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

2007

Table of contents

1. Team

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 at the École normale supérieure de Lyon.*

Head of team
Frédéric Vivien [Research Associate (CR) Inria]

Administrative assistants
Isabelle Pera [CNRS, 25% on the project]
Edwige Royboz [INRIA, 25% on the project]

INRIA staff
Frédéric Desprez [Research Director (DR), HdR]
Jean-Yves L'Excellent [Research Associate (CR)]
Frédéric Vivien [Research Associate (CR)]

CNRS staff
Loris Marchal [Research Associate (CR), since October 1, 2007]

Faculty members from ENS Lyon
Anne Benoît [Assistant Professor (MdC)]
Eddy Caron [Assistant Professor (MdC)]
Yves Robert [Professor, HdR]

Faculty members from Université Lyon 1 - UCBL
Yves Caniou [Assistant Professor (MdC)]
Bernard Tourancheau [Professor, joined GRAAL February 6, 2007, HdR]

Faculty members from Université de Franche-Comté (external collaborators)
Jean-Marc Nicod [Assistant Professor, HdR]
Laurent Philippe [Professor, HdR]

Project technical staff
Abdelkader Amar [INRIA, 50% on the project]
Nicolas Bard [CNRS]
Aurélien Ceyden [ENS Lyon, 50% on the project]
Philippe Combes [CNRS, since December 16, 2007]
Aurélia Fèvre [INRIA, until August 31, 2007]
David Loureiro [INRIA]
Vincent Pichon [CNRS, since November 1, 2007]
Emmanuel Quemener [CNRS, until August 31, 2007]

Post-doctoral fellows
Mourad Hakem [INRIA, since October 1, 2007]
Lionel Eyraud-Dubois [ENS Lyon, until August 31, 2007]

Ph. D. students (ENS Lyon)
Emmanuel Agullo [MENRT grant]
Raphaël Bolze [BDI CNRS]
Ghislain Charrier [INRIA Cordi-S grant, since November 1, 2007]
Benjamin Depardon [MENRT Grant, since September 1, 2007]
Matthieu Gallet [ENS Grant]
Jean-Sébastien Gay [Rhône-Alpes region grant]
Jean-François Pineau [ENS grant]
Veronika Rehn-Sonigo [MENRT grant]
Clément Rezvoy [MENRT Grant, since September 1, 2007]
Cédric Tedeschi [MENRT grant]

Ph. D. students (External Members, Univ. Franche Comté)
Sékou Diakité [MENRT grant]
Alexandru Dobrila [MENRT grant, since October 1, 2007]

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 [71]. 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 [84]. 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 the American 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 [78], [69], [71]. 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

- DIET was chosen to be the Grid middleware of the Décrypthon Grid, a joint initiative of AFM (French association against muscular dystrophy), CNRS, and IBM, to help genomics and proteomics research. DIET ensures the load-balancing of jobs over the 6 computing centers federated by this Grid (see section 4.7).

3. Scientific Foundations

3.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Anne Benoît, Lionel Eyraud-Dubois, Matthieu Gallet, 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 [70], [83], [92], [95] 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 $t + \mathrm{comm}(T, T', P_i, P_j)$, where $\mathrm{comm}(T, T', P_i, P_j)$ 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, Aurélia Fèvre, 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 [81], [82], 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 [88] 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], [76]. 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: Abdelkader Amar, Nicolas Bard, Raphaël Bolze, Yves Caniou, Eddy Caron, Aurélien Ceyden, Ghislain Charrier, Frédéric Desprez, Jean-Sébastien Gay, David Loureiro, Vincent Pichon, Frédéric Vivien.

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 [72] or Global computing [86] 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 [89], [90]) 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 [77], Ninf [91], NEOS [85], OmniRPC [94], 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.

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,

[1] http://www.globus.org/
[2] http://www.cs.wisc.edu/condor/condorg/
[3] http://wwws.sun.com/software/gridware/

- 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.

The origin of this work comes from a collaboration with MAPLY, a laboratory of applied mathematics at University Lyon 1 (UCBL).

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 technics. The computational space is decomposed among the available processors using a *mesh partitioning* strategy based on the Peano–Hilbert cell ordering.

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.

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

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 (modelization, 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 Décryphton 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.

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 as 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 it for macroscopic productions. In this context, the cell control will evolve to become more cooperative and distributed.

The micro-factory may be modelled in a way that allows reusing 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.

5. Software

5.1. DIET

Participants: Abdelkader Amar, 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 (http://graal.ens-lyon.fr/DIET) project 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) [96] 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) [80].

Moreover applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via *plug-in schedulers* [80]. 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 of 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 of the packager-person of 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 applications execution. Moreover, existent tools have limited scheduling capabilities, and one of our objectives is to provide an open system which provides several scheduling algorithms, but also that permits to the 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*, but to be flexible we can execute workflow 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 exact (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-workflow based applications. The previous support allows to manage multiple workflow submissions but each submitted workflow was scheduled alone. To study the behaviour 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 concurrency 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 don't really execute tasks (to minimize *MA_DAG load*) but trigger the corresponding clients to execute them, so scheduling decisions can be respected since scheduling and tasks execution start are done in a centralised way.

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

5.1.2. DAGDA: A new data manager for the DIET middleware

"Data Arrangement for Grid and Distributed Applications" is a new data manager for the DIET middleware. Indeed, the previous data manager could not manage data replication and explicit data redistribution among the nodes. We developed this new data manager which allows to control the data placement and which manages several replica for a given data. This data manager is backward compatible with existing DIET applications and some possible extensions such as data encryption, compression etc. should be easy to implement. DAGDA is using the pull model for the data management. That means data is not sent with the request as with the previous data manager, but asked by the server when it needs them. This system gives more flexibility to the data management with, for example, the possibility to download a data from several sources simultaneously. We will use this new data manager to evaluate some data replication scheduling algorithms with DIET.

5.1.3. Batch and parallel job management

Currently, Grids are built on a cluster hierarchy model, as used by the two projects EGEE [5] (*Enabling Grids for E-science in Europe*) and Grid'5000 (see Section 7.2). The production platform for the EGEE project aggregates more than one hundred sites spread over 31 countries. Grid'5000 is the French Grid for the research, which aims to own 5000 nodes spread over France (9 sites are currently participating).

Generally, the use of a parallel computing resource is done via a batch reservation system: users wishing to submit parallel tasks to the resource have to write *scripts* which notably describe the number of required nodes and the walltime of the reservation. Once submitted, the 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 provides during the submission information like the number of demanded resources, the job deadline, etc.

First, we have extended the DIET functionalities. DIET servers are now able to submit tasks to parallel resources, via a batch system or not. For the moment, DIET servers can submit to both OAR and Loadleveler reservation systems, the latter being used in the Décrypthon project. Furthermore, a DIET client can specify if its job must be considered specifically for the corresponding type of resource (sequential task to sequential resource) or if DIET has in charge to choose the best among all available resources. In consequence, the API has been extended with two new calls on the client side, and several new functionalities on the server side: we provide an abstraction layer to batch systems to make reservation information available to the SED. For example, a parallel MPI program must know the identity of the machines on which it is deployed. These are generally reported in a file, which is specific to each batch system. Using a given keyword provided by DIET (here `DIET_BATCH_NODELIST`), the program can access the needed information.

5.1.4. 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 to the user to design graphically a DIET hierarchy. The DIET Mapping tool allows the user 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 SeD. And the DIET Deploy tool is a graphical interface to GoDIET intended for the deployment of DIET hierarchies.

[5] http://public.eu-egee.org/

- Workflow tools including workflow designer and workflow log service. **The Workflow designer** is dedicated to workflow applications written in DIET provide to the user an easy way to design and execute workflows with DIET. The user can compose the different available services and link them by drag'n'drop or load a workflow description file in order to reuse it. Finally these 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 informations about clusters, nodes and jobs. **Resources allocation**: this feature provides an easy way to allocate resources by selecting in a Grid'5000 map the number of required nodes and defining time. The allocated resources can be stored and used with DIET mapping tool. **Resources monitoring** through the use of the Ganglia plugin that gives you 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.

All these tools have been presented at the SuperComputing 2007 conference in Reno, Nevada on the DIET slot of the INRIA booth.

5.1.5. GridRPC data management API

The GridRPC paradigm is now an OGF standard. The GridRPC community has interests in the Data Management within the GridRPC paradigm. Because of previous works performed in the DIET middleware concerning Data Management, Eddy Caron has been promoted co-chair of the GridRPC working group in order to lead the project to propose a powerful Grid Data Management API which can extend the GridRPC paradigm.

Data Management is a challenging issue inside the GridRPC 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 during the OGF'21. We are currently improving it, remarks having already been taken into account. The new proposal will be presented during the OGF'22.

5.2. MUMPS

Participants: Emmanuel Agullo, Aurélia Fèvre, 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,
- preprocessing and scaling for symmetric and unsymmetric matrices
- partial factorization and Schur complement matrix,
- 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 4.7.3, available since June 2007 (see http://graal.ens-lyon.fr/MUMPS/avail.html). The most recent features available are: Matlab and Scilab interfaces, better numerical processing of symmetric indefinite matrices, detection of null pivots, and estimate of a null space basis. Furthermore, an out-of-core functionality has been made available to a number of users, and we are currently taking into account their feedback to design a new version.

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, Lionel Eyraud-Dubois, Matthieu Gallet, Loris Marchal, Jean-Marc Nicod, Laurent Philippe, Jean-François Pineau, Veronika Rehn-Sonigo, Clément Rezvoy, Yves Robert, Bernard Tourancheau, Frédéric Vivien.

6.1.1. Steady-State Scheduling

The traditional objective, when scheduling sets of computational tasks, is to minimize the overall execution time (the *makespan*). However, in the context of heterogeneous distributed platforms, makespan minimization problems are in most cases NP-complete, sometimes even APX-complete. But, when dealing with large problems, an absolute minimization of the total execution time is not really required. Indeed, deriving *asymptotically optimal* schedules is more than enough to ensure an efficient use of the architectural resources. In a nutshell, the idea is to reach asymptotic optimality by relaxing the problem to circumvent the inherent complexity of minimum makespan scheduling. The typical approach can be decomposed in three steps:

1. Neglect the initialization and clean-up phases, in order to concentrate on steady-state operation.
2. Derive an optimal steady-state scheduling, for example using linear programming tools.
3. Prove the asymptotic optimality of the resulting schedule.

This year we have studied a complex application where users, or clients, submit several bag-of-tasks applications on a heterogeneous master-worker platform, using a classical client-server model. The applications are submitted on-line, which means that there is no a priori (static) knowledge of the workload at the beginning of the execution. When several applications are executed simultaneously, they compete for hardware (network and CPU) resources. The traditional measure to quantify the benefits of concurrent scheduling on shared resources is the maximum stretch. The stretch of an application is defined as the ratio of its response time under the concurrent scheduling policy over its response time in dedicated mode, i.e. if it were the only application executed on the platform. The objective is then to minimize the maximum stretch of any application, thereby enforcing a fair trade-off between all applications. Because we target an on-line framework, the scheduling policy will need to be modified upon the arrival of a new application, or upon the completion of another one. Our scheduling strategy relies on complicated mathematical tools but can be computed in time polynomial to the problem size. Also, it can be shown optimal for the off-line version of the problem, with release dates for the applications. On the practical side, we have run extensive simulations and several MPI experiments to assess the quality of our solutions.

6.1.2. Algorithmic kernels on master-slave platforms with limited memory

This work is aimed at designing efficient parallel matrix-product algorithms for heterogeneous master-worker platforms. While matrix-product is well-understood for *homogeneous 2D-arrays of processors* (e.g., Cannon algorithm and ScaLAPACK outer product algorithm), there are three key hypotheses that render our work original and innovative:

Centralized data. We assume that all matrix files originate from, and must be returned to, the master. The master distributes both data and computations to the workers (while in ScaLAPACK, input and output matrices are initially distributed among participating resources). Typically, our approach is useful in the context of speeding up MATLAB or SCILAB clients running on a server (which acts as the master and initial repository of files).

Heterogeneous star-shaped platforms. We target fully heterogeneous platforms, where computational resources have different computing powers. Also, the workers are connected to the master by links of different capacities. This framework is realistic when deploying the application from the server, which is responsible for enrolling authorized resources.

Limited memory. Because we investigate the parallelization of large problems, we cannot assume that full matrix panels can be stored in the worker memories and re-used for subsequent updates (as in ScaLAPACK). The amount of memory available in each worker is expressed as a given number m_i of buffers, where a buffer can store a square block of matrix elements. The size q of these square blocks is chosen so as to harness the power of Level 3 BLAS routines: $q = 80$ or 100 on most platforms.

We have devised efficient algorithms for resource selection (deciding which workers to enroll) and communication ordering (both for input and result messages). We report a set of numerical experiments on various platforms at École Normale Supérieure de Lyon and the University of Tennessee. These platforms are either homogeneous or heterogeneous. In the latter case, the impact of our new algorithms on the overall performance is even greater.

6.1.3. Replica placement

This study consists in introducing and comparing several policies to place replicas in tree networks, subject to server capacity and QoS constraints. In this framework, the flows of client requests are known beforehand, while the number and location of the servers are to be determined. The standard approach in the literature is to enforce that all requests of a client be served by the closest server in the tree.

Our work on replica placement has been finalized this year by the preparation of a survey paper that assesses the usefulness of our two new policies (*Upwards* and *Multiple*) to place replicas in tree networks, subject to server capacity and QoS constraints. The survey encompasses many new theoretical results, and provides a comprehensive set of experiments. In particular, the experiments analyze the impact of server heterogeneity, together with the difficulty to find a good trade-off between favoring clients with a large number of requests and clients with a very constrained QoS. The survey paper will appear in *IEEE Trans. Parallel and Distributed Systems*.

6.1.4. 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.

This year, we have discussed 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 result is the NP-completeness of the throughput maximization problem for interval-based mappings on Communication-Homogeneous platforms, which is the extension of the well-known chains-to-chains problem in a heterogeneous setting.

We have established several new theoretical complexity results for a simplified model with no communication cost, but considering bi-criteria optimization problems (throughput, latency) and both pipeline and fork graphs. We considered that pipeline or fork stages can be replicated in order to increase the throughput of the workflow, by sending consecutive data sets onto different processors. In some cases, stages can also be data-parallelized, i.e. the computation of one single data set is shared between several processors. This leads to a decrease of the latency and an increase of the throughput. Some instances of this simple model are shown to be NP-hard, thereby exposing the inherent complexity of the mapping problem. We provided polynomial algorithms for other problem instances. Altogether, we provided solid theoretical foundations for the study of mono-criterion or bi-criteria mapping optimization problems.

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.

6.1.5. VoroNet

In collaboration with ASAP (IRISA) and CEPAGE (LaBRI), we have proposed the design of VoroNet, an object-based peer to peer overlay network relying on Voronoi tessellations, along with its theoretical analysis and experimental evaluation. VoroNet differs from previous overlay networks in that peers are application objects themselves and get identifiers reflecting the semantics of the application instead of relying on hashing functions. Thus it provides a scalable support for efficient search in large collections of data. In VoroNet, objects are organized in an attribute space according to a Voronoi diagram. VoroNet is inspired from the Kleinberg's small-world model where each peer gets connected to close neighbors and maintains an additional pointer to a long-range node. VoroNet improves upon the original proposal as it deals with general object topologies and therefore copes with skewed data distributions. We show that VoroNet can be built and maintained in a fully decentralized way. The theoretical analysis of the system proves that the routing in VoroNet can be achieved in a poly-logarithmic number of hops in the size of the system. The analysis is fully confirmed by our experimental evaluation by simulation.

6.1.6. 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 demonstrated the feasibility of this parallelization at the scale of a cluster or a Grid, yielding a 50+ acceleration factor over the sequential algorithm. 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.

6.1.7. Automatic discovery of platform topologies

Most of the advanced scheduling techniques require a good knowledge of the interconnection network. This knowledge, however, is rarely available. We are thus interested in automatically building models, from an application point of view, of the interconnection networks of distributed computational platforms.

In the scope of this work we have contributed to the software ALNeM which is a framework to perform network measures and modelling, and which can also be used to perform simulations. In the same framework, we can therefore build a model and assess its quality.

Initially, we have shown that the commonly used model building algorithms (building cliques or spanning trees) all have serious weaknesses which forbid them to accurately predict the running times of simple algorithmic kernels. We have then proposed three new algorithms and we have assess their quality. We have shown that one of these algorithms is able to produce an accurate model of an interconnection network. This algorithm requires network measures to be performed on each of the active elements in the interconnection network (computing nodes, routers, etc.). In the future, we will try to overcome this limitation.

6.1.8. Scheduling multiple divisible loads on a linear processor network

Min, Veeravalli, and Barlas have recently proposed strategies to minimize the overall execution time of one or several divisible loads on a heterogeneous linear processor network, using one or more installments [98], [97]. We have shown on a very simple example that their approach does not always produce a solution and that, when it does, the solution is often suboptimal. We have also shown how to find an optimal schedule for any instance, once the number of installments per load is given. Then, we formally stated that any optimal schedule has an infinite number of installments under a linear cost model as the one assumed in [98], [97]. Therefore, such a cost model cannot be used to design practical multi-installment strategies. Finally, through extensive simulations we confirmed that the best solution is always produced by our linear programming approach.

6.1.9. Scheduling small to medium batches of identical jobs

When considering the scheduling of small to medium batches of identical jobs, we have the choice between steady-state, makespan and batch oriented techniques. Steady-state techniques allow to achieve an optimal use of the resources for job series of infinite size. However, the cost of the initialization and termination phase is not controlled and, if the size of the considered series is too small, the overhead generated during the initial phase may lead to an inefficient scheduling. Makespan or on-line oriented techniques are usually designed to optimize the execution of graphs of tasks on a platform. As the problem of scheduling a set of jobs on an heterogeneous platform is NP-complete, these techniques relies on heuristics to compute a sub-optimal scheduling. Makespan oriented techniques computes this scheduling off-line, before the execution, with the assumption that no other jobs will be run on the platform during the execution of the task set. Each task to be scheduled is managed independently of the other tasks. As these techniques try to optimize the execution of the whole set of tasks, they do not suffer from the initialization problem. However, if the number of tasks scales up the time needed to compute a optimized scheduling usually becomes too long due to the complexity of the algorithm. On-line oriented techniques computes dynamically the scheduling taking into account already running tasks to place the tasks of the arriving jobs. These two last techniques do not benefit from the knowledge that the executed jobs are identical.

Using the SimGrid toolkit, we have developed a simulator to compare the performances of these three approaches. We have exhibited their domain of interest depending on the batch size, the application and platform characteristics. Generally, Makespan or on-line oriented schedules get better performances for small sized batches and, in this case, the steady state schedules are penalized by their initialization and termination phases. The steady state scheduling is however not time consuming, so it is worth adapting it to this context. As the size of the suboptimal phases directly depends on the optimal period size, we compute an optimal period size to extend the use of steady state scheduling to small to medium sized batches.

6.1.10. Scheduling for real-time brain-machine interfaces

In collaboration with the ACIS Laboratory, University of Florida, we have studied how to schedule a particular application with real-time constraint, on a distributed environment. The target application is a brain-machine interface: it receives signals coming from the premotor cortex of an animal's brain, treats these signals and produces a motor command which operates a robotic arm. The signal processing consists in the collaboration of a big number of "expert models": each expert model computes an output (the motor command). All outputs are gathered using a responsibility estimator to predict the significance of the models at a given time. The big number of models to be computed supports the use of a distributed architecture. Each model is implemented as a linear filter of the neural signals with its own set of parameters. Periodically, a set of models have to be trained, and their parameter updated, so that they can improve their accuracy. Before considering the scheduling problems linked to this application, we concentrate on optimizing the computation of the models, and especially the training phase. About 10 seconds of data are needed to train a model and computes its new parameters. Performing the whole computation of the training leads to huge running time which are not acceptable in the context of the real-time application. We proposed that this computation is performed "online", without waiting for the last data to be received before starting the training. We adapt existing adaptive filters like recursive least square filters from signal processing literature to take into account the particularities of the application, such that multi-dimensional inputs and outputs. This online computation offers satisfactory reactivity and numerical stability.

6.1.11. Numerical Simulation for Energy Efficiency

Numerical simulation can have a great impact in the design of buildings in order to predict their energy consumption. We worked on the Ener+ framework provided by CETHIL (Centre de Thermique de Lyon, UMR 5008) in order to optimize the design of an house using parametric optimization by multiple executions of the TRNSYS simulation engine [87]. The results were very promising with a final design of a house which yearly produce twice its needs, including its own energy and its four inhabitants specific energy consumption.

6.1.12. Routing in Low Power and Lossy Networks

The size reduction of computing and networking components opened a new field of applications, the embedded sensors and actuators on motes networks. Ad-hoc networking provides a foundation basis for these new devices systems but their low power and lossy characteristics are adding new challenges as well as their potential application on a very large scale in our environment for monitoring and control purposes. We explored these sensor networked platforms and setup an experimental testbed at the lab. We developed a parameterized and programmable interface on top of the existing middleware provided by the vendor. We were following closely the IETF charters related to sensor networks and especially last year the 6lowPAN, RL2N and RSN discussions. In this context we are now working to implement an IPv6 stack for motes running TinyOS. Our aim is to take advantage of our large experience in networking optimization and communication scheduling in highly connected graphs to both provide very efficient mesh-routing for motes networks and long distance connectivity for motes clouds. On the application side, we are working on the calibration of our platform in order to better estimate the quality of measurement obtained from low cost embedded sensors.

6.2. Providing access to HPC servers on the Grid

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

Participants: Abdelkader Amar, Raphaël Bolze, Yves Caniou, Eddy Caron, Ghislain Charrier, Benjamin Depardon, Frédéric Desprez, Jean-Sébastien Gay, David Loureiro, Jean-Marc Nicod, Laurent Philippe, Vincent Pichon, Emmanuel Quemener, Cédric Tedeschi, Frédéric Vivien.

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 scientifc 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, and precedence contraints usually follow from data flow between them. 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 in the context that 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 performances: for example, a powerful resource may stay idle in order to be able to submit a little later a job that only it can provide. Numerous heuristics have been designed, and we currently evaluate them before implementing them in the DIET Grid middleware.

6.2.2. Service Discovery in Peer-to-Peer environments

The area studied is computational Grids, peer-to-peer systems and their possible interactions to provide a large scale service discovery within computational Grids. In order to address this issue, we first developed a new architecture called *DLPT (for Distributed Lexicographic Placement Table)*. This is a distributed indexing system based on a prefix tree that offers a totally distributed way of indexing and retrieving services at a large scale. The DLPT offers flexible search operations for users while offering techniques to replicate the structure for fault tolerance over dynamic platforms and a greedy algorithm partially taking into account the performance of the underlying network.

One of the fundamental aspects of the peer-to-peer systems is the fault tolerance. Starting from the fact that replication is costly and does not ensure the system to recover after transient failures and during a collaboration with Franck Petit from the LaRIA, we developed some new fault tolerance algorithms for our architecture. First, we provided a protocol to repair it after node crashes. Second, we proposed a self-stabilizing version of such architectures [45]. We have begun a collaboration with Prof Ajoy K. Datta from the University of Nevada, Las Vegas about a self-stabilizing version of the structures used in DLPT, but designed in a less restricted model than [45]. A paper resulting from this work is currently under submission.

We recently developed some algorithms allowing to efficiently map the logical structures used within our architecture onto networks structured as rings. We also developed new load balancing heuristics for DHTs, and adapted them and others to our case. We obtained some good results when comparing our heuristic with others. This work is also currently under submission.

Finally, a prototype of this architecture has been developed as a part of a collaboration about *Networked Service Discovery* involving Pascale Primet and Pierre Bozonnet from the RESO team and Alcatel. This prototype is currently under development and is studied for its experimentation over a real platform such as Grid'5000.

6.2.3. Deployment for DIET: Software and Research

Using distributed heterogeneous resources requires efficient and simple tools to deploy applications. However, current tools still lack maturity concerning their resource selection methods. This year we have proposed an extension to the generic application description model (*GADe*) of a Grid deployment software: ADAGE ("Automatic Deployment of Applications in a Grid Environment") developed in the project team PARIS-IRISA at Rennes.

Our extension for *GADe* proposes a model for hierarchical applications (tree shaped applications). We present a heuristic to find the shape of a hierarchical application on a given platform, and also two kinds of heuristics: one based on two sub-heuristics (one to define a set of nodes, and one to choose among the nodes), and the second based on affinity lists between the nodes and the processes. We try to satisfy three criteria: minimize the communication costs, balance the load among the nodes, and maximize the number of deployed instances. Our simulations show that there is not *a* better heuristic, even if the most interesting one is *affinity*. One has to choose a heuristic depending on what part of the objective function he wants to prioritize (which combination of *number of deployed instances*, *communication cost*, and *load balancing*). We also deployed hundreds of DIET elements using ADAGE, and showed that this tool is far more efficient than the current DIET deployment tool: GoDIET.

The subject is far from being closed. The future works will be to propose deployment heuristics for parallel applications, which represent a large part of the applications used on a Grid. An important point which has not been taken into account is the compatibility between the applications and the resources: an application may not be launched on the whole platform due to memory, or disk space, or even libraries constraints. This reduces the possibilities to map the instances.

Even if we do not know when the communications take place, they can generate bottlenecks on the communication links. Indeed, if we do not consider the platform as a fully connected graph, we should take into account the path that the connections have to follow, and if they take place simultaneously it is possible that the communication link do not support the load.

Finally, our work considers only static deployments. We consider that we have a set of resources, and a set of processes to deploy at a given time t. This deployment does not change afterwards. The utilization of the processes deployed at a time t will certainly be different from its utilization later on. This raises the redeployment problem, that is to say the modification of the current deployment to take into account the new parameters. This requires to take into account the current mapping of the processes, as well as the modifications on the processes parameters.

To validate our heuristics on deployment in a real environment, we implemented them in the deployment software ADAGE. Our experiments on Grid'5000 allowed us to deploy hundreds of DIET elements in a much more efficient fashion than GoDIET (two times faster). We intend to interface DIET Dashboard with ADAGE and to replace GoDIET in the deployment process of DIET.

6.2.4. Grid'5000 large scale experiments

- Large Experiment for Cosmological Simulations. We studied the possibility of computing a lot of low-resolution simulations. The client requests a 128^3 particles $100Mpc.h^{-1}$ simulation (first part). When it receives the results, it requests simultaneously 100 sub-simulations (second part). As each server cannot work on more than one simulation at the same time, we won't be able to have more than 11 parallel computations at the same time. The experiment (including both the first and the second part of the simulation) lasted 16h 18min 43s (1h 15min 11s for the first part and an average of 1h 24min 1s for the second part). After the first part of the simulation, each SeD received 9 requests (one of them received 10 requests) to compute the second part. The total execution time for each server is not the same: about 15h for Toulouse and 10h30 for Nancy. Consequently, the schedule is not optimal. The equal distribution of the requests does not take into account the machines processing power. In fact, at the time when DIET receives the requests (all at the same time) the second part of the simulation has never been executed, hence DIET doesn't know anything on its processing time, the best it can do is to share the total amount of requests on the available SEDs. A better makespan could be attained by writing a plug-in scheduler. We work on this problem.
- DIET at Supercomputing. At the INRIA booth from the conference SuperComputing'07, we have shown from Reno a real execution on Grid'5000 through GRUDU, using the DIET Dashboard. The challenge was that we performed this demo with all steps of Grid usage, from operating system deployment to workflow execution. To evaluate the performance of DIET on the French Grid Grid'5000 and present its functionnalities in a demo, the DIET DashBoard and its fork GRUDU are very useful. GRUDU (Grid'5000 Reservation Utility for Deployment Usage) is a tool for managing Grid'5000 resources, reservations and deployments. Initially developed to help DIET users on Grid'5000, this tool from DIET Dashboard can be used in a stand-alone version called GRUDU. A first part of the demo of GRUDU highlights how it can be interesting for the Grid end-users. The second part of the demo focuses on how to use DIET and the workflow part on a real Grid through the DIET Dashboard.

6.2.5. 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. But our results hold if the submitted requests follow precisely the usage frequencies given as an input for the static replication and scheduling algorithm. Due to particular biological experiments these schemes 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. Again 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 requests scheduling to get an efficient usage of all computation resources.

In this previous work, we designed a scheduling strategy based on the hypothesis that, as 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 that 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 necessary information to 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.

6.2.6. Parallel Job Submission Management

We have performed several experiments, some with *Ramses* (see Section 4.5) and others using the Décrypthon applications. We plan to build a client/server for the LAMMPS software (see Section 4.2). We have undertaken some work to add performance prediction for parallel resources to DIET: communicate with batch system and simulating them with the Simbatch simulator that we have developed (see next section). Hence, we will have sufficient information to incorporate pertinent distributed scheduling algorithms into DIET.

6.2.7. Job Submission Simulations

Generally, the use of a parallel computing resource is done via a batch reservation system. The algorithms involved can greatly impact performance and consequently, be critical for the efficiency of Grid computing. Unfortunately, few Grid simulators take those batch reservation systems into account. They provide at best a very restricted modeling using an FCFS algorithm and few of them deal with parallel tasks. In this context we have proposed a reusable module, named Simbatch [6] as a built-in for the Grid simulator Simgrid [7] allowing to easily model various batch schedulers.

Simbatch is an API written in C providing the core functionalities to easily model batch schedulers, and design and evaluate algorithms. For the moment, three of the most famous algorithms for batch schedulers are already incorporated: *Round Robin* (RR), *First Come First Served* (FCFS) and *Conservative BackFilling* (CBF). A simple use of batch schedulers provided by Simbatch in a Simgrid simulation is done via the two traditional configuration files of SimGrid (platform file and deployment file) and another file named `simbatch.xml` describing every batch used in it. For an advanced use of Simbatch, a set of functions is available to make new plug-in algorithms.

We have compared the flow metrics (time of a task spent in the system) for each task between a real batch system (OAR, developed in Grenoble, which instantiates CBF) and the Simbatch simulator. Simulations without communication costs show an error rate on the flow metrics generally below 1% while simulations involving communication costs show an error rate around 3%. Schedules are in the majority of our experiments and in both cases strictly the same. Those good results allow us to consider the use of Simbatch as a prediction tool that can be integrated in Grid middleware such as DIET.

6.2.8. Scheduling for Ocean-Atmosphere simulations

We have analyzed and modelled a real climatology application with the purpose of deriving appropriate scheduling heuristics. First, the application has been modelled as independent identical workflows derived through the chaining of several basic DAGs. Then a simplified model with clustered tasks based upon the actual time parameters of the application has been derived. For this new model, a first scheduling heuristic (driven by the principle of allocating the same number of processors to all multi-processor tasks and leaving what is left to post-processing tasks) has been issued. Three improved versions have been proposed: a first one that distributed resources left unused evenly across the groups of processors, a second one which doesn't leave any resource for the post processing tasks and distributes all left resources evenly to the groups of processors

[6] http://simgrid.gforge.inria.fr/doc/contrib.html
[7] http://simgrid.gforge.inria.fr/

and a third one that models the problem of dividing the resources of the platform in disjoint sets as an instance of the Knapsack problem with a supplementary constraint. The three improved versions have been simulated and yielded gains of upto 9 %.

Finally, scheduling heuristics for the generalized problem of scheduling independent identical chains of identical DAGs (composed of an independent pre-processing task, an independent post-processing task, a main processing task and an inter-processing task linking succesive DAGs, all tasks being multi processor) have been proposed and compared to the approach of applying a mixed-parallelism scheduling algorithm to the composite DAG resulting when linking all entry tasks to a common entry node and all exit tasks to a common exit node. The results of the 4 heuristics proposed were highly encouraging not only in terms of gains obtained with respect to the results of the CPA mixed parallelism scheduling algorithms, but also in terms of running times for finding the solution (at most a second for determining the optimal pipeline compared to tens of minutes or even an hour for running CPA on a problem of the dimension 10 chains of 1800 iterations of the basic DAG each).

6.3. Parallel Sparse Direct Solvers

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

Participants: Emmanuel Agullo, Aurélia Fèvre, Jean-Yves L'Excellent.

6.3.1. Extension, support and maintenance of the software package MUMPS

This year, we have pursued some work to add functionalities and improve the MUMPS software package. The out-of-core functionality (in which computed factors are written to disk) has been made available to a number of users (Samtech S.A., Free Field Technologies, EADS, ...) and we are taking into account their feedback to design and develop a new out-of-core version. As usual, we have had strong interactions with many users, and this has led us to work on the following points:

- Reduction of the memory requirements of the solution step. By modifying our algorithms and storing parts of solutions in a datastructure that follows the tree structure, some workarrays of fixed size could be reduced and now scale with the number of processors. This was critical for some applications (e.g., seismic imaging at Geosciences Azur), where the memory usage for the solution step could lead to system paging when using many right-hand sides.
- Build a reduced right-hand side and use a partial solution. This new functionality can be used in conjunction with the Schur complement/partial factorization functionality. The need for it arised during the MUMPS Users' day (24 October 2006, ENS Lyon). It is particularly useful in domain decomposition methods or in coupled problems, where we can distinguish interior and interface variables. The idea is that when MUMPS works on the interior variables, it returns to the user a reduced problem (Schur complement) but now also a reduced right-hand side corresponding to the interface variables. The calling application uses this information to build the solution on the interface problem, which can be reinjected to MUMPS in order to build the solution on the interior problem. Note that this implies that forward and backward substitution steps may now be called separately.
- Detection of null pivots. A first version was developed that allows the detection of null (tiny) pivots and the treatment of singular problems.
- Various more minor improvements and bug corrections.
- While the above developments are part of MUMPS 4.7.3, the latest release, we have also been working on research work (see the following subsections) that will affect future releases of MUMPS. For instance, as part of the ANR Solstice project, we are currently working on coupling parallel scalings algorithms developed by Bora Ucar (CERFACS) with MUMPS.

Furthermore, in the context of an INRIA ODL ("Opération de Développement Logiciel"), Aurélia Fèvre worked on several software issues: portability to various platforms including Grid'5000, tools for checking the performance from one release to the next, extension of the non-regression tests running each night, as well as code coverage with associated tests, in order to validate existing code and identify dead parts.

Notice that MUMPS users constantly provide us with new challenging problems to solve and help us validate and improve our algorithms. We have informal collaborations around MUMPS with a number of institutions: (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 publically available collections of sparse matrices (Davis collection, Rutherford-Boeing and PARASOL).

6.3.2. Memory usage and asynchronous communications

The memory usage for the management of asynchronous communications can represent a large portion of the total memory of the parallel multifrontal approach. This is especially true if we envisage out-of-core executions, where part of the other required storage is on disk, but even for in-core executions, that memory is generally far from being negligible. In the algorithm used, data are copied to a buffer before being sent; this is necessary because sent data are not contiguous in memory and we need to reuse that memory in order to perform other tasks during the asynchronous communications. The granularity for the send operations depends on the tasks graph, and one message (containing a block of matrix) is sent from one task to the one that depends on it. The work consisted in dividing such messages into several smaller ones, at the cost of possibly stronger synchronizations: when communication buffers are full (and one process cannot send all the data it needs to send), it sends as much as it can. Because the receiver may be in a similar situation (trying to send but without enough space in its send buffer), we need a mechanism to avoid deadlock. This mechanism is the following: when a process is blocked not being able to send as much as it can, it tries to perform receptions and process messages (slave tasks, assemblies, ...), thus freeing some memory in the buffers of other processes.

This work should have a strong impact on the overall memory usage of parallel sparse direct methods like MUMPS as it will allow to process significantly larger problems both in-core and out-of-core. We still have to study more precisely its impact on performance before making it available in a future release.

6.3.3. Computing the nullspace of large sparse matrices

In the context of the LU or LDL^t factorization of large sparse matrices, we have developed a new approach allowing to (i) detect (pseudo-) zero pivots; (ii) delay the factorization of suspect (small) pivots, on which a specific numerical treatment (rank-revealing QR for example) is performed. Furthermore, once this has been done, an approximation of the null space basis is computed thanks to backward substitutions. A first prototype including these features has been developed within MUMPS. A numerical study of the behaviour and the limits of this approach is currently being done in collaboration with CERFACS on test problems provided by EDF, in order to validate the approach, tune it and understand its limits. This type of algorithms is particularly useful in electromagnetism (where the dimension of the nullspace can be large), and in FETI-like domain decomposition methods (where the dimension of the nullspace is typically smaller than 6).

6.3.4. Serial out-of-core factorization

Because of the large memory requirements of sparse direct solvers, the use of out-of-core approaches is mandatory for large-scale problems, where disks are used to extend the core memory. In this context, left-looking and multifrontal methods are the two most widely used approaches. Even though several algorithms have been proposed for both methods, it is not yet obvious which one best fits an out-of-core context. Noticing that there was still room before reaching the intrinsic limits of each method, the natural first step has been to study each method separately in order to improve their respective efficiency.

Concerning the multifrontal method, we have modelled the problem of the minimization of the I/O volume [34] in the classical (*terminal allocation scheme*) case. We have proposed several algorithms and possible associated memory managements for 3 different assembly schemes, including a new assembly scheme that we specifically designed for an out-of-core context. We have then extended this minimization problem to the more general *flexible parent allocation* algorithm [9]: we have explained [75] how to compute the I/O volume with respect to this scheme and proposed an efficient greedy heuristic which further reduces the I/O volume on real-life problems in this new context. These new algorithms should allow to improve the new generation of serial out-of-core multifrontal codes based on the flexible allocation scheme (such as [93]), as well as the serial parts of parallel codes [59].

Thanks to a collaboration with Xiaoye S. Li and Esmond G. Ng (Lawrence Berkeley National Laboratory, Berkeley, California, USA), the I/O volume minimization problem for unsymmetric supernodal methods has also been studied. This was done in the context of a six-month visit of Emmanuel Agullo to Berkeley from June to November 2007, which followed a first visit of fifteen days in September 2006. Contrary to multifrontal methods for which locality is quite natural, one difficulty of supernodal approaches consists in designing a correct scheme respecting some locality constraints, before intending to minimize the I/O volume under these constraints. We have selected two such different schemes from the state-of-the-art: (i) a left-looking method, for which we have designed a prototype simulating the out-of-core behaviour (the actual I/Os are not performed) by extending the SUPERLU in-core code; and (ii) a left-looking/right-looking hybrid method for which we have computed the subsequent I/O volume by instrumenting the previous prototype. Note that the implementation of the latter method would require to write the whole kernel of computations and is out of the scope of our short-term perspectives. For both approaches, the I/O volume depends on the partition of the supernodal elimination tree. We have proposed an optimal I/O minimization partitioning algorithm for the left-looking approach. We have shown that this problem is NP-complete in the hybrid method case for which we have thus proposed and implemented a new heuristic. This is a work in progress: we are currently doing some experiments to assess the potential of these algorithms.

Now each method has been accurately modelled and improved to further fit out-of-core requirements, the next step will consist in comparing multifrontal and supernodal approaches both through simulations of the I/O volume and experiments on large real-life problems.

6.3.5. Parallel out-of-core factorization

As well as reducing the factorization time, parallelism allows one to further decrease the memory requirements of direct solvers. We have designed (by improving a previous prototype [74]) a robust parallel out-of-core multifrontal solver to solve very large sparse linear systems (several million equations and several hundred million nonzeros). This solver processes the factors out-of-core (but not temporary data). We have shown [59] how the low-level I/O mechanisms impact the performance and have designed a low-level I/O layer that avoids the perturbations introduced by system buffers and allows consistently good performance results. This out-of-core solver is publicly available and is already used by several academic and industrial groups. To go significantly further in the memory reduction, it will be interesting to also store the intermediate working memory on disk. However, before that, two critical issues are being addressed: the first one concerns the memory consumption of *communication buffers* (see the corresponding paragraph above); the second one concerns the memory for *I/O buffers* used in the case of asynchronous direct I/Os. To reduce that memory usage, we have worked on reducing the granularity of I/Os, now done panel by panel, in order to allow for almost arbritrarily small buffers. This work is performed in the context of the PhD of Emmanuel Agullo, in collaboration with Abdou Guermouche (LaBRI and INRIA project ScAlAppliX) and Patrick Amestoy (ENSEEIHT-IRIT).

Once the factors are on disk, they have to be read back for the solution step. In order to improve that step, we collaborate with Tzvetomila Slavova (Ph.D. CERFACS) who focuses on this phase of the computation. For instance we are currently designing an algorithm which allows to schedule the I/O in a way that separates the L and U factors on disk during the factorization step in the unsymmetric case: this allows to perform twice less read operations at the solution step.

6.3.6. Hybrid Direct-Iterative Methods

We collaborate with Haidar Azzam (Ph.D., CERFACS) and Luc Giraud (ENSEEIHT-IRIT) on hybrid direct-iterative solvers. The substructuring methods developed in this context rely on the possibility to compute a partial factorization, with a so-called Schur complement matrix, that is typically computed by a direct solver such as MUMPS. The direct solver is called on each subdomain of a physical mesh, and the iterative approach takes care of the interface problem, based on Schur complements provided by our direct solver. We have been working on tuning this functionality and giving advice on how to best exploit the direct solver in the context of such iterative approaches.

6.3.7. Expertise site for sparse direct solvers (GRID TLSE project)

The GRID TLSE project (see [73]), coordinated by ENSEEIHT-IRIT, is an expertise site providing a one-stop shop for users of sparse linear algebra software. This project was initially funded by the ACI Grid (2002-2005). A user can access matrices, databases, information and references related to sparse linear algebra, and can also obtain actual statistics from runs of a variety of sparse matrix solvers on his/her own problem. Each expertise request leads to a number of elementary requests on a Grid of computers for which the DIET middleware developed by GRAAL is used. MUMPS is one of the packages interfaced within the project and that a user will be able to experiment through GRID TLSE. This year, part of the site was opened to the public: functionalities to share sparse matrices and bibliographic tools. For instance, test problems related to the Solstice project are shared among Solstice partners thanks to the TLSE site. Much work has also been done this year (mainly at ENSEEIHT-IRIT) to develop and validate the missing functionalities and be able to define expertise scenarios, with the goal to open the site to the public sometime in 2008.

7. Other Grants and Activities

7.1. Regional Projects

7.1.1. 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.

7.1.2. MUSINE: Franche-Comté: conception, validation et pilotage de la micro-usine multi-cellulaire (2007-2008)

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.

7.1.3. 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.

7.2. National Contracts and Projects

7.2.1. INRIA Grant: Software development for MUMPS ("Opération de Développement Logiciel")

INRIA has been financing Aurélia Fèvre, on contract from September 1, 2005 to August 31, 2007, as an engineer to work on the development of the MUMPS software package. This year, Aurélia has mainly worked on extending the test suite of MUMPS (non regression tests and improvement of the code coverage).

7.2.2. French ministry of research grant: Grid'5000, 3 years, 2004-2007

ENS Lyon is involved in the GRID'5000 project [79], which aims at building an experimental Grid platform gathering nine sites geographically distributed in France (17 laboratories). Each site hosts several clusters connected through the RENATER network.

GRAAL is participating in the design of the École normale supérieure de Lyon node. The scalability of DIET together with several scheduling heuristics will be evaluated on this platform.

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

The goal of this project is to gather researchers from the distributed systems and parallel algorithms communities in order to develop efficient and robust algorithms for some elementary applications, such as broadcast and multicast, distribution of tasks that may or may not share files, resource discovery. These fundamental applications correspond to the spectrum of the applications that can be considered on large scale, distributed platforms.

Yves Robert is leading the Rhône-Alpes site of this project, which comprises two other sites: Paris (LIX and LRI laboratories) and Bordeaux-Rennes (Paris and Scalapplix projects). Anne Benoit and Frédéric Vivien participate in this project, together with Lionel Eyraud, who held a post-doctoral position until August 31.

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

Grid computing platforms and components are subject to a great variability. Statistical models are mandatory to deal with changes in resource performance, such as CPU speeds or link bandwidths. Traditionally, Markov chains are used to capture the inherent uncertainty linked to parameter estimation. However, Markov chains lack a key feature: because they are memoryless, they cannot accurately model the performance of parallel systems periodically interacting through message exchanges in steady-state mode.

In contrast, sophisticated static scheduling strategies have been developed to map workflow applications on static Grid computing platforms. Optimal algorithms have been designed to map simple pipeline skeleton kernels onto heterogeneous clusters and Grids. Such applications operate in pipeline mode, and standard objective functions include maximizing the throughput and/or minimizing the response time (latency), for each data set.

A major goal of this project is to fill the gap between both approaches. On the one hand, statistical models are mandatory to account for the variability and dynamicity of resources. On the other hand, efficient scheduling algorithms only exist for static, dedicated platforms. We need a new stochastic model able to capture the performance of dynamic parallel systems accurately. This new model will be non-Markov for system interaction but will be Markov-based for platform characteristics (fault-tolerance and variability). The design and evaluation of this new model will be the first key contribution of the project. New, robust, scheduling algorithms will be designed and evaluated on top of this model, thereby providing the first stochastic testbed for workflow applications on Grid platforms. The third key contribution of the project will be the design of a prototype library for deploying workflow applications on computational Grids.

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

7.2.5. 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 in this project.

7.2.6. 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, Aurélia Fèvre and Jean-Yves L'Excellent participate to this project.

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

The objective of the Gwendia[8] 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.

7.2.8. SEISCOPE Consortium

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 (see publication [29]). 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 ANR (Agence National de la Recherche Française), and by BP, CGG, SHELL and TOTAL.

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

7.3. European Contracts and Projects

7.3.1. 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.

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

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.

7.4. International Contracts and Projects

7.4.1. Explora'Doc, Lawrence Berkeley National Laboratory, USA

Thanks to this grant from the Rhône-Alpes region, and thanks to additional funding from INRIA's "explorateur" program, PhD student E. Agullo spent a 6-month period in the scientific computing group of the Lawrence Berkeley National Laboratory (California, USA). In Berkeley, he worked under the supervision of Xiaoye S. Li. The collaboration aims at comparing two direct out-of-core approaches (multifrontal and left-looking) for solving large sparse linear systems.

7.4.2. France-Berkeley Fund Award (project starts in 2008)

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 collaboration will focus on massively parallel solvers for large sparse matrices and will reinforce the collaboration initiated by Emmanuel Agullo (see above). On the French side, this project also involves P. Amestoy (ENSEEIHT-IRIT), A. Guermouche (LaBRI) and I. Duff (CERFACS).

Emmanuel Agullo and Jean-Yves L'Excellent participate to this project.

7.4.3. 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.

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

7.4.4. MIT-France Fund Award (2007)

Multicore architectures are now entering the mainstream, as we can now find them on simple laptops. This architectural change induces a pressing demand for new solutions to existing problems, like the efficient execution of streaming applications such as image, video, and digital signal processing applications. In this collaboration with S. Amarasinghe and B. Thies from the MIT CSAIL laboratory, we target the efficient scheduling and mapping of streaming applications on multicore architectures, especially on heterogeneous multicore architectures. This collaboration is an extension on our work on steady-state scheduling.

Matthieu Gallet and Frédéric Vivien participate to this project.

7.4.5. 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 Manoa, 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.

7.4.6. 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. Dissemination

8.1. Scientific Missions

Aladdin is a proposal of an INRIA action of technological development for "A LArge-scale DIstributed and Deployable INfrastructure". In Aladdin, E. Caron is co-responsible of the working-group "Efficient and scalable composition and orchestration of services", and Frédéric Vivien of the working-group "Efficient exploitation of highly heterogeneous and hierarchical large-scale systems".

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.

8.2. Edition and Program Committees

Anne Benoit co-organized the Fourth International Workshop on Practical Aspects of High-level Parallel Programming (PAPP 2007), University of Beijing, China, May 2007; she is co-organizing the fifth edition of the workshop PAPP 2008, Krakow, Poland, June 2008.

A. Benoit was a member of the program committee of ICCS 2007, and she is a member of the program committee of ICCS 2008 and IPDPS 2008.

Yves Caniou was a member of the program committee of the Heterogeneous Computing Worshop 2007 (HCW'07) and of the International Conference on Computational Science and Applications (ICCSA'07). He 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 HCW'07 (Heterogeneous Computing Workshop), Long Beach, California, March 26 2007, held in conjunction with IPDPS 2007, and he is a program committee member of HCW'08. He was a member of the program committee of ISPA'07 (The Fifth International Symposium on Parallel and Distributed Processing and Applications), August 29-31, 2007. He was a member of the program committee of RenPar'2007 (Rencontres francophones du Parallélisme).

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 CLADE'07, High-Performance Scientific and Engineering Computing (HPCC'07), HPDC-16 (16th IEEE Int. Symp. on High-Performance Distributed Computing, 2007), ICCS'07, the tutorial program for SC07, Grid'2007, HeteroPar'07, and of the CoreGRID Symposium (within EuroPAR'07).

Jean-Yves L'Excellent is a member of the program committee of IPDPS'08 (Miami, Florida), and will be a member of the program committee of CSE'08 (Sao Paulo, Brazil).

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

Y. Robert is program chair of IPDPS'08 (IEEE International Parallel and Distributed Processing Symposium), Miami. He will be vice-chair of the "Scheduling and load balancing workshop" of EuroPar'2008.

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

Y. Robert gave an invited talk at:
- the ICCS'2007 conference, Beijing, China, May 2007;
- the WS'07 Workshop on Scheduling in Cetraro, Italy, June 2007;
- the POP'07 (Workshop on Parallelism Oblivious Programming), Tokyo, Japan, July 2007;
- the HeteroPar'07 conference, Austin, TX, USA, September 2007;
- Arny Fest (A Celebration of Arnold Rosenberg's Distinguished Career), Amherst, MA, USA, October 2007.

Bernard Tourancheau was a member of the program committees of EuroPVMMPI'07 and HPCC'07. He will co-organize CCGSC'08.

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

F. Vivien was a member of the program committee of RenPar 2008 (18e Rencontres francophones du Parallélisme), Fribourg, Switzerland, February 2008; EuroPDP 2008 (16th Euromicro Conference on Parallel, Distributed and Network-based Processing), Toulouse, France, February 2008; CoreGRID workshop on Grid Middleware 2007, Dresden, Germany, June 2007; Grid2007 (8th IEEE International Conference on Grid Computing), Austin, Texas, USA, September 19-21, 2007; Workshop on Scheduling for Parallel Computing, Gdansk, Poland, September 2007; PMGC 2007 (Workshop on Programming Models for Grid Computing), Rio de Janeiro, Brazil, May 2007.

Laurent Philippe is member of the program committee of CFSE, the French ACM Conference on Operating Systems.

8.3. Administrative and Teaching Responsibilities

8.3.1. Administrative Responsibilities

National University Committee (CNU) J.-M. Nicod is member of the computer science section of the National University Committee.

8.3.2. Teaching Responsibilities

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

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.

EPIT 2007. Y. Robert and F. Vivien organized the 35th (French) Spring school in theoretical computer science (EPIT 2007) in June 2007. The theme chosen for the school was *Scheduling algorithms*. During the school, an audience of around 40 students and researchers followed 11 talks.

9. 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", To appear, vol. 19, 2007.

[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", To appear, vol. 19, 2007.

[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, H. RENARD, Y. ROBERT, F. VIVIEN. *Mapping and load-balancing iterative computations on heterogeneous clusters with shared links*, in "IEEE Trans. Parallel Distributed Systems", vol. 15, n^o 6, 2004, p. 546-558.

Year Publications

Books and Monographs

[11] J. DONGARRA, B. TOURANCHEAU (editors). *Parallel Processing Letters (PPL), special issue for the Workshop on Clusters and Computational Grids for Scientific Computing*, vol. 17, n^o 1, World Scientific Publishing, March 2007.

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

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

[14] A. LEGRAND, H. CASANOVA, Y. ROBERT. *Parallel Algorithms*, To appear, CRC Press, 2008.

Articles in refereed journals and book chapters

[15] A. AMAR, R. BOLZE, Y. CANIOU, E. CARON, B. DEPARDON, J.-S. GAY, G. LE MAHEC, D. LOUREIRO. *Tunable Scheduling in a GridRPC Framework*, in "Concurrency & Computation: Practice & Experience", To appear, 2008.

[16] 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", To appear, vol. 19, 2008.

[17] O. BEAUMONT, L. MARCHAL. *Steady-state scheduling*, in "Introduction to Scheduling", To appear, Chapman and Hall/CRC Press, 2008.

[18] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Replica placement and access policies in tree networks*, in "IEEE Trans. Parallel Distributed Systems", To appear, vol. 19, 2008.

[19] A. BENOIT, Y. ROBERT. *Mapping pipeline skeletons onto heterogeneous platforms*, in "J. Parallel and Distributed Computing", To appear, 2008.

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

[21] E. CARON, F. DESPREZ, C. TEDESCHI. *Enhancing Computational Grids with Peer-to-Peer technology for Large Scale Service Discovery*, in "Journal of Grid Computing", vol. 5, n^o 3, September 2007, p. 337-360, http://www.springerlink.com/content/946324035g533050.

[22] M. COSNARD, Y. ROBERT. *Algorithmique parallèle*, in "Encyclopédie de l'Informatique et des Systèmes d'Information", Vuibert, 2007, p. 955-965.

[23] B. DEL-FABBRO, D. LAIYMANI, J.-M. NICOD, L. PHILIPPE. *DTM: a service for managing data persistency and data replication in network-enabled server environments*, in "Concurrency and Computation: Practice and Experience", vol. 19, n^o 16, November 2007, p. 2125-2140.

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

[25] 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", To appear, 2008.

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

[27] J.-S. GAY, Y. CANIOU. *Étude de la précision de Simbatch, une API pour la simulation de systèmes batch*, in "Special edition of TSI, Techniques et Sciences Informatiques", To appear, 2007, 21 p..

[28] L. MARCHAL, V. REHN, Y. ROBERT, F. VIVIEN. *Scheduling algorithms for data redistribution and load-balancing on master-slave platforms*, in "Parallel Processing Letters", vol. 17, n^o 1, 2007, p. 61-77.

[29] S. OPERTO, J. VIRIEUX, P. AMESTOY, J.-Y. L'EXCELLENT, L. GIRAUD, H. BEN HADJ ALI. *3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study*, in "Geophysics", vol. 72, n^o 5, 2007, p. SM195-SM211, http://link.aip.org/link/?GPY/72/SM195/1.

[30] J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *The impact of heterogeneity on master-slave scheduling*, in "Parallel Computing", To appear, 2008.

[31] C. REZVOY, D. CHARIF, L. GUEGUEN, G. A. MARAIS. *MareyMap: a R-based tool with graphical interface for estimating recombination rates*, in "Bioinformatics", vol. 23, n^o 16, 2007, p. 2188-2189.

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

[33] W. THIES, F. VIVIEN, S. AMARASINGHE. *A step towards unifying schedule and storage optimization*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", vol. 29, n^o 6, 2007, 45 p..

Publications in Conferences and Workshops

[34] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *On Reducing the I/O Volume in Sparse Out-of-core Solver*, in "HiPC'07 14th International Conference On High Performance Computing, Goa, India", Lecture Notes in Computer Science, n^o 4873, December 17-20 2007, p. 47-58.

[35] 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", To appear, 2008.

[36] G. ANTONIU, E. CARON, F. DESPREZ, A. FÈVRE, M. JAN. *Towards a Transparent Data Access Model for the GridRPC Paradigm*, in "HiPC'2007. 14th International Conference on High Performance Computing., Goa. India", LNCS, n° 4873, Springer Verlag Berlin Heidelberg, December 17-20 2007, p. 269-284.

[37] 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 on Parallel and Distributed Processing Symposium (APDCM 2008)", To appear, IEEE Computer Society Press, 2008.

[38] A. BENOIT, V. REHN, Y. ROBERT. *Strategies for replica placement in tree networks*, in "HCW'2007, the 16th Heterogeneous Computing Workshop", IEEE Computer Society Press, 2007.

[39] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Impact of QoS on replica placement in tree networks*, in "ICCS'2007, the 7th 2007 International Conference on Computational Science", LNCS 4487, Springer Verlag, 2007, p. 366-373.

[40] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Multi-criteria scheduling of pipeline workflows*, in "HeteroPar'2007: International Conference on Heterogeneous Computing, jointly published with Cluster'2007", IEEE Computer Society Press, 2007.

[41] A. BENOIT, Y. ROBERT. *Complexity results for throughput and latency optimization of replicated and data-parallel workflows*, in "HeteroPar'2007: International Conference on Heterogeneous Computing, jointly published with Cluster'2007", IEEE Computer Society Press, 2007.

[42] A. BENOIT, Y. ROBERT. *Mapping pipeline skeletons onto heterogeneous platforms*, in "ICCS'2007, the 7th International Conference on Computational Science", LNCS 4487, Springer Verlag, 2007, p. 591-598.

[43] Y. CANIOU, E. CARON, H. COURTOIS, B. DEPARDON, R. TEYSSIER. *Cosmological Simulations using Grid Middleware*, in "Fourth High-Performance Grid Computing Workshop (HPGC'07), Long Beach, California, USA", IEEE, March 26 2007.

[44] E. CARON, P. K. CHOUHAN, F. DESPREZ. *Automatic Middleware Deployment Planning on Heterogeneous Platfoms*, in "The 17th Heterogeneous Computing Workshop (HCW'08)., Miami, Florida", To appear, In conjunction with IPDPS 2008., April 2008.

[45] E. CARON, F. DESPREZ, F. PETIT, C. TEDESCHI. *Snap-stabilizing Prefix Tree for Peer-to-peer Systems*, in "9th International Symposium on Stabilization, Safety, and Security of Distributed Systems, Paris, France", Lecture Notes in Computer Science, vol. 4838, Springer Verlag Berlin Heidelberg, November 2007, p. 82-96.

[46] 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", To appear, In conjunction with IPDPS 2008., April 2008.

[47] 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", To appear, 2008.

[48] F. DESPREZ, A. VERNOIS. *Semi-Static Algorithms for Data Replication and Scheduling Over the Grid*, in "IEEE 3rd International Conference on Intelligent Computer Communication and Processing, workshop on Grid Computing, Cluj-Napoca, Romania", September 2007.

[49] J. DIGIOVANNA, L. MARCHAL, P. RATTANATAMRONG, M. ZHAO, S. DARMANJIAN, B. MAHMOUDI, J. SANCHEZ, J. PRÍNCIPE, L. HERMER-VAZQUEZ, R. FIGUEIREDO, J. FORTES. *Towards Real-Time Distributed Signal Modeling for Brain Machine Interfaces*, in "Proceedings of Dynamic Data Driven Application Systems (workshop of ICCS)", LNCS, vol. 4487, Springer Verlag, 2007, p. 964-971.

[50] 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", To appear, 2008.

[51] J. DONGARRA, J.-F. PINEAU, Y. ROBERT, Z. SHI, F. VIVIEN. *Revisiting matrix product on master-worker platforms*, in "9th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2007", IEEE Computer Society Press, 2007.

[52] J. DONGARRA, J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Matrix Product on Heterogeneous Master-Worker Platforms*, in "13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Salt Lake City, Utah", To appear, February 20-23 2008.

[53] L. EYRAUD-DUBOIS, A. LEGRAND, M. QUINSON, F. VIVIEN. *A First Step Towards Automatically Building Network Representations*, in "Proceedings of Euro-Par 2007", LNCS, vol. 4641, 2007, p. 160-169.

[54] M. GALLET, Y. ROBERT, F. VIVIEN. *Scheduling communication requests traversing a switch: complexity and algorithms*, in "PDP'2007, 15th Euromicro Workshop on Parallel, Distributed and Network-based Processing", IEEE Computer Society Press, 2007, p. 39-46.

[55] M. GALLET, Y. ROBERT, F. VIVIEN. *Scheduling multiple divisible loads on a linear processor network*, in "ICPADS'2007, the 13th International Conference on Parallel and Distributed Systems", 2007.

[56] L. MARCHAL, V. REHN, Y. ROBERT, F. VIVIEN. *Scheduling and data redistribution strategies on star platforms*, in "PDP'2007, 15th Euromicro Workshop on Parallel, Distributed and Network-based Processing", IEEE Computer Society Press, 2007, p. 288-295.

[57] V. REHN-SONIGO. *Optimal Closest Policy with QoS and Bandwidth Constraints for Placing Replicas in Tree Networks*, in "CoreGRID'2007, Core GRID Symposium 2007", Springer Verlag, 2007.

Internal Reports

[58] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Reducing the I/O Volume in an Out-of-core Sparse Multifrontal Solver*, Also appeared as LIP report RR2007-22, Research Report, n^{o} RR-6207, INRIA, May 2007, https://hal.inria.fr/inria-00150588.

[59] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Towards a Parallel Out-of-core Multifrontal Solver: Preliminary Study*, Also available as LIP report RR2007-06, Research report, n^{o} RR-6120, INRIA, February 2007, http://hal.inria.fr/inria-00130278.

[60] A. BENOIT, L. MARCHAL, J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Offline and Online Scheduling of Concurrent Bags-of-Tasks on Heterogeneous Platforms*, Research Report, n° RR-6401, INRIA, 2007, http://hal.inria.fr/inria-00200261/.

[61] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Multi-criteria scheduling of pipeline workflows*, Also available as LIP research report 2007-32, Research Report, n° RR-6232, INRIA, 2007, https://hal.inria.fr/inria-00156732.

[62] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Optimizing Latency and Reliability of Pipeline Workflow Applications*, Also available as LIP research report 2007-43, Research Report, n° 6345, INRIA, November 2007, https://hal.inria.fr/inria-00186152.

[63] A. BENOIT, Y. ROBERT. *Complexity results for throughput and latency optimization of replicated and data-parallel workflows*, Research Report, n° RR-6308, INRIA, 2007, http://hal.inria.fr/inria-00175066.

[64] L. EYRAUD-DUBOIS, A. LEGRAND, M. QUINSON, F. VIVIEN. *A First Step Towards Automatically Building Network Representations*, Also available as LIP research report 2007-08, Research Report, n° RR-6133, INRIA, February 2007, https://hal.inria.fr/inria-00130734.

[65] 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"*, Also available as LIP research report 2007-07, Research Report, n° RR-6123, INRIA, February 2007, https://hal.inria.fr/inria-00130294.

[66] M. GALLET, Y. ROBERT, F. VIVIEN. *Scheduling multiple divisible loads on a linear processor network*, Research Report, n° RR-6235, INRIA, 2007, http://hal.inria.fr/inria-00158027.

[67] V. REHN-SONIGO. *Optimal Replica Placement in Tree Networks with QoS and Bandwidth Constraints and the Closest Allocation Policy*, Also available as LIP research report 2007-10, Research Report, n° 6233, INRIA, 2007, https://hal.inria.fr/inria-00156747.

[68] Y. TANIMURA, K. SEYMOUR, E. CARON, A. AMAR, H. NAKADA, Y. TANAKA, F. DESPREZ. *Interoperability Testing for The GridRPC API Specification*, OGF Reference: GFD.102, Open Grid Forum, May 2007, http://www.ogf.org/documents/GFD.102.pdf.

References in notes

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

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

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

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

[73] *GRID TLSE*, http://www.gridtlse.org/.

[74] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *A Preliminary Out-of-core Extension of a Parallel Multifrontal Solver*, in "EuroPar'06 Parallel Processing", 2006, p. 1053–1063.

[75] 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", Submitted, 2008.

[76] 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.

[77] 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° CS-01-467, University of Tennessee, Knoxville, TN, July 2001, http://www.cs.utk.edu/netsolve/.

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

[79] F. CAPPELLO, F. DESPREZ, M. DAYDE, E. JEANNOT, Y. JEGOU, S. LANTERI, N. MELAB, R. NAMYST, P. PRIMET, O. RICHARD, E. CARON, J. LEDUC, G. MORNET. *Grid'5000: A Large Scale, Reconfigurable, Controlable and Monitorable Grid Platform*, in "Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing, Grid'2005, Seattle, Washington, USA", November 2005.

[80] 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.

[81] 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.

[82] 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.

[83] H. EL-REWINI, H. H. ALI, T. G. LEWIS. *Task Scheduling in Multiprocessing Systems*, in "Computer", vol. 28, n° 12, 1995, p. 27–37.

[84] 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.

[85] M. FERRIS, M. MESNIER, J. MORI. *NEOS and Condor: Solving Optimization Problems Over the Internet*, in "ACM Transactions on Mathematical Sofware", vol. 26, n° 1, 2000, p. 1-18, http://www-unix.mcs.anl.gov/metaneos/publications/index.html.

[86] C. GERMAIN, G. FEDAK, V. NÉRI, F. CAPPELLO. *Global Computing Systems*, in "Lecture Notes in Computer Science", vol. 2179, 2001, p. 218–227.

[87] G. KRAUSS, B. LIPS, J. VIRGONE, E. BLANCO. *Modelisation sous TRNSYS d'une maison a energie positive*, in "IBPSA France", I. B. P. S. ASSOCIATION (editor), Nov 2006, http://www.ibpsa-france.net/.

[88] 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.

[89] 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.

[90] 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° 2536, November 2002, p. 274-278.

[91] 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° 5-6, 1999, p. 649-658.

[92] M. G. NORMAN, P. THANISCH. *Models of Machines and Computation for Mapping in Multicomputers*, in "ACM Computing Surveys", vol. 25, n° 3, 1993, p. 103–117.

[93] J. K. REID, J. A. SCOTT. *An out-of-core sparse Cholesky solver*, Revised March 2007, Technical report, n° RAL-TR-2006-013, Rutherford Appleton Laboratory, 2006.

[94] 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.

[95] B. A. SHIRAZI, A. R. HURSON, K. M. KAVI. *Scheduling and Load Balancing in Parallel and Distributed Systems*, IEEE Computer Science Press, 1995.

[96] 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° 5–6, October 1999, p. 757–768.

[97] H. M. WONG, B. VEERAVALLI, G. BARLAS. *Design and performance evaluation of load distribution strategies for multiple divisible loads on heterogeneous linear daisy chain networks*, in "J. Parallel Distributed Computing", vol. 65, n° 12, 2005, p. 1558-1577.

[98] H. M. WONG, B. VEERAVALLI. *Scheduling divisible loads on heterogeneous linear daisy chain networks with arbitrary processor release times*, in "IEEE Trans. Parallel Distributed Systems", vol. 15, n° 3, 2004, p. 273-288.