDRIAMARO, Y. ROBERT. *Scheduling block-cyclic array redistribution*, in "IEEE Trans. Parallel Distributed Systems", vol. 9, n^{o} 2, 1998, p. 192-205.

[8] F. DESPREZ, F. SUTER. *Impact of Mixed-Parallelism on Parallel Implementations of Strassen and Winograd Matrix Multiplication Algorithms*, in "Concurrency and Computation: Practice and Experience", vol. 16, n^{o} 8, July 2004, p. 771–797.

[9] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Constructing Memory-minimizing Schedules for Multifrontal Methods*, in "ACM Transactions on Mathematical Software", vol. 32, n^{o} 1, 2006, p. 17–32.

[10] A. LEGRAND, A. SU, F. VIVIEN. *Minimizing the stretch when scheduling flows of divisible requests*, in "Journal of Scheduling", vol. 11, n^{o} 5, 2008, p. 381-404.

Year Publications

Doctoral Dissertations and Habilitation Theses

[11] E. AGULLO. *On the Out-of-core Factorization of Large Sparse Matrices*, Ph. D. Thesis, École Normale Supérieure de Lyon, November 2008.

[12] R. BOLZE. *Analyse et déploiement de solutions algorithmiques et logicielles pour des applications bioinformatiques à grande échelle sur la grille*, Ph. D. Thesis, École Normale Supérieure de Lyon, October 2008.

[13] J.-F. PINEAU. *Communication-aware scheduling on heterogeneous master-worker platforms*, Ph. D. Thesis, École Normale Supérieure de Lyon, September 2008.

[14] C. TEDESCHI. *Peer-to-Peer Prefix Tree for Large Scale Service Discovery*, Ph. D. Thesis, École normale supérieure de Lyon, October 2008.

[15] F. VIVIEN. *On scheduling for distributed heterogeneous platforms*, Habilitation à diriger des recherches, École normale supérieure de Lyon, May 2008.

Articles in International Peer-Reviewed Journal

[16] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *A Parallel Out-of-core Multifrontal Method: Storage of Factors on Disk and Analysis of Models for an Out-of-core Active Memory*, in "Parallel Computing, Special Issue on Parallel Matrix Algorithms", vol. 34, n^{o} 6-8, 2008, p. 296-317.

[17] A. AMAR, R. BOLZE, Y. CANIOU, E. CARON, B. DEPARDON, F. DESPREZ, J.-S. GAY, G. LE MAHEC, D. LOUREIRO. *Tunable Scheduling in a GridRPC Framework*, in "Concurrency and Computation: Practice and Experience", vol. 20, n^{o} 9, 2008, p. 1051–1069.

[18] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Centralized versus distributed schedulers for multiple bag-of-task applications*, in "IEEE Trans. Parallel Distributed Systems", vol. 19, n^{o} 5, 2008, p. 698-709.

[19] A. BENOIT, M. HAKEM, Y. ROBERT. *Contention awareness and fault tolerant scheduling for precedence constrained tasks in heterogeneous systems*, in "Parallel Computing", to appear, 2008.

[20] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Replica placement and access policies in tree networks*, in "IEEE Trans. Parallel Distributed Systems", vol. 19, n^{o} 12, 2008, p. 1614-1627.

[21] A. BENOIT, Y. ROBERT. *Complexity results for throughput and latency optimization of replicated and data-parallel workflows*, in "Algorithmica", to appear, 2008.

[22] A. BENOIT, Y. ROBERT. *Mapping pipeline skeletons onto heterogeneous platforms*, in "J. Parallel and Distributed Computing", vol. 68, n^{o} 6, 2008, p. 790-808.

[23] E. CARON, A. CHIS, F. DESPREZ, A. SU. *Design of plug-in schedulers for a GridRPC environment*, in "Future Generation Computer Systems", vol. 24, n^{o} 1, January 2008, p. 46-57.

[24] M. GALLET, Y. ROBERT, F. VIVIEN. *Comments on "Design and performance evaluation of load distribution strategies for multiple loads on heterogeneous linear daisy chain networks"*, in "J. Parallel and Distributed Computing", vol. 68, n^{o} 7, 2008, p. 1021-1031.

[25] A. LEGRAND, A. SU, F. VIVIEN. *Minimizing the stretch when scheduling flows of divisible requests*, in "Journal of Scheduling", vol. 11, n^{o} 5, 2008, p. 381-404.

[26] J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *The impact of heterogeneity on master-slave scheduling*, in "Parallel Computing", vol. 34, n^{o} 3, 2008, p. 158-176.

[27] J.-F. PINEAU, Y. ROBERT, F. VIVIEN, Z. SHI, J. DONGARRA. *Revisiting Matrix Product on Master-Worker Platforms*, in "International Journal of Foundations of Computer Science", to appear, 2008.

[28] F. SOURBIER, S. OPERTO, J. VIRIEUX, P. R. AMESTOY, J.-Y. L'EXCELLENT. *FWT2D: a massively parallel program for frequency-domain Full-Waveform Tomography of wide-aperture seismic data – Part 1: algorithm*, in "Computer and Geosciences", To appear, 2008.

[29] F. SOURBIER, S. OPERTO, J. VIRIEUX, P. R. AMESTOY, J.-Y. L'EXCELLENT. *FWT2D: a massively parallel program for frequency-domain Full-Waveform Tomography of wide-aperture seismic data – Part 2: numerical examples and scalability analysis*, in "Computer and Geosciences", To appear, 2008.

Articles in National Peer-Reviewed Journal

[30] J.-S. GAY, Y. CANIOU. *Étude de la précision de Simbatch, une API pour la simulation de systèmes batch*, in "RSTI - Techniques et Sciences Informatiques", vol. 27, n^{o} 3-4, 2008, p. 373-394.

International Peer-Reviewed Conference/Proceedings

[31] K. AGRAWAL, A. BENOIT, Y. ROBERT. *Mapping linear workflows with computation/communication overlap*, in "ICPADS'2008, the 14th IEEE International Conference on Parallel and Distributed Systems", IEEE Computer Society Press, 2008, p. 195-202.

[32] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *On the I/O volume in Out-of-Core Multifrontal Methods with a Flexible Allocation Scheme*, in "VECPAR'08 International Meeting on High Performance Computing for Computational Science", LNCS, vol. 5336, Springer-Verlag Berlin Heidelberg, June 2008, p. 328-335.

[33] H. ASTSATRYAN, V. SAHAKYAN, Y. SHOUKOURYAN, M. DAYDÉ, A. HURAULT, M. PANTEL, E. CARON. *A Grid-Aware Web Interface with Advanced Service Trading for Linear Algebra Calculations*, in "8th International Meeting High Performance Computing for Computational Science (VECPAR'08), Toulouse", June 2008, p. 106-113.

[34] A. BENOIT, F. DUFOSSÉ, Y. ROBERT. *On the Complexity of Mapping Pipelined Filtering Services on Heterogeneous Platforms*, in "International Parallel and Distributed Processing Symposium IPDPS'2008", IEEE Computer Society Press, 2009.

[35] A. BENOIT, M. HAKEM, Y. ROBERT. *Fault tolerant scheduling of precedence task graphs on heterogeneous platforms*, in "10th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2008", IEEE Computer Society Press, 2008.

[36] A. BENOIT, L. MARCHAL, J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Offline and online scheduling of concurrent bags-of-tasks on heterogeneous platforms*, in "10th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2008", IEEE Computer Society Press, 2008.

[37] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Bi-criteria pipeline mappings for parallel image processing*, in "ICCS'2008, the 8th International Conference on Computational Science", LNCS, vol. 5102, Springer Verlag, 2008.

[38] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Optimizing latency and reliability of pipeline workflow applications*, in "HCW'2008, the 17th Heterogeneous Computing Workshop", IEEE Computer Society Press, 2008.

[39] A. BENOIT, Y. ROBERT. *Multi-criteria mapping techniques for pipeline workflows on heterogeneous platforms*, in "Recent Developments in Grid Technology and Applications", to appear, Nova Science Publishers, 2008.

[40] A. Benoit, Y. Robert, A. L. Rosenberg, F. Vivien. *Static Strategies for Worksharing with Unrecoverable Interruptions*, in "International Parallel and Distributed Processing Symposium IPDPS'2008", IEEE Computer Society Press, 2009.

[41] V. Bertis, R. Bolze, F. Desprez, K. Reed. *Large Scale Execution of a Bioinformatic Application on a Volunteer Grid*, in "Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC08)", April 2008.

[42] Y. Caniou, E. Caron, G. Charrier, A. Chis, F. Desprez, E. Maisonnave. *Ocean-Atmosphere Modelization over the Grid*, in "The 37th International Conference on Parallel Processing (ICPP 2008), Portland, Oregon. USA", W.-c. Feng, Y. Yang (editors), IEEE, September 2008, p. 206-213.

[43] Y. Caniou, E. Caron, G. Charrier, F. Desprez, E. Maisonnave, V. Pichon. *Ocean-Atmosphere Application Scheduling within DIET*, in "APDCT-08 Symposium. International Symposium on Advanced in Parallel and Distributed Computing Techniques, Sydney. Australia.", Invited paper from the reviewed process of ISPA'08, IEEE Computer Society, In conjunction with ISPA'2008, December 2008, p. 675-680.

[44] Y. Caniou, J.-S. Gay. *Simbatch: an API for simulating and predicting the performance of parallel resources managed by batch systems*, in "Workshop on Secure, Trusted, Manageable and Controllable Grid Services (SGS), held in conjunction with EuroPar'08", To appear, 2008.

[45] Y. Caniou, J.-S. Gay, P. Ramet. *Tunable parallel experiments in a GridRPC framework: application to linear solvers*, in "VECPAR'08 International Meeting on High Performance Computing for Computational Science", LNCS, vol. 5336, 2008, p. 430–436.

[46] Y. Caniou, N. Kushida, N. Teshima. *Implementing interoperability between the AEGIS and DIET GridRPC middleware to build an International Sparse Linear Algebra Expert System*, in "The Second IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2008)", 2008, p. 205–210.

[47] E. Caron, P. K. Chouhan, F. Desprez. *Automatic Middleware Deployment Planning on Heterogeneous Platfoms*, in "The 17th Heterogeneous Computing Workshop (HCW'08), Miami, Florida", In conjunction with IPDPS 2008, April 2008.

[48] E. Caron, A. Datta, F. Petit, C. Tedeschi. *Self-stabilization in tree-structured P2P Service Discovery Systems*, in "27th International Symposium on Reliable Distributed Systems (SRDS 2008), Napoli, Italy", IEEE, October 2008, p. 207-216.

[49] E. Caron, F. Desprez, G. Le Mahec. *Parallelization and Distribution Strategies of Large Bioinformatics Requests over the Grid*, in "International Conference on Algorithms and Architectures for Parallel Processing 2008 (ICA3PP 2008), Cyprus", LNCS, vol. 5022, Springer Verlag, June 2008.

[50] E. Caron, F. Desprez, D. Loureiro. *All-in-one Graphical Tool for the management of DIET a GridRPC Middleware*, in "CoreGRID Workshop on Grid Middleware (in conjunction with OGF'23), Barcelona, Spain", To appear, June 2008.

[51] E. Caron, F. Desprez, C. Tedeschi. *Efficiency of Tree-structured Peer-to-peer Service Discovery Systems*, in "Fifth International Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P), Miami, Florida", In conjunction with IPDPS 2008, April 2008.

[52] P. COMBES, E. CARON, F. DESPREZ, B. CHOPARD, J. ZORY. *Relaxing Synchronization in a Parallel SystemC Kernel*, in "ISPA 2008. International Symposium on Parallel and Distributed Processing with Applications, Sydney. Australia", IEEE Computer Society, December 2008, p. 180-187.

[53] S. DAHAN, A. DOBRILA, J.-M. NICOD, L. PHILIPPE. *Performances Study of the Distributed Spanning Tree an Overlay Network for Server Lookup*, in "ICIW - The Third International Conference on Internet and Web Applications and Services", IEEE Computer Society, 2008, p. 330-335.

[54] F. DESPREZ, E. CARON, G. LE MAHEC. *DAGDA: Data Arrangement for the Grid and Distributed Applications*, in "AHEMA 2008. International Workshop on Advances in High-Performance E-Science Middleware and Applications. In conjunction with eScience 2008, Indianapolis, Indiana, USA", To appear, December 2008.

[55] S. DIAKITÉ, J.-M. NICOD, L. PHILIPPE. *Comparison of Batch Scheduling for Identical Multi-Tasks Jobs on Heterogeneous Platforms*, in "Proceedings of PDP 2008, 16th Euromicro International Conference on Parallel, Distributed and network-based Processing, Toulouse, France", 2008, p. 374–378.

[56] M. GALLET, L. MARCHAL, F. VIVIEN. *Allocating Series of Workflows on Computing Grids*, in "Proceedings of the 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS'08)", To appear, IEEE Computer Society Press, 2008, p. 48-55.

[57] M. GALLET, L. MARCHAL, F. VIVIEN. *Efficient Scheduling of Task Graph Collections on Heterogeneous Resources*, in "International Parallel and Distributed Processing Symposium IPDPS'2008", IEEE Computer Society Press, 2009.

[58] M. GALLET, Y. ROBERT, F. VIVIEN. *Divisible load scheduling*, in "Introduction to Scheduling", to appear, Chapman and Hall/CRC Press, 2008.

[59] D. KAHN, C. REZVOY, F. VIVIEN. *Parallel large scale inference of protein domain families*, in "ICPADS'2008, the 14th IEEE International Conference on Parallel and Distributed Systems", IEEE Computer Society Press, 2008, p. 72-79.

[60] N. KUSHIDA, Y. SUZUKI, N. TESHIMA, N. NAKAJIMA, Y. CANIOU, M. DAYDÉ, P. RAMET. *Toward an International Sparse Linear Algebra Expert System by Interconnecting the ITBL Computational Grid with the GridTLSE Platform*, in "VECPAR'08 International Meeting on High Performance Computing for Computational Science", LNCS, vol. 5336, 2008, p. 424–429.

[61] Y. MAZZER, B. TOURANCHEAU. *MPI in Wireless Sensor Networks*, in "Recent Advances in Parallel Virtual Machine and Message Passing Interface, 15th European PVM/MPI Users' Group Meeting", A. L. LASTOVETSKY, T. KECHADI, J. DONGARRA (editors), Lecture Notes in Computer Science, vol. 5205, 2008, p. 334-339.

[62] J.-F. PINEAU, Y. ROBERT, F. VIVIEN, J. DONGARRA. *Matrix Product on Heterogeneous Master-Worker Platforms*, in "13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Salt Lake City, Utah", February 2008, p. 53–62.

National Peer-Reviewed Conference/Proceedings

[63] S. DAHAN, A. DOBRILA, J.-M. NICOD, L. PHILIPPE. *Étude des performances du Distributed Spanning Tree : un Overlay Network pour la Recherche de Services*, in "Proceedings of CFSE'6, 6th Conférence Française en Systèmes d'Exploitation, Fribourg, Switzerland", 2008.

[64] S. DIAKITÉ, J.-M. NICOD, L. PHILIPPE. *Adaptation d'un algorithme optimal d'ordonnancement en régime permanent pour des lots bornés*, in "Proceedings of RenPar'18, 18èmes Rencontres francophones du Parallélisme, Fribourg, Switzerland", 2008.

[65] C. TEDESCHI. *Arbre de préfixes auto-stable pour les systèmes pair-à-pair*, in "Fribourg'2008 - Conférences conjointes RenPar'18 / SympA'2008 / CFSE'6, Fribourg, Suisse", February 2008.

[66] B. TOURANCHEAU, G. KRAUSS, R. BLANCHARD. *Parametric Sensitivity Study and Optimization of the SDHW and PV Subsystems in an Energy Positive House*, in "IBPSA08, International Building Performance Simulation Association", November 2008.

[67] B. TOURANCHEAU, Y. MAZZER, V. GAVAN, F. KUZNIK, G. KRAUSS. *Calibration study of wirelessly networked temperature sensors*, in "IBPSA08, International Building Performance Simulation Association", November 2008.

Scientific Books (or Scientific Book chapters)

[68] O. BEAUMONT, L. MARCHAL. *Steady-state scheduling*, in "Introduction to Scheduling", To appear, Chapman and Hall/CRC Press, 2009.

[69] Y. CANIOU, E. CARON, F. DESPREZ, H. NAKADA, K. SEYMOUR, Y. TANAKA. *High performance GridRPC middleware*, in "Recent Developments in Grid Technology and Applications", G. GRAVVANIS, J. MORRISON, H. ARABNIA, D. POWER (editors), To appear, Nova Science Publishers, 2008.

[70] E. CARON, F. DESPREZ, F. PETIT, C. TEDESCHI. *DLPT: A P2P tool for Service Discovery in Grid Computing*, in "Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications", N. ANTONOPOULOS, G. EXARCHAKOS, M. LI, A. LIOTTA (editors), To appear, IGI Global, 2009.

[71] A. LEGRAND, H. CASANOVA, Y. ROBERT. *Parallel Algorithms*, Chapman and Hall/CRC Press, 2008.

[72] Y. ROBERT, F. VIVIEN. *Algorithmic Issues in Grid Computing*, in "Algorithms and Theory of Computation Handbook", To appear, Chapman and Hall/CRC Press, 2008.

Books or Proceedings Editing

[73] J. DONGARRA, B. TOURANCHEAU (editors). *Future Generation Computing Systems (FGCS), special issue for the Workshop on Clusters and Computational Grids for Scientific Computing*, vol. 24, n^o 1, Science Direct, January 2008.

[74] Y. ROBERT (editor). *Special issue on IPDPS'2008*, to appear, J. Parallel and Distributed Computing, 2008.

[75] Y. ROBERT, F. VIVIEN (editors). *Introduction to Scheduling*, To appear, Chapman and Hall/CRC Press, 2008.

Other Publications

[76] P. R. AMESTOY, A. BUTTARI, PH. COMBES, A. GUERMOUCHE, J.-Y. L'EXCELLENT, TZ. SLAVOVA, B. UÇAR. *Overview of MUMPS (A multifrontal Massively Parallel Solver)*, Presentation at the 2nd French-Japanese workshop on Petascale Applications, Algorithms and Programming (PAAP'08), June 2008.

[77] P. R. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT. *Towards a parallel analysis phase for a multifrontal sparse solver*, Presentation at the 5th International workshop on Parallel Matrix Algorithms and Applications (PMAA'08), June 2008.

References in notes

[78] R. BUYYA (editor). *High Performance Cluster Computing*, ISBN 0-13-013784-7, vol. 2: Programming and Applications, Prentice Hall, 1999.

[79] P. CHRÉTIENNE, E. G. COFFMAN JR., J. K. LENSTRA, Z. LIU (editors). *Scheduling Theory and its Applications*, John Wiley and Sons, 1995.

[80] I. FOSTER, C. KESSELMAN (editors). *The Grid: Blueprint for a New Computing Infrastructure*, Morgan-Kaufmann, 1998.

[81] A. ORAM (editor). *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology*, O'Reilly, 2001.

[82] P. R. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT. *Towards a parallel analysis phase for a multifrontal sparse solver*, Presentation at the 5th International workshop on Parallel Matrix Algorithms and Applications (PMAA'08), June 2008.

[83] P. R. AMESTOY, T. A. DAVIS, I. S. DUFF. *An approximate minimum degree ordering algorithm*, in "SIAM Journal on Matrix Analysis and Applications", vol. 17, 1996, p. 886–905.

[84] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT. *Multifrontal Parallel Distributed Symmetric and Unsymmetric Solvers*, in "Comput. Methods Appl. Mech. Eng.", vol. 184, 2000, p. 501–520.

[85] P. R. AMESTOY, I. S. DUFF, D. RUIZ, B. UÇAR. *A parallel matrix scaling algorithm*, in "VECPAR 08", Lecture Notes in Computer Science, To appear, Springer, 2008.

[86] D. ARNOLD, S. AGRAWAL, S. BLACKFORD, J. DONGARRA, M. MILLER, K. SAGI, Z. SHI, S. VADHIYAR. *Users' Guide to NetSolve V1.4*, Computer Science Dept. Technical Report, n^{o} CS-01-467, University of Tennessee, Knoxville, TN, July 2001, http://www.cs.utk.edu/netsolve/.

[87] M. BAKER. *Cluster Computing White Paper*, 2000.

[88] E. CARON, A. CHIS, F. DESPREZ, A. SU. *Plug-in Scheduler Design for a Distributed Grid Environment*, in "4th International Workshop on Middleware for Grid Computing - MGC 2006, Melbourne, Australia", In conjunction with ACM/IFIP/USENIX 7th International Middleware Conference 2006, November 27th 2006.

[89] C. CHEVALIER, F. PELLEGRINI. *PT-Scotch: A tool for efficient parallel graph ordering*, in "Proceedings of PMAA2006, Rennes, France", oct 2006.

[90] I. S. Duff, J. K. Reid. *The Multifrontal Solution of Indefinite Sparse Symmetric Linear Systems*, in "ACM Transactions on Mathematical Software", vol. 9, 1983, p. 302-325.

[91] I. S. Duff, J. K. Reid. *The Multifrontal Solution of Unsymmetric Sets of Linear Systems*, in "SIAM Journal on Scientific and Statistical Computing", vol. 5, 1984, p. 633-641.

[92] H. El-Rewini, H. H. Ali, T. G. Lewis. *Task Scheduling in Multiprocessing Systems*, in "Computer", vol. 28, n^{o} 12, 1995, p. 27–37.

[93] G. Fedak, C. Germain, V. Néri, F. Cappello. *XtremWeb : A Generic Global Computing System*, in "CCGRID2001, workshop on Global Computing on Personal Devices", IEEE Press, May 2001.

[94] M. Ferris, M. Mesnier, J. Mori. *NEOS and Condor: Solving Optimization Problems Over the Internet*, in "ACM Transactions on Mathematical Sofware", vol. 26, n^{o} 1, 2000, p. 1-18, http://www-unix.mcs.anl.gov/metaneos/publications/index.html.

[95] C. Germain, G. Fedak, V. Néri, F. Cappello. *Global Computing Systems*, in "Lecture Notes in Computer Science", vol. 2179, 2001, p. 218–227.

[96] G. Karypis, V. Kumar. *ParMetis Parallel Graph Partitioning and Sparse Matrix Ordering Library*, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN 55455, U.S.A., August 2003.

[97] J. W. H. Liu. *The Role of Elimination Trees in Sparse Factorization*, in "SIAM Journal on Matrix Analysis and Applications", vol. 11, 1990, p. 134–172.

[98] S. Matsuoka, H. Nakada, M. Sato, S. Sekiguchi. *Design Issues of Network Enabled Server Systems for the Grid*, Grid Forum, Advanced Programming Models Working Group whitepaper, 2000.

[99] H. Nakada, S. Matsuoka, K. Seymour, J. Dongarra, C. Lee, H. Casanova. *GridRPC: A Remote Procedure Call API for Grid Computing*, in "Grid 2002, Workshop on Grid Computing, Baltimore, MD, USA", Lecture Notes in Computer Science, n^{o} 2536, November 2002, p. 274-278.

[100] H. Nakada, M. Sato, S. Sekiguchi. *Design and Implementations of Ninf: towards a Global Computing Infrastructure*, in "Future Generation Computing Systems, Metacomputing Issue", vol. 15, n^{o} 5-6, 1999, p. 649-658.

[101] M. G. Norman, P. Thanisch. *Models of Machines and Computation for Mapping in Multicomputers*, in "ACM Computing Surveys", vol. 25, n^{o} 3, 1993, p. 103–117.

[102] M. Sato, M. Hirano, Y. Tanaka, S. Sekiguchi. *OmniRPC: A Grid RPC Facility for Cluster and Global Computing in OpenMP*, in "Lecture Notes in Computer Science", vol. 2104, 2001, p. 130–136.

[103] B. A. Shirazi, A. R. Hurson, K. M. Kavi. *Scheduling and Load Balancing in Parallel and Distributed Systems*, IEEE Computer Science Press, 1995.

[104] R. WOLSKI, N. T. SPRING, J. HAYES. *The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing*, in "Future Generation Computing Systems, Metacomputing Issue", vol. 15, n^{o} 5–6, October 1999, p. 757–768.