INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team GRAAL

Algorithms and Scheduling for Distributed Heterogeneous Platforms

Rhône-Alpes

THEME NUM

Activity Report

2005

Table of contents

1. Team

Head of project-team
Frédéric Desprez [DR INRIA]

Administrative assistants
Sylvie Boyer [INRIA, 30% on the project]

INRIA staff
Frédéric Desprez [DR]
Jean-Yves L'Excellent [CR]
Frédéric Vivien [CR]

Faculty members from ENS Lyon
Anne Benoît [Assistant Professor, starting September 1, 2005]
Aurélien Bouteiller [Lecturer, starting September 1, 2005]
Yves Caniou [Lecturer, until August 31, 2005]
Eddy Caron [Assistant Professor]
Yves Robert [Professor]

Faculty members from Université de Franche-Comté
Jean-Marc Nicod [Assistant Professor]
Laurent Philippe [Professor]

Faculty members from Université Lyon 1
Yves Caniou [Assistant Professor, starting September 1, 2005]

Project technical staff
Éric Boix [CNRS, 50% on the project]
Raphaël Bolze [on contract, until September 30, 2005]
Holy Dail [on contract from INRIA]
Aurélia Fèvre [on contract from INRIA, starting September 1, 2005]

Post-doctoral fellow
Abdelkader Amar [INRIA grant]
Alan Su [Rhône-Alpes region grant]

Ph. D. students
Emmanuel Agullo [MENRT grant]
Raphaël Bolze [BDI CNRS, starting October 1, 2005]
Pushpinder-Kaur Chouhan [MENRT grant]
Sylvain Dahan [Rhône-Alpes region grant]
Bruno Del Fabbro [FAF grant]
Jean-Sébastien Gay [Rhône-Alpes region grant]
Loris Marchal [ENS grant]
Suphakit Niwattanakul [Thailand grant]
Jean-François Pineau [ENS grant]
Hélène Renard [MENRT grant (ACI GRID)]
Cédric Tedeschi [MENRT grant]
Antoine Vernois [MENRT grant (ACI GRID)]

2. Overall Objectives

2.1. Overall Objectives

Keywords: *algorithmics for heterogeneous systems*, *distributed application*, *grid computing*, *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 [93]. 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, 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 [103]. 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 13.6 Teraflops. At a smaller scale but with a high bandwidth, one can mention the RNRT VTHD++ project [1] which connects several France Telecom and INRIA research centers (and the PC clusters available in those centers) and several other laboratories (including ours) with a 2.5 Gb/s network. On such a platform, the network between the research centers is even faster than the network within each cluster connected to it. 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 SMP 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 [99], [91], [93]. Today's environments are based either on the adaptation of "classical" parallel environments or on the development of toolboxes based on Web Services.

2.1.1. 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.

One strength of our project has always been its activities of transfer to the industry and its international collaborations. Among recent collaborations, we can mention

- collaboration with Sun Labs Europe for the deployment of Application Service Provider (ASP) environments over the Grid,
- collaboration with the GRAIL Lab. at University of California, San Diego, on scheduling for heterogeneous platforms and the development of a simulator of schedulers for heterogeneous architectures,

[1] Réseau à Vraiment Très Haut Débit (*Network with Very High Broadband*).

- collaboration with ICL Lab. at University of Tennessee, Knoxville around the *ScaLAPACK* library for parallel linear algebra and the NetSolve environment which are both internationally distributed.

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

3. Scientific Foundations

3.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Anne Benoît, Loris Marchal, Jean-François Pineau, Hélène Renard, Yves Robert, Alan Su, 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 [92], [102], [113], [116] and the references therein. This model was introduced for homogeneous processors, and has been (straightforwardly) extended to heterogeneous computing resources. In a word, there is a limited number of computing resources, or processors, to execute the tasks. Communication delays are taken into account as follows: let task T be a predecessor of task T' in the task graph; if both tasks are assigned to the same processor, no communication overhead is incurred, the execution of T' can start immediately at the end of the execution of T; on the contrary, if T and T' are assigned to two different processors P_i and P_j, a communication delay is incurred. More precisely, if P_i completes the execution of T at time-step t, then P_j cannot start the execution of T' before time-step $t + \text{comm}(T,T', P_i, P_j)$, where $\text{comm}(T,T', P_i, P_j)$ is the communication delay, which depends upon both tasks T and T' and both processors P_i and P_j. Because memory accesses are typically several orders of magnitude cheaper than inter-processor communications, it is sensible to neglect them when T and T' are assigned to the same processor.

The major flaw of the macro-dataflow model is that communication resources are not limited in the 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 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 of them related to simulation: geophysics, chemistry, electromagnetism, structural optimization, computational fluid dynamics, ... 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 larger and larger problems arising 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 [100], [101], for symmetric positive definite, general symmetric or unsymmetric problems, with numerical pivoting 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 [109] which results from the graph structure corresponding to the nonzero pattern of the problem to be solved, and 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 as much adaptive as possible to existing and newer parallel computer architectures, we are especially interested in approaches that are intrinsically dynamic and asynchronous [96], [97]. 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 acting 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 platform MUMPS (see Section 5.2) which is our main platform to experiment and validate new ideas and research directions. Finally, note that for very large problems (tens of millions of equations), either parallel out-of-core approaches are required, or direct solvers should be used within an iterative scheme, leading to hybrid direct-iterative methods.

3.3. Providing Access to HPC Servers on the Grid

Participants: Raphaël Bolze, Yves Caniou, Eddy Caron, Pushpinder-Kaur Chouhan, Sylvain Dahan, Holy Dail, Bruno Del Fabbro, Frédéric Desprez, Jean-Sébastien Gay, Jean-Marc Nicod, Laurent Philippe, Alan Su, Antoine Vernois, 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 [2] 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 [3] or Sun GridEngine [4]. Finally, peer-to-peer [94] or Global computing [106] 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 [110], [111]) 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 [98], Ninf [112], NEOS [104], OmniRPC [115], 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,
- online and offline scheduling of requests,
- link with data management,
- distributed algorithms to manage the requests and the dynamic behavior of the platform.

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

4. Application Domains

4.1. Applications of Sparse Direct Solvers

Our activity on sparse direct (multifrontal) solvers in distributed environments goes as far as building competitive software available to users. Such 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 differences, or requires the optimization of a complex linear or nonlinear function, one almost always ends up in solving a system of equations involving sparse matrices. There are therefore a number of application fields, among which we can list the most frequently cited by our users, i.e. the applications in which our sparse direct solver MUMPS (see Section 5.2) has been or is currently used: 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, 3D wave propagation in inhomogeneous media for geophysical or optical 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 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, education, astrophysics (e.g., supernova, thermonuclear reaction networks, neutron diffusion equation, quantum chaos, quantum transport), research on domain decomposition (MUMPS can for example be used inside each subdomain and return Schur complements handled by an iterative method), circuit simulations, etc.

Notice that the MUMPS users include:

- students and academic users from all over the world: Europe, USA, Corea, India, Argentina, Brazil, etc;
- various developers of finite element software;
- companies such as Dassault, EADS, NEC, or Samtech.

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. The current version is LAMMPS 2001, which 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 UCBL.

4.3. Geographical Application Based on Digital Elevation Models

This parallel application is based on a stereo vision algorithm. We focus on the particular stereo vision problem of accurate Digital Elevation Models (DEMs) reconstruction from a pair of images of the SPOT satellite. We start from an existing algorithm and optimize it while focusing on the cross-correlation problem based on a statistical operator.

The input data consists in two images from the SPOT satellite of a particular region taken from different points of view. From these images, we extract the three-dimensional information by finding couples of corresponding points and computing 3D coordinates using camera information. Then, for each pixel in this image, we try to find its counterpart in the other image. We can restrict the search domain of counterparts by transforming input images in epipolar geometry. This geometry, based on optical principles, has the very interesting feature to align the corresponding points on the same lines of images. Then, the search domain is drastically reduced to at most one image line. Nonetheless, the input data size may be very large especially from satellite imagery which produces 6000×6000-pixel images, involving important computation times as well as very large memory demand. We used the DIET architecture to solve this problem in collaboration with the Earth Science Laboratory (LST ENS Lyon).

4.4. Electronic Device Simulation

The determination of circuit and device interaction appears to be one of the major challenges of mobile communication engineering in the next few years. The ability to design simultaneously (co-design) devices and circuits will be a major feature of CAD tools for the design of MMIC circuits. The coupling of circuit simulators and physical simulators is based either on time-domain methods or harmonic balance methods (HB). Our approach consists in the direct integration of physical HBT model in a general circuit simulator. Thus, the popular HB formulation has been adopted in the proposed approach coupled to a fully implicit discretization scheme of device equations. The resulting software allows the optimization of circuit performance in terms of physical and geometrical parameter devices as well as in terms of terminating impedances. This result has been achieved by making use of dedicated techniques to improve convergence including the exact Jacobian matrix computation of the nonlinear system that has to be solved. This application requires high performance computation and heavy resources, because of the size of the problem. This application is well adapted to metacomputing and parallelism. In collaboration with the laboratory IRCOM (UMR CNRS/University of Limoges), this application is being ported to DIET.

4.5. 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 allows 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 computation. 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.

4.6. 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. Functional sites and signatures of proteins are very useful for analyzing these 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, and to the clusterization into protein families of the sequences contained in international databanks.

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.

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 genomic research field: it can provide scientists with a transparent access to large computational and data management resources. DIET will be used as one Grid platform.

5. Software

5.1. DIET

Participants: Raphaël Bolze, Yves Caniou, Eddy Caron, Pushpinder-Kaur Chouhan, Sylvain Dahan, Holy Dail, Frédéric Desprez [correspondent], Bruno Del Fabbro, Jean-Sébastien Gay, Jean-Marc Nicod, Laurent Philippe, Alan Su, Cédric Tedeschi, Antoine Vernois.

Huge problems can now be computed 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) [118] 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 (see below).

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 from 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 the clients. 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 FAST module, which is described in the next section.

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

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

DIET has been validated on several applications. Some of them have been described in Section 4.

5.2. MUMPS

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

MUMPS (for *MUltifrontal Massively Parallel Solver*) 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, and in collaboration mainly with ENSEEIHT-IRIT (Toulouse, France), lots of developments have been done, to enhance the software with more functionalities and integrate recent research work.

MUMPS uses 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,
- 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 several hundred academic and industrial users, from a wide range of application fields (see Section 4.1). Recent work related to performance scalability, preprocessing of both symmetric and unsymmetric matrices, two by two pivots for symmetric indefinite problems, and dynamic scheduling has been incorporated in the new improved version of the package (release 4.5.5 available since October 2005 at http://graal.ens-lyon.fr/MUMPS/avail.html). Scilab and Matlab interfaces, as well as preliminary work towards an out-of-core version for largest test problems will be available in a future release.

6. New Results

6.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

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

Participants: Anne Benoît, Loris Marchal, Jean-François Pineau, Hélène Renard, Yves Robert, Alan Su, 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.

We have written a survey paper on steady-state scheduling techniques for heterogeneous systems, such as clusters and grids. In this survey, we give several successful examples of this approach, before discussing its limitations. In a nutshell, successful examples correspond to problems where determining the optimal throughput, as well as reconstructing the final (periodic) schedule, can be achieved in polynomial time. However, there are problems for which even the sole determination of the best throughput remains difficult.

An example of such problems appears in our work on mixed task and data parallelism. We have considered steady-state scheduling techniques for mapping a collection of application graphs onto heterogeneous platforms. We have shown that the most general instance of this problem is NP-complete. However, most situations of practical interest are amenable to a periodic solution which can be described in compact form (polynomial size) and is asymptotically optimal. In other words, for “reasonable” application graphs and arbitrary platform graphs, steady-state scheduling is a viable approach to the problem.

Finally, we have investigated the impact of memory constraints (limited buffer capacity) when mapping independent tasks onto star-shaped platform graphs. Not surprisingly, finding the optimal throughput becomes NP-hard, but we have designed polynomial heuristics that deliver an efficient throughput, as confirmed by a wide range of simulations.

6.1.2. Pipelined Execution of Macro-communication Schemes

When analyzing the communications involved by the execution of complex applications, deployed on a heterogeneous “grid” platform, we see that such applications intensively use collective macro-communication schemes, such as scatters, personalized all-to-all or gather/reduce operations. As explained above, rather than aiming at minimizing the execution time of a single macro-communication, we focus on the steady-state operation. We assume that there is a large number of macro-communications to perform in pipeline fashion, and we aim at maximizing the throughput, i.e., the (rational) number of macro-communications which can be initiated every time-step. It is worth pointing out that optimal algorithms for series of broadcasts, say, will also prove asymptotically optimal for the problem of a single broadcast with a long message (because the long message will be split into slices whose diffusion will be pipelined across the platform).

While we had provided polynomial solutions for series of scatters, series of personalized all-to-all, series of reduce operations, and series of broadcasts, we have shown that computing the best throughput for a multicast operation is NP-hard. Thus we have introduced several heuristics to deal with this problem; most of them are based on linear programming. We prove that some of these heuristics are approximation algorithms. We perform simulations to test these heuristics and show that their results are close to a theoretical upper bound on the throughput that we obtain with the linear programming approach.

We have also investigated the series of broadcasts problem under a new communication model, the unidirectional one-port model: at a given time step, a processor can be involved in at most one (incoming or outgoing) communication. Achieving the best throughput may well require that the target platform is used in totality: we show that neither spanning trees nor DAGs are as powerful as general graphs. We propose a (rather sophisticated) polynomial algorithm for determining the optimal throughput that can be achieved using a platform, together with a (periodic) schedule achieving this throughput. The algorithm is based on the use of polynomial oracles and of the ellipsoid method for solving linear programs in rational numbers. The polynomial compactness of the description comes from the decomposition of the schedule into several broadcast trees that are used concurrently to reach the best throughput. It is important to point out that a concrete scheduling algorithm based upon the steady-state operation is asymptotically optimal, in the class of all possible schedules (not only periodic solutions).

Finally, we have provided a more practical approach for the "classical" single broadcast problem, with the traditional bidirectional one-port model, and extensions to limited multi-port capabilities. Typically, the message to be broadcast is split into several slices, which are sent by the source processor in a pipeline fashion. A spanning tree is used to implement this operation, and the objective is to find the tree which maximizes the throughput, i.e., the average number of slices sent by the source processor every time-unit. We introduce several heuristics to solve this problem. The good news is that the best heuristics perform quite efficiently, reaching more than 70% of the absolute optimal throughput, thereby providing a simple yet efficient approach to achieve very good performance for broadcasting on heterogeneous platforms.

6.1.3. Divisible Loads

Divisible load applications consist of an amount of data and associated computation that can be divided arbitrarily into any number of independent pieces. This model is a good approximation of many real-world scientific applications, lends itself to a natural master-worker implementation, and has thus received a lot of attention [114]. The critical issue of divisible load scheduling has been studied extensively in previous work. However, only a few authors have explored the simultaneous scheduling of multiple such applications on a distributed computing platform. We focus on this increasingly relevant scenario and make the following contributions. We use a novel and more realistic platform model that captures some of the fundamental network properties of Grid platforms. We formulate a steady-state multi-application scheduling problem as a linear program that expresses some notion of fairness between applications. This scheduling problem is NP-complete and we propose several heuristics that we evaluate and compare via extensive simulation experiments conducted over 250,000 platform configurations. Our main finding is that some of our heuristics can achieve performance close to the optimal and we quantify the trade-offs between achieved performance and heuristic complexity.

We have also revisited the traditional divisible load problem, namely scheduling independent tasks onto an heterogeneous star platform. We consider the case where the workers, after processing the tasks, send back some results to the master processor. This corresponds to a more general framework than the one used in many divisible load papers, where only forward communications are taken into account. To the best of our knowledge, this work constitutes the first attempt to derive optimality results under this general framework (forward and backward communications, heterogeneous processing and communication resources). We prove that it is possible to derive the optimal solution both for LIFO and FIFO distribution schemes. Nevertheless, the complexity of the general problem remains open. We also prove that the optimal distribution scheme may be neither LIFO nor FIFO.

6.1.4. Off-line and on-line scheduling

We have considered the problem of scheduling comparisons of motifs against biological databanks. We have shown that this problem can be expressed within the divisible load framework. In this framework, we proposed a polynomial-time algorithm to solve the maximum weighted flow off-line scheduling problem on unrelated machines. We have also shown how to solve the maximum weighted flow off-line scheduling problem with preemption on unrelated machines. This set of results gives us theoretical bounds against which we are able to compare any on-line solution.

After this study of the theoretical off-line version of our problem, we looked at its on-line version. Scheduling divisible loads on uniform machines is equivalent to scheduling preemptible jobs on a single machine. We therefore first studied the on-line version of scheduling with preemption on a single machine. We improved the best known lower bound on the competitive ratio of on-line algorithms minimizing either the maximum stretch or the average stretch. Then we designed some new online heuristics for the maximum stretch minimization. We showed through extensive simulations that our heuristics performed better than the existing approximation algorithms.

In a different context, we have started an exhaustive study of the complexity of off-line and on-line scheduling of non-preemptible jobs on a master-slave platform. The originality of this work lies in the fact that we consider communication costs between the master and the slaves. For the simpler case with homogeneous communications and computations, we were able to derive optimal algorithms for the online case. Then we derived lower bounds on the competitive factor of any online algorithm for the different other cases : homogeneous communications, homogeneous computations, heterogeneous communications and computations. This showed the impact of heterogeneity on some of the main objectives functions (makespan, maximum flow, and average flow).

6.1.5. Load-Balancing for Communication-Aware Models

For all our studies we use communication models as realistic as possible. In communication-aware models, there is a limited number of communication links, and these links have bounded bandwidths. Furthermore, the use of the communication links can be restricted in various manners:

1. Each processor may be given a routing table which specifies the links to be used to communicate with each other processor (hence the routing is fully static). Another hypothesis is to assume a dynamic routing, which is computed on the fly so as to optimize the network use.
2. At most one message can circulate on one link at a given time-step, so that contention for communication resources is taken into account statically. Another hypothesis is that several messages can circulate on one link at a given time-step, but the different messages share the total link bandwidth. The eXplicit Control Protocol XCP [108], for example, does enable to implement a bandwidth allocation strategy that complies with our hypotheses.

Following our work on load balancing for communication-aware models, we have investigated redistribution algorithms for homogeneous and heterogeneous rings of processors. The problem arises in several applications, each time that a load-balancing mechanism is invoked (but we do not discuss the load-balancing mechanism itself). We have provided algorithms that aim at optimizing the data redistribution, both for unidirectional and bi-directional rings. One major contribution of this work is that we are able to prove the optimality of the proposed algorithms in all cases except that of a bi-directional heterogeneous ring, for which the problem remains open (we do not even know whether the problem is NP-complete).

6.1.6. Tasks Sharing Files

Most of the time, the tasks to be scheduled depend on files (or more generally, data). As we map a task to a processor, we also map the files which this task depends upon. Thus, we must take into account the communications needed to send a file from the server originally storing it to the processor executing the task. Furthermore, some files may be shared by several tasks and the scheduling strategies can either map several tasks sharing a file on the same processor (which may induce load-imbalance) or replicate files among processors (which may induce communication overheads).

We extended our work on the case where we have to schedule a large collection of independent tasks onto a large distributed heterogeneous platform, which is composed of a set of servers. Each server is a processor cluster equipped with a file repository. The (input) files are initially distributed on the server repositories. For each task, the problem is to decide on which server to execute it, and to transfer the required files (those which the task depends upon) to that server repository. On the theoretical side, we established complexity results that assess the difficulty of the problem. On the practical side, we designed several new heuristics, including an extension of the `min-min` heuristic to our decentralized framework, and several lower cost heuristics, which we compared through extensive simulations.

6.1.7. Network Resource Sharing

In collaboration with the RESO team, we have investigated the problem of network resource sharing in the context of grid scheduling. Given a set of communication requests, one can separate grid sites into ingress and egress points: traffic flow enters the network from ingress points, and leaves it from egress points. With the assumption of ample resources in the network core, these ingress and egress points at the network edge are where network resource bottlenecks arise. Resource requests, corresponding to different application scenarios, can be classified as long-lived and short-lived. The fundamental difference is that short-lived requests have time windows specified. Scheduling optimization problems are formulated, and proven NP-hard, for both types of requests. Several heuristic algorithms are designed. Simulation results show that the heuristics achieve fairly satisfying performance.

6.1.8. Applications Mapping

We have investigated the problem of the mapping of an application together with colleagues of the University of Pisa, Italy. This work is aimed at grid technologies. Since the programming and optimization burden of a low level approach to grid computing is clearly unacceptable for large scale, complex applications, we focus in this work on high-level programming environments, which aim to simplify the development of grid applications.

We thus address the problem of the mapping of a high-level grid application onto the computational resources. In order to optimize the mapping of the application, we propose to automatically generate performance models from the application using the process algebra PEPA. We target applications written with the high-level environment ASSIST, since the use of such a structured environment allows us to automate the study of the application more effectively. The first results obtained on a data mining classification algorithm have shown that this approach is promising.

6.2. Providing access to HPC servers on the Grid

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

Participants: Raphaël Bolze, Yves Caniou, Eddy Caron, Pushpinder-Kaur Chouhan, Sylvain Dahan, Holy Dail, Frédéric Desprez, Bruno Del Fabbro, Jean-Sébastien Gay, Jean-Marc Nicod, Laurent Philippe, Alan Su, Cédric Tedeschi, Antoine Vernois, Frédéric Vivien.

6.2.1. Deadline Scheduling

We developed algorithms for scheduling sequential tasks for client-server systems on the grid as an extension of the paper: "A Study of Deadline Scheduling for Client-Server Systems on the Computational Grid" [117]. We mainly focused on the management of tasks with respect to their priorities and deadlines. Load correction using FAST and fallback mechanisms are used to increase the number of executed tasks. We also presented an algorithm that considers both priority and deadline of the tasks to select a server.

We showed through experiments that the number of tasks that can meet their deadlines can be increased by 1) using task priorities and by 2) using a fallback mechanism to reschedule tasks that were not able to meet their deadline on the selected servers.

6.2.2. Hierarchical scheduling

The DIET scheduling process is distributed with servers providing their own performance predictions and agents forming a distributed sorting network for server selection. Effective scheduling algorithms are a key component of any NES environment. The distributed nature of DIET scheduling provides the unusual opportunity to experiment with distributed scheduling algorithms in a efficient, stable middleware package that can be easily used in heterogeneous, distributed resource environments. We added extensions to DIET scheduling to support control over request flows at various levels in the DIET hierarchy.

The first extension provides the ability to limit the number of concurrent jobs that can run on a server at a time. This feature can be important under high-load conditions or for very resource-intensive jobs to avoid overloading the server. The second extension involved adding a level of global task scheduling control at the master agent. This extension provides two advantages: the ability to control the flow of requests into the DIET system, thus avoiding the problem of scheduling too far in advance under heavy load conditions; and the ability to re-arrange task placements within a scheduling window.

We then presented experiments comparing the performance of the standard DIET scheduling approach against that of the SeD queue and global task scheduler approaches [63]. We demonstrated that for a bursty client scenario, the SeD queue can improve application makespan as well as mean turnaround time. We then tested the performance of the standard and global approaches when system conditions change. These experiments demonstrated that the global approach is better able to adapt to changing system conditions because it does not schedule as far in advance as the standard approach. Finally, we presented the results of long-running experiments with a steady, heavy client load. In these experiments the three approaches showed very similar performance overall. It is to be expected here that the SeD queue and Global approaches did not improve performance: in such a heavy load situation all of the servers are kept busy all of the time by all of the strategies, and without task affinities it is not important how and when the requests are allocated to servers. This indicates that we succeeded in introducing new functionality to the standard scheduling approach without losing the performance benefits of the standard approach.

6.2.3. *Large Scale Service Lookup*

The first DIET prototype was based on an agent hierarchy with a Master Agent as a root. To further increase the overall scalability of the system, multiples MAs can be connected together. If a service is not locally available, the Master Agent will forward the request to its neighbors (i.e., other Master Agents). The request will be broadcast on the network of MA until an agent finds the appropriate server. Two approaches have been used to implement this interconnection. One (6.2.4) is based on the JXTA [107] Peer-to-peer platform and the other used CORBA communications to define an overlay network.

In this context, service localization and discovery algorithms are research issues close to lookup problems in Peer-to-peer networks. The Master Agent interconnection may be modeled by a graph where each vertex is a peer. Efficient traversal algorithms need to be found to avoid network flooding. Depending on the network size, several classes of algorithms may fit our needs. For small sized networks, standard broadcasting algorithms will be efficient enough when the number of clients is small. When the size of the network grows, standard broadcasting algorithms will lead to bottlenecks and Master Agent overloading. To study the influence of these parameters on the lookup algorithm performances, we simulate the interconnection network and its behavior using SimGrid. Our results shows that the network is the critical resource in the algorithm execution.

For large scale networks, we propose a data structure which allows to distribute spanning trees among the nodes and links of the interconnection network. This structure has been called Distributed Spanning Tree as it provides a different spanning tree for each Master Agent. It decreases bottlenecks in the graph traversal algorithms. The structure proposed has been simulated and results shows its efficiency: it limits the number of messages to find a server and distributes the load over the interconnection network.

6.2.4. *A peer-to-peer extension for DIET.*

We have developed a peer-to-peer extension for DIET using JXTA that allows a dynamic connection of DIET components. JXTA provides functionalities such as passing through a firewall and similar network protections, or dynamically discovering other peers. These tools are mandatory to develop a Multi Master-Agents version of DIET using Peer-to-peer technology.

One of the current implementations of the Multi-MA has been developed with JXTA. This is a prototype of a future powerful Multi-MA version using clever algorithms for discovery. However, connecting Corba components to JXTA is not easy. We can consider that the current JXTA Multi-MA has two parts. In this extension, the client is written in Java. Once the client has received the reference of the server, it connects to it and thus uses JXTA. The JXTA Multi-MA has to launch and communicate with a C++ Master Agent. The same interface appears in the SeD communication process.

We also did an implementation of an asynchronous PIF algorithm used for resource discovery in peer-to-peer grids [57]. A dynamic version of the DIET middleware that connects small hierarchies together using JXTA has been developed, which is able to dynamically adapt its connections as the network performance evolves and as the number of requests increases. The use of JXTA and the asynchronous PIF algorithm allows a quick and efficient discovery of available servers. The propagation has been first implemented as an asynchronous star graph traversal algorithm (STAR_{async}) and then using the asynchronous PIF scheme (PIF_{async}).

Our experimental results show that the PIF_{async} algorithm has the same cost as the STAR_{async} algorithm when the network performance are homogeneous. Moreover, when the network traffic increases on some links of the target platform, our PIF_{async} algorithm outperforms the STAR_{async} one by choosing the less loaded links to build an optimal tree in the connection graph.

6.2.5. A Monitoring Software for DIET

LogService is a monitoring software for DIET. It centralizes system information collected on each Agent/SeD and offers them to concerned tools. LogService has three parts. The first part (LogComponent) deals with collecting log messages on the component side (e.g. DIET agents) and sending them to the monitor core (LogCentral). The second part (LogCentral) connects components and tools by offering APIs for both sides. It gathers and merges incoming messages and offers them to connected tools. The third part (LogTool) is on the tool side (e.g. VizDiet [5]) to deliver incoming log messages from LogCentral. In this distributed approach all log messages must be sorted, and thus all monitoring tools connected will receive ordered logs. A clock synchronization is done for each component (modification of messages timestamp). All logs have a tag field which indicates the log type. Tags can be defined using a configuration file on the LogCentral side.

6.2.6. Deployment for DIET: Software and Research

To deploy easily DIET we have designed GoDIET, a new tool for the hierarchical deployment of distributed DIET agents and servers. With the help of this tool we can launch without effort DIET and its services. We have studied experiments testing the performance of three approaches to handling inter-element dependencies in the launch process: usage of feedback from logservice to guide the launch, fixed sleep period between dependent elements, and an aggressive approach that uses feedback for all agents but launches all servers without waiting. Based on experimental results we conclude that using feedback is the most effective approach. In the current version of GoDIET, users must write a simple XML file to describe statically the platform. We are studying how to do that automatically.

We worked on an automatic deployment solution for DIET [76]. The first step was to study the homogeneous case. Thus we provide an approach to determine an appropriate hierarchical middleware deployment for a homogeneous resource platform of a given size. The approach determines how many nodes should be used and in what hierarchical organization; the goal is to maximize steady-state throughput. The model provides an optimal real-valued solution without resource constraints; we then apply round-up or round-down to obtain integer factors for the hierarchy definition. We also provide algorithms to modify the obtained hierarchy to limit the number of resources used to the available number. We instantiate the model for the hierarchical scheduling system used in DIET. Our experiments validated the throughput performance model used for servers and agents and demonstrated that the automatic deployments performed well as compared to other intuitive deployments. We plan to work on deployment planning and re-deployment algorithms for middleware on heterogeneous clusters and Grids.

[5] A graphic Java tools to visualize the current state of a DIET platform.

6.2.7. 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 then proposed an algorithm that computes at the same time the mapping of data and computational requests on these data [65]. Our motivation for this work comes from an application in life science and more precisely around the search of sites and signatures of proteins into databanks of protein sequences (GriPPS [105], see Section 4.6).

Our approach uses a good knowledge of databank usage scheme and of the target platform. Starting with this information, we have designed a linear program and a method to obtain a mixed solution, i.e., integer and rational numbers, of this program. With the OptorSim simulator, we have been able to compare the results of our algorithm to other approaches: a greedy algorithm for data mapping, and an on-line algorithm for the scheduling of requests.

We came to the conclusion that when the storage space available on the grid is not large enough to store all databanks that lead to very time consuming requests on all computation servers, then our approach increases the throughput of the platform.

6.2.8. Data Management

The DIET approach consists in selecting appropriate servers to solve computational requests on behalf of the clients connected to it. For scientific applications which use large data, the choice of the server does not only depend on the server efficiency. The cost of one request is composed of the data transfers to the chosen server (and the gathering of the results) and of the computation time. By avoiding useless data transfers we reduce the global execution time.

We have designed and implemented a data management service tightly coupled with the DIET hierarchy [28]. In this data management service, data are described by a handle created by the client. Using this handle, the client can choose or act on the management policy of his data: persistence, replication and preplacement.

The data persistence policy allows to store a data on a server where it has been transfered or generated. If this data is used in a further computation, the client does not have to upload it again. For instance, this is the case for intermediate data generated by one request and used by a following request of a same computation sequence. The replication policy allows to replicate data on several servers. Data replication is used when executing in parallel a sequence of requests which use the same data. The client does not manage the replica of a data. This is done by the data management service. However, no support is given for request synchronization and data coherency. These have to be done by a scheduler on the client side. The preplacement policy allows the client to upload a data on the platform before submitting a request. Thus, the client will benefit from the overlap of the current computation and the data transfer for the next request. This functionality intend to be used by a scheduler rather than directly by the client: data might then be moved between servers for scheduling purpose.

6.2.9. Parallel Job Submission Management

In order to launch parallel applications such as LAMMPS in a transparent manner for the user, several points have to be addressed. *DIET* servers must be able to launch MPI process on a cluster on top of the different MPI implementations. They must also integrate the corresponding mechanisms to communicate with Batch schedulers to submit the parallel jobs. Furthermore, when a job is submitted to a batch scheduler, its start time is computed as a function of the number of demanded resources, the job deadline, etc. Such information must be communicated to the agent hierarchy for scheduling reasons, and trade-off functions have to be implemented in the *DIET* servers, to decide the number of resources to use, for the job to start in a reasonable time and to finish accordingly.

There are no common semantics, nor standard options, among batch schedulers. A library called Elagi[6] has been proposed by the GRAIL Lab, which allows a user to remotely submit tasks to numerous batch schedulers.

[6] http://grail.sdsc.edu/projects/elagi/

First, we have improved the Elagi library so that it now takes into account OAR[7], the batch scheduler developed in Grenoble and deployed on Grid'5000. Second, Elagi has been linked to the *DIET* project. Third, the *DIET* system has been modified to render the parallel submission mechanisms.

Work in progress concerns the start of the parallel MPI tasks, which must deal with the different MPI implementations and must be performed transparently from the user point of view.

6.3. Parallel Direct Solvers for Sparse Systems of Linear Equations

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 of the software package MUMPS

Since MUMPS version 4.3, released in 2003, we have developed several functionalities that are now part of the latest public version, MUMPS 4.5, released in 2005. In particular, the work resulting of the PhDs of A. Guermouche and S. Pralet have been integrated in the software: hybrid schedulers that significantly improve the parallel performance (while respecting tighter memory constraints), and efficient solution of symmetric indefinite problems (preprocessing, static pivoting and 2x2 pivots). Other recent features available in MUMPS 4.5 include sparse and multiple right-hand sides, better numerical stability for partial factorizations, ability to return a 2D block cyclic Schur complement matrix, and various improvements resulting from the feedback and intense interaction with both academic and industrial users, to whom we have continuously provided a significant level of support.

We are currently working on Scilab and Matlab interfaces (for the sequential version of MUMPS), as well as on a preliminary out-of-core extension. Those two functionalities will be made available in a future release.

6.3.2. Hybrid scheduling strategies for the parallel multifrontal method

We have pursued the work on hybrid dynamic scheduling approaches in the context of parallel multifrontal methods, that respect optimistic memory scenarios obtained during the analysis. While this research work was started in 2004, it has been enhanced for large-scale matrices and the implementation is now part of MUMPS Version 4.5. Using this approach, we were able to factor a problem of 3.7 million unknowns, requiring 5 billion non-zeros entries in the factored matrix and 31 Tera operations, in 284 seconds on 64 Power 4 processors of the IBM machine at IDRIS. This work is in collaboration with P. Amestoy (ENSEEIHT-IRIT), as well as A. Guermouche and S. Pralet (both while at ENSEEIHT-IRIT).

6.3.3. A preliminary out-of-core extension of a parallel multifrontal solver

The memory usage of sparse direct solvers can be the bottleneck to solve large-scale problems. This is why we are working on a parallel out-of-core extension to parallel multifrontal methods. In a first approach, the factors are written to disk as soon as they are computed, while the active memory fully remains in-core. We have implemented both a synchronous and an asynchronous approach for the I/O and have observed that the cost of disk accesses could remain reasonable for the factorization step, while the solution step can be critical from a performance point of view. Furthermore, for large problems, writing only the factors to disk is not sufficient, and we currently work on simulations to better understand the limits of our approach and of parallel multifrontal methods in general; the goal is to identify the critical points to take into account when designing a more general out-of-core approach where the active memory is also out-of-core. This work is performed in the context of the PhD of Emmanuel Agullo, in collaboration with Abdou Guermouche (LABRI).

6.3.4. Experimentation on real-life test problems

MUMPS users provide us with new challenging problems to solve and constantly help us validating and improving our algorithms. For example, SAMTECH S.A. or BRGM have provided huge problems that we use to assess the performance and limits of our approaches. We have informal collaborations around MUMPS with a number of institutions: (i) industrial teams which experiment and validate our package, (ii) research teams

[7] http://oar.imag.fr/

with which we discuss new functionalities they would need, (iii) designers of finite element packages who integrate MUMPS as a solver for the linear systems arising, (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 algorithmic studies on large-scale industrial problems, either coming directly from MUMPS users, or from standard collections of sparse matrices now in the public domain (Rutherford-Boeing and PARASOL).

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

The GRID TLSE project (see [95]), coordinated by ENSEEIHT-IRIT, is developing an expertise site providing a one-stop shop for users of sparse linear algebra software. A user will be able to interrogate databases for information and references related to sparse linear algebra, and will also be able to 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 that a user will be able to experiment via GRID TLSE. We are involved in the software layers related to the use of the DIET middleware and for our expertise in sparse direct solvers.

7. Contracts and Grants with Industry

7.1. Contract with CERFACS/CNES, 2005

In order to satisfy some needs from CNES (Centre National d'Etudes Spatiales), we have signed a contract with CERFACS and ENSEEIHT-IRIT to study how an extension of the MUMPS library can be used for certain classes of problems in electromagnetics.

In a parallel distributed environment (required because of the size of the problems involved), MUMPS is one of the rare packages having the functionality of returning a Schur complement. This functionality is useful, for example, in the context of coupled simulation codes for wave propagations from antennas, developed by the *Computational Electromagnetics* project at CERFACS. However, the Schur complement is currently centralized on one of the processors. The objective of the contract was to study how to remove this bottleneck, allowing the Schur complement matrix to be built in parallel, directly distributed onto the processors. We have enhanced the MUMPS package in that purpose, while CERFACS has validated and used our work on real-life problems from CNES. The new resulting functionality has been made available in the last public release of MUMPS.

Jean-Yves L'Excellent participated to this contract.

7.2. Contract with SAMTECH, 2005-2006

INRIA (project team GRAAL) and ENSEEIHT-IRIT have signed a contract with the company SAMTECH S.A. (Belgium), who develop the European finite element software package SAMCEF. The goal is to study how a parallel sparse out-of-core approach can help solving problems from customers of SAMTECH, for which classical parallel direct methods require too much memory, even on high-end platforms, and where iterative platforms solvers fail to provide a correct solution.

The contract is 18 months long, and will rely on the use of the software package MUMPS. The new functionalities developed for this contract will be made available in a future public release of the package, that will then be used as an environment for our research on scheduling aspects in the context of out-of-core approaches.

In Lyon, Emmanuel Agullo and Jean-Yves L'Excellent participate to this contract.

8. Other Grants and Activities

8.1. Regional Projects

8.1.1. Fédération lyonnaise de calcul haute performance (Federation for high-performance computing in Lyon)

This project federates various local communities interested in high-performance and parallel and distributed computing. This project allows a good contact with people from various application fields, to whom we aim at providing advices or solutions related to either grid computing, parallel numerical solvers or the parallelization of scientific software. This project also gathers several hardware platforms that can be used as a local Grid.

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

8.1.2. Institut des Sciences et Technique de l'Information

J.-M. Nicod and L. Philippe are involved in ISTI (Regional Institute for Information Sciences and Technologies). J.-M. Nicod leads the "Automatic Detection and Correction Methods of Artefacs in Myocardium Tomography" project. The aim of this project is to correct medical images obtained by gamma-camera.

8.1.3. RAGTIME: Rhône-Alpes: Grille pour le Traitement d'Informations Médicales (2003-2006)

RAGTIME, a project of *Région Rhône-Alpes*, is devoted to the use of the grid to perform efficient distributed accesses and computations on medical data. It federates most of the local researchers on grid computing together with medical centers, hospitals, and industrial partners.

E. Caron and F. Vivien participate to this project.

8.1.4. 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) has started in September around shceduling problems for physics and bioinformatic applications.

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

8.2. National Contracts and Projects

8.2.1. Ministry Grant: ACI Grid Grid2, 3 years, 2002-2005

Y. Robert is a member of the ACI Grid "Grid2", a project whose aim is to promote scientific exchanges among researchers. He is leading one of the five topics of the project, entitled "Algorithm design and scheduling techniques".

8.2.2. Ministry Grant: ACI Grid TLSE, 3 years, 2002-2005

The project ACI GRID TLSE aims at setting up a Web expertise site for sparse matrices, including software and a database. Using the middleware developed by GRAAL and the sparse codes (including MUMPS) developed by various partners, this project will allow users to submit requests of expertise for the solution of sparse linear systems. For example a typical request could be "which sparse matrix reordering heuristic leads to the smallest number of operations for my matrix?", or "which software is most robust for my problems?"

The project partners also include ENSEEIHT-IRIT (coordinator, Toulouse), CERFACS (Toulouse) and LABRI (INRIA ScAlApplix project, Bordeaux).

E. Caron, F. Desprez, J.-Y. L'Excellent participate to this project.

8.2.3. INRIA new investigation Grant: ARC INRIA Otaphe, 2 years, 2005-2006

This project (*Ordonnancement de tâches parallélisables en milieu hétérogène*, coordinated by Frédéric Suter from the INRIA Algorille project) aims at designing new algorithms for the scheduling of data-parallel tasks on heterogeneous platforms.

Y. Caniou, E. Caron, and F. Desprez participate to this project.

8.2.4. INRIA new investigation Grant: ARC INRIA Georep, 2 years, 2005-2006

This project (Geometrical Representations for Computer Graphics, coordinated by Bruno Levy from the INRIA ISA-ALICE project) aims at designing new solutions to convert a raw representation of a 3D object into a higher-level representation. In this context, our participation consists in providing expertise and support for the underlying numerical problems involved (sparse systems of equations, use of our sparse direct solver MUMPS).

A. Fèvre and J.-Y. L'Excellent participate to this project.

8.2.5. INRIA Grant: Software development for MUMPS

INRIA is financing Aurélia Fèvre, who has been recruited on September 1st, 2005, as an engineer to work on the development of the MUMPS software package. While helping with various aspects of the software issues and maintenance, she has more specifically been developing a Scilab interface to the sequential version of MUMPS, allowing the use of an efficient solver for sparse systems of linear equations from within Scilab. The Scilab interface will be made available in the next release of MUMPS.

8.2.6. Ministry Grant: ACI Grandes masses de données GridExplorer, 2003-2005

The aim of this project is to create a computational grid emulator. We are interested in the validation of DIET by this emulator. Especially, we study several techniques of deployment and of hierarchical and distributed scheduling.

E. Caron and F. Desprez participate to this project.

8.2.7. Ministry Grant: ACI Grandes masses de données Grid Data Service, 2003-2005

The main goal of this project is to specify, design, implement, and evaluate a data sharing service for mutable data and integrate it into DIET. This service is built using the generic JuxMem[8] platform for peer-to-peer data management. The platform will serve to implement and compare multiple replication and data consistency strategies defined together by the PARIS team (IRISA) and by the REGAL team (LIP6).

E. Caron and F. Desprez participate to this project.

8.2.8. French ministry of research grant: GRID5000, 3 years, 2004-2007

ENS Lyon is involved in the GRID'5000 project, which aims at building an experimental Grid platform gathering eight sites geographically distributed in France. 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 will be evaluated on this platform as well as several scheduling heuristics.

8.2.9. ANR grant: ALPAGE (ALgorithmique des Plates-formes A Grande Echelle)

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.

Y. Robert is leading the Rhône-Alpes site of this project, which comprises three sites: Paris (LIX and LRI laboratories) and Bordeaux-Rennes (Paris and Scalapplix projects). F. Vivien also participates in this project.

[8] http://www.irisa.fr/paris/Juxmem/welcome.htm

8.3. International Contracts and Projects

8.3.1. INRIA Associated Team I-Arthur

In 2003, we obtained a grant from INRIA to set an associated team with the Grid Research And Innovation Laboratory (GRAIL) of the University of California, San Diego. Our aim is to work on scheduling for heterogeneous and Grid platforms in collaboration with researchers from GRAIL. We had several exchanges of researchers and students from both sides during the last 3 years and we organized a workshop in 2004 (Aussois), and one in 2005 (San Diego). The DIET software from GRAAL was used to validate some of the scheduling heuristics and SimGrid2 was improved and used to simulate our platform.

The final evaluation of the project was presented at INRIA in November 2005.

Workshop *Scheduling for large-scale heterogeneous platforms*

This workshop was organized on November 12-14, 2005, in San Diego. The topic was "scheduling for large-scale heterogeneous platforms". The organizers were Larry Carter (University of California at San Diego), Henri Casanova (University of Hawaï at Manoa), and Jeanne Ferrante (University of California at San Diego), Frédéric Desprez and Yves Robert.

Here is the list of the talks:

Session I

- A. Rosenberg: A Pebble Game for Internet-Based Computing
- A. Legrand: Scheduling Competing Regular Applications on a Heterogeneous Master-Worker Platform
- F. Vivien: Off- and On-Line Scheduling of Divisible Requests

Session II

- J. Weissman: Scheduling Challenges in Hosting Services on the Public Grid
- G. Fedak: Scheduling Independent Tasks Sharing Large Data Distributed with BitTorrent
- F. Cappello: Status of Grid'5000 development: a 5000 CPU, nationwide experimental Grid

Session III

- E. Jeannot: Messages Scheduling for Data Redistributions
- M. den Burger: Balanced Multicasting: High-throughput Communication for Grid Applications

Session IV

- L. Eyraud: Scheduling with Reservations
- C. Bailey Lee: Investigating Inaccuracy in User Runtime Estimates and "The Padding Hypothesis"
- Z. Shi: Scheduling Workflow Applications on Processors with Different Capabilities

Session V

- S. Nandy: Efficiently Programming Decentralized Systems
- C. Varela: Towards an Internet Operating System: Middleware for Adaptive Distributed Computation
- E. Caron: Automatic Middleware Deployment Planning on Clusters

8.3.2. NSF-INRIA, The University of Tennessee, Knoxville, USA

F. Desprez is the French coordinator of an NSF-INRIA project entitled "Environments and Tools for Grid-enabled Scientific Computing". The project is conducted with the Innovative Computing Laboratory from the University of Tennessee (J. Dongarra) and the ALGORILLE project from LORIA (with E. Jeannot).

8.3.3. NSF-INRIA, University of California at San Diego, USA

Y. Robert is the French coordinator of an NSF-INRIA project entitled "Algorithms and simulations for scheduling on large-scale distributed platforms". The project is conducted with the Computer Science Department of the University of California at San Diego (L. Carter, H. Casanova, and J. Ferrante).

8.3.4. STAR Project KISTI-INRIA

Together with the PARIS Project-Team located at INRIA/IRISA, the GRAAL Project has been selected by the STAR program of the French Embassy in Seoul to conduct a 2-year cooperation with the Department of Aerospace Engineering (Prof. Seung Jo Kim) of the Seoul National University. This cooperation, which started in June 2003, aims at experimenting a Grid infrastructure, made with the computing equipments of the two participants, with aerospace applications (SNU) and middleware and programming tools designed by INRIA. Four researchers from the two INRIA project-teams visited SNU in December 2004 to give talks and work on the gridification of an aerospace application on the DIET software.

E. Caron and F. Desprez participate to this project.

9. Dissemination

9.1. Scientific Missions

CoreGrid: CNRS is a partner of the CoreGrid network of excellence. The CNRS partnership involves Algorille in Nancy (E. Jeannot), ID-Imag in Grenoble (G. Huard, D. Trystram) and the Graal project (A. Benoit, E. Caron, F. Desprez, Y. Robert, F. Vivien). Yves Robert has been leading the CNRS contribution until September 15. Since then, Frédéric Vivien assumed that role. Frédéric Vivien is also responsible for two tasks in the scheduling workpackage.

Austrian Science Fund (FWF): F. Desprez has evaluated a proposal for a Grid project for the Austrian Science Fund.

Anticipating Scientific and Technological Needs: Basic Research (NEST): F. Desprez has evaluated one Grid proposal for the European Commission (NEST).

Science Foundation Ireland: F. Desprez and Y. Robert have evaluated two projects for the Basic Research Grant Programme of Science Foundation Ireland.

Netherlands Organisation for Scientific Research (NWO): F. Vivien has evaluated a project for the Inovational Research Incentives Scheme - Veni programme of the Netherlands Organisation for Scientific Research.

9.2. Animation Responsibilities

Jean-Yves L'Excellent is a member of the ERCIM working group "Application of numerical mathematics in science".

9.3. Edition and Program Committees

Anne Benoit is co-organizing the Third International Workshop on Practical Aspects of High-level Parallel Programming (PAPP 2006), University of Reading, UK, May 2006.
A. Benoit was a member of the program committee of ICCS 2005, and she is a member of the program committee of ICCS 2006 and CMPP 2006.

Eddy Caron will be a member of the program committee of HCW 06 (Heterogeneous Computing Workshop), Rhodes Island, Greece, April 25, 2006, to be held in conjunction with IPDPS 2006.

Frédéric Desprez is an associate editor of *Parallel and Distributed Computing Practices* and *Computing Letters* (COMPULETT).
F. Desprez participated to the program committees of ICCS2005, ICCSA 2005, PGaMS'05 (held in conjuction with ICCS 2005), CLADE'05 (held in conjunction with HPDC), LaSCoG'05 (held inconjunction with PPAM 2005), the 12th European PVM/MPI Users' Group Meeting, Grid 2005 (held in conjunction with SC05). He is a member of the EuroPar Advisory board and of the editorial board of "Scalable Computing: Practice and Experience" (SCPE).
F. Desprez was vice-chair at the HiPC'04 conference in Bangalore (algorithm track). He also gave a keynote talk at this conference.

Jean-Yves L'Excellent was a member of the organizing/scientific committee of the Second International Workshop on Combinatorial Scientific Computing (CSC05), Toulouse, France, June 2005. He is a member of the program committee of ICPADS'2006 (IEEE International Conference on Parallel and Distributed Systems), Minneapolis, USA.

Yves Robert is an associate editor of *IEEE Transactions on Parallel and Distributed Systems*. He is a member of the editorial board of the *International Journal of High Performance Computing Applications* (Sage Press).
Y. Robert participated to the following program committees: EuroPDP'05 (European Symposium on Parallel and Distributed Processing), Lugano, Switzerland, and EuroPDP'06, Montbéliard, France; Euro PVM-MPI 2004, Bupadest, Hungary, and Euro PVM-MPI 2005, Sorrento, Italy; HeteroPar'04, Cork, Ireland and HeteroPar'05, Boston, USA (2005).
Y. Robert was vice-chair (topic Applications) of the program committee of IPDPS'05 (IEEE International Parallel and Distributed Processing Symposium), Denver, USA. Y. Robert was general chair of the HCW'2005 workshop (IEEE Heterogeneous Computing Workshop), Denver, USA, and is now a member of the Steering Committee of the conference. Y. Robert will be program chair of HiPC'2006 (IEEE Int. Conf. on High Performance Computing), Bangalore, India. Y. Robert will be vice-chair (topic Algorithms) of the program committee of ICPADS'2006 (IEEE International Conference on Parallel and Distributed Systems), Minneapolis, USA.
Y. Robert will give an invited talk ar IPDPS'2006 in Rhodes, Greece.
Y. Robert has been elected an IEEE Fellow (promotion 2006).
In addition to the special issue of IJHPCA following the Aussois workshop, we are editing two special issues of leading scientific journals:

- Special issue of Parallel Computing on *Heterogeneous computing*, edited by A. Kalinov, A. Lastovetsky, and Y. Robert, which appeared in 2005.
- Special issue of IEEE Trans. Parallel Distributed Systems on *Algorithm design and scheduling techniques (realistic platform models) for heterogeneous clusters*, edited by H. Casanova, Y. Robert, and H.J. Siegel, to appear in February 2006.

Frédéric Vivien is an associate editor of *Parallel Computing* since August 1, 2005.
F. Vivien was a member of the program committee of ISPA'05 (International Symposium on Parallel and Distributed Processing and Applications), Nanjing, China, November 2-5, 2005, of the Workshop on scheduling for parallel computing, Poznan, Poland, September 13-16, 2005, held in conjunction with PPAM 2005, and of Grid 2005 (6th IEEE/ACM International Workshop on Grid Computing), Seattle, Washington, USA, November 14, 2005, held in conjunction with Supercomputing 05.
He will be a member of the program committee of HCW 06 (Heterogeneous Computing Workshop), Rhodes Island, Greece, April 25, 2006 to be held in conjunction with IPDPS 2006; and of HiPC 2006 (13th International Conference on High Performance Computing), Bangalore, India, December 2006.

Laurent Philippe was a member of the International Conference on Distributed Frameworks for Multimedia Applications (DFMA'05) program committee. Laurent Philippe is member of the program committee of CFSE, the French ACM Conference on Operating Systems.
L. Philippe is a member of the program committee of *Journées des composants*, French workshop on Components models and applications.

9.4. Administrative and Teaching Responsibilities

9.4.1. Administrative Responsibilities

Competitive selection for ENS Lyon students. Y. Robert was responsible of the computer science test which is part of the written examination in the competitive selection of the students of the École normale supérieure de Lyon.
F. Vivien is co-responsible of the theoretical test part of the oral examination in the competitive selection of the students of the three Écoles normales supérieures (Cachan, Lyon, and Paris).

Université de Franche Comté. L. Philippe is the head of the Master in Computer Science of Université de Franche-Comté.

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

9.4.2. 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 vice-head of the École Doctorale *Mathématiques et Informatique Fondamentale*.

Master in Computer Science at Université de Franche Comté. Bachelor degree (Maîtrise) in computer science, L. Philippe has been responsible of the Client/Server and distributed programming lecture since 2000.
Bachelor degree (Maîtrise) in computer science, J.-M. Nicod has been responsible for the Graphs Algorithms lecture since 2002.
Master degree in computer science, L. Philippe is responsible of the Distributed Systems Engineering and the Engineering for distributed applications lectures.
Master degree in computer science (Maîtrise), J.-M. Nicod is responsible of the Distributed Algorithms and Graphs and Optimizations lectures.

10. Bibliography

Major publications by the team in recent years

[1] 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.

[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, V. BOUDET, A. PETITET, F. RASTELLO, Y. ROBERT. *A proposal for a heterogeneous cluster ScaLAPACK (dense linear solvers)*, in "IEEE Trans. Computers", vol. 50, n^o 10, 2001, p. 1052-1070.

[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", to appear, 2005.

[5] E. CARON, G. UTARD. *On the Performance of Parallel Factorization of Out-of-Core Matrices*, in "Parallel Computing", vol. 30, n^o 3, February 2004, p. 357-375.

[6] 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.

[7] 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.

[8] A. GUERMOUCHE, J.-Y. L'EXCELLENT, G. UTARD. *Impact of reordering on the Memory of a Multifrontal Solver*, in "Parallel Computing", vol. 29, n^o 9, 2003, p. 1191–1218.

[9] 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.

Books and Monographs

[10] L. CARTER, H. CASANOVA, F. DESPREZ, J. FERRANTE, Y. ROBERT (editors). *Special issue on Scheduling techniques for large-scale distributed platforms*, to appear, Int. J. High Performance Computing Applications 20, 1, 2006.

[11] H. CASANOVA, Y. ROBERT, H. SIEGEL (editors). *Special issue on Algorithm design and scheduling techniques (realistic platform models) for heterogeneous clusters*, to appear, IEEE Trans. Parallel Distributed Systems 17, 2, 2006.

[12] A. KALINOV, A. LASTOVETSKY, Y. ROBERT (editors). *Special issue on Heterogeneous computing*, Parallel Computing 31, 2005.

Doctoral dissertations and Habilitation theses

[13] S. DAHAN. *Mécanismes de recherche de services extensibles pour les environnements de grilles de calcul*, PhD Thesis, Université de Franche Comté, 2005.

[14] B. DEL-FABBRO. *Contribution à la gestion des données dans les grilles de calcul à la demande : de la conception à la normalisation.*, PhD Thesis, Université de Franche Comté, 2005.

[15] H. RENARD. *Équilibrage de charge et redistribution de données sur plates-formes hétérogènes*, Ph. D. Thesis, École normale supérieure de Lyon, December 2005.

Articles in refereed journals and book chapters

[16] P. R. AMESTOY, A. GUERMOUCHE, J.-Y. L'EXCELLENT, S. PRALET. *Hybrid scheduling for the parallel solution of linear systems*, in "Parallel Computing", 2005.

[17] G. ANTONIU, M. BERTIER, L. BOUGÉ, E. CARON, F. DESPREZ, M. JAN, S. MONNET, P. SENS. *Future Generation Grids*, CoreGrid Series, vol. 2, chap. GDS: An Architecture Proposal for a Grid Data-Sharing Service, Springer Verlag, 2005.

[18] 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° 3, 2005, p. 207-218.

[19] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Pipelining broadcasts on heterogeneous platforms*, in "IEEE Trans. Parallel Distributed Systems", 2005.

[20] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Steady-state scheduling on heterogeneous clusters*, in "Int. J. of Foundations of Computer Science", vol. 16, n° 2, 2005.

[21] O. BEAUMONT, L. MARCHAL, Y. ROBERT. *Complexity results for collective communications on heterogeneous platforms*, in "Int. Journal of High Performance Computing Applications", to appear, 2006.

[22] A. BENOIT, M. COLE, S. GILMORE, J. HILLSTON. *Evaluating the performance of pipeline-structured parallel programs with skeletons and process algebra*, in "Scalable Computing: Practice and Experience", vol. 6, n° 4, December 2005, p. 1-16.

[23] A. BENOIT, M. COLE, S. GILMORE, J. HILLSTON. *Scheduling skeleton-based grid applications using PEPA and NWS*, in "The Computer Journal, Special issue on Grid Performability Modelling and Measurement", vol. 48, n° 3, 2005, p. 369–378.

[24] A. BENOIT, B. PLATEAU, W. J. STEWART. *Memory Efficient Kronecker algorithms with applications to the modelling of parallel systems*, in "Future Generation Computer Systems, special issue on System Performance Analysis and Evaluation", 2005.

[25] A. BENOIT, B. PLATEAU, W. J. STEWART. *Réseaux d'automates stochastiques à temps discret*, in "Revue des

sciences et technologies de l'information, Technique et science informatiques, "Evaluation de Performances"", vol. 24, n° 2-3, 2005, p. 229–248.

[26] F. BERMAN, H. CASANOVA, A. CHIEN, K. COOPER, H. DAIL, A. DASGUPTA, W. DENG, J. DONGARRA, L. JOHNSSON, K. KENNEDY, C. KOELBEL, B. LIU, X. LIU, A. MANDAL, G. MARIN, M. MAZINA, J. MELLOR-CRUMMEY, C. MENDES, A. OLUGBILE, M. PATEL, D. REED, Z. SHI, O. SIEVERT, H. XIA, A. YARKHAN. *New Grid Scheduling and Rescheduling Methods in the GrADS Project*, in "International Journal of Parallel Programming", vol. 33, n° 2-3, June 2005, p. 209–229.

[27] Y. CANIOU, E. JEANNOT. *Multi-Criteria Scheduling Heuristics for GridRPC Systems*, in "Special edition of The International Journal of High Performance Computing Applications (IJHPCA)", 2005.

[28] E. CARON, B. DEL-FABBRO, F. DESPREZ, E. JEANNOT, J.-M. NICOD. *Managing Data Persistence in Network Enabled Servers*, in "Scientific Programming Journal", to appear in Special Issue on Dynamic Grids and Worldwide Computing, 2005.

[29] E. CARON, F. DESPREZ. *DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid*, in "International Journal of High Performance Computing Applications", 2005.

[30] E. CARON, F. DESPREZ, M. DAYDÉ, A. HURAULT, M. PANTEL. *On Deploying Scientific Software within the Grid-TLSE Project*, in "Computing Letters (CoLe)", vol. 1, n° 3, 2005, p. 1-5.

[31] E. CARON, F. DESPREZ, J.-Y. L'EXCELLENT, C. HAMERLING, M. PANTEL, C. PUGLISI-AMESTOY. *Future Generation Grids*, CoreGrid Series, vol. 2, chap. Use of A Network Enabled Server System for a Sparse Linear Algebra Application, Springer Verlag, 2005.

[32] E. CARON, F. DESPREZ, F. SUTER. *Overlapping Communications and Computations with I/O in Wavefront Algorithms*, in "Concurrency & Computation: Practice & Experience", 2005.

[33] H. DAIL, F. DESPREZ. *Experiences with Hierarchical Request Flow Management for Network-Enabled Server Environments*, in "International Journal of High Performance Computing Applications", to appear, vol. 20, n° 1, February 2006.

[34] F. DESPREZ, A. VERNOIS. *Simultaneous Scheduling of Replication and Computation for Data-Intensive Applications on the Grid*, in "Journal of Grid Computing", to appear, 2006.

[35] A. GIERSCH, Y. ROBERT, F. VIVIEN. *Scheduling tasks sharing files on heterogeneous master-slave platforms*, in "Journal of Systems Architecture", 2005.

[36] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Constructing Memory-minimizing Schedules for Multifrontal Methods*, in "ACM Transactions on Mathematical Software", 2005.

[37] A. LEGRAND, L. MARCHAL, Y. ROBERT. *Optimizing the steady-state throughput of scatter and reduce operations on heterogeneous platforms*, in "J. Parallel and Distributed Computing", vol. 65, n° 12, 2005, p. 1497-1514.

[38] L. MARCHAL, Y. YANG, H. CASANOVA, Y. ROBERT. *Steady-state scheduling of multiple divisible load applications on wide-area distributed computing platforms*, in "Int. Journal of High Performance Computing Applications", to appear, 2006.

[39] H. RENARD, Y. ROBERT, F. VIVIEN. *Data redistribution algorithms for heterogeneous processor rings*, in "Int. Journal of High Performance Computing Applications", to appear, 2006.

Publications in Conferences and Workshops

[40] A. BALLIER, E. CARON, D. EPEMA, H. MOHAMED. *Simulating Grid Schedulers with Deadlines and Co-Allocation*, in "CoreGRID integration workshop, Pisa, Italy", Network of Excellence CoreGRID, November 2005.

[41] O. BEAUMONT, V. BOUDET, P.-F. DUTOT, A. LEGRAND, Y. ROBERT. *Gestion des ressources*, in "Informatique répartie", Hermes Sciences, 2005.

[42] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Centralized versus distributed schedulers for multiple bag-of-task applications*, in "International Parallel and Distributed Processing Symposium IPDPS'2006", to appear, IEEE Computer Society Press, 2006.

[43] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Independent and divisible tasks scheduling on heterogeneous star-schaped platforms with limited memory*, in "PDP'2005, 13th Euromicro Workshop on Parallel, Distributed and Network-based Processing", IEEE Computer Society Press, 2005, p. 179-186.

[44] O. BEAUMONT, L. MARCHAL, V. REHN, Y. ROBERT. *FIFO scheduling of divisible loads with return messages under the one-port model*, in "HCW'2006, the 15th Heterogeneous Computing Workshop", to appear, IEEE Computer Society Press, 2006.

[45] O. BEAUMONT, L. MARCHAL, Y. ROBERT. *Broadcast trees for heterogeneous platforms*, in "International Parallel and Distributed Processing Symposium IPDPS'2005", IEEE Computer Society Press, 2005.

[46] O. BEAUMONT, L. MARCHAL, Y. ROBERT. *Scheduling divisible loads with return messages on heterogeneous master-worker platforms*, in "International Conference on High Performance Computing HiPC'2005", LNCS, Springer Verlag, 2005.

[47] A. BENOIT, M. ALDINUCCI. *Automatic mapping of ASSIST applications using process algrebra*, in "High Level Parallel Programming HLPP 2005, Warwick University, Coventry, United Kingdom", July 2005.

[48] A. BENOIT, M. ALDINUCCI. *Towards the Automatic Mapping of ASSIST Applications for the Grid*, in "Proceedings of CoreGRID Integration Workshop, University of Pisa, Italy", November 2005.

[49] A. BENOIT, M. COLE. *Two fundamental concepts in skeletal parallel programming*, in "The International Conference on Computational Science (ICCS 2005), Part II", V. SUNDERAM, D. VAN ALBADA, P. SLOOT, J. DONGARRA (editors). , LNCS 3515, Springer Verlag, 2005, p. 764–771.

[50] A. BENOIT, M. COLE, S. GILMORE, J. HILLSTON. *Analyse quantitative de programmes applicatifs à base de squelettes algorithmiques*, in "Seizièmes journée francophones des langages applicatifs JFLA 2005", March

2005.

[51] A. BENOIT, M. COLE, S. GILMORE, J. HILLSTON. *Enhancing the effective utilisation of Grid clusters by exploiting on-line performability analysis*, in "CCGrid workshop "1st International Workshop on Grid Performability", Cardiff, Wales", May 2005.

[52] A. BENOIT, M. COLE, S. GILMORE, J. HILLSTON. *Flexible Skeletal Programming with eSkel*, in "11th International Euro-Par Conference, Lisbon, Portugal", LNCS Volume 3648, Springer Verlag, August 2005, p. 761–770.

[53] A. BENOIT, M. COLE, S. GILMORE, J. HILLSTON. *Using eSkel to implement the multiple baseline stereo application*, in "ParCo 2005, Malaga, Spain", September 2005.

[54] C. BLANCHET, F. DESPREZ, A. VERNOIS. *Simultaneous Scheduling of Replication and Computation for Bioinformatic Applications on the Grid*, in "Biological and Medical Data Analysis, Proceedings of 6th International Symposium, ISBMDA 2005, Aveiro, Portugal", vol. 3745 / 2005, Springer-Verlag GmbH, November 2005.

[55] Y. CANIOU, E. JEANNOT. *Le HTM : un module de prédiction de performances non-intrusif pour l'ordonnancement de tâches sur plate-forme de métacomputing*, in "16èmes Rencontres Francophones du Parallélisme - RENPAR'05, Le Croisic, Presqu'île de Guérande", April 2005.

[56] F. CAPPELLO, E. CARON, M. DAYDÉ, F. DESPREZ, E. JEANNOT, Y. JEGOU, S. LANTERI, J. LEDUC, N. MELAB, G. MORNET, R. NAMYST, P. PRIMET, O. RICHARD. *Grid'5000: a large scale, reconfigurable, controlable and monitorable Grid platform*, in "Grid'2005 Workshop, Seattle, USA", IEEE/ACM, November 13-14 2005.

[57] E. CARON, F. DESPREZ, F. PETIT, C. TEDESCHI. *A Peer-to-Peer Extension of Network-Enabled Server Systems*, in "e-Science 2005. First IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia", 5-8 December 2005, p. 430-437.

[58] E. CARON, F. DESPREZ, F. SUTER. *Out-of-Core and Pipeline Techniques for Wavefront Algorithms*, in "Proceedings of the 19th International Parallel and Distributed Processing Symposium (IPDPS'05), Denver - Colorado", 3-8April 2005.

[59] E. CARON, V. GARONNE, A. TSAREGORODTSEV. *A study of meta-scheduling architectures for high throughput computing: Pull vs. Push*, in "ISPDC 2005. The 4th International Symposium on Parallel and Distributed Computing, Lille. France", University of Lille 1, July 2005.

[60] E. CARON, V. GARONNE, A. TSAREGORODTSEV. *Étude d'architectures de méta-ordonnancement pour le calcul intensif en régime permanent et saturé*, in "RenPar 2005. 16ème Rencontres Francophones du Parallélisme, Croisic", 5-8April 2005.

[61] S. DAHAN. *Distributed Spanning Tree Algorithms for Large Scale Traversals*, in "11th International Conference on Parallel and Distributed Systems, ICPADS 2005, Fukuoka, Japan", IEEE Press, July 2005.

[62] S. DAHAN, J.-M. NICOD, L. PHILIPPE. *The Distributed Spanning Tree: A Scalable Interconnection Topology*

for Efficient and Equitable Traversal, in "5th Int. Symposium on Cluster Computing and the Grid, CCGRID05, Cardiff, UK", CD-ROM, 8 pages, IEEE Press, May 2005.

[63] H. DAIL, F. DESPREZ. *Adaptive window scheduling for a hierarchical agent system*, in "The Fourth International Symposium on Parallel and Distributed Computing (ISPDC 2005), Lille, France", IEEE Computer Society, July 2005.

[64] B. DEL-FABBRO, D. LAIYMANI, J.-M. NICOD, L. PHILIPPE. *Data Management in Grid Applications Providers*, in "Procs of the 1st Int. Conf. on Distributed Frameworks for Multimedia Applications, DFMA'2005, Besançon, France", February 2005, p. 315–322.

[65] F. DESPREZ, A. VERNOIS. *Simultaneous Scheduling of Replication and Computation on the Grid*, in "CLADE 2005, Research Triangle Park, NC", IEEE Computer Society Press, July 2005.

[66] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *A Study of Various Load Information Exchange Mechanisms for a Distributed Application using Dynamic Scheduling*, in "19th International Parallel and Distributed Processing Symposium (IPDPS'05)", 2005.

[67] A. LEGRAND, A. SU, F. VIVIEN. *Off-line scheduling of divisible requests on an heterogeneous collection of databanks*, in "Proceedings of the 14th Heterogeneous Computing Workshop, Denver, Colorado, USA", IEEE Computer Society Press, April 2005, (10 pages).

[68] L. MARCHAL, Y. YANG, H. CASANOVA, Y. ROBERT. *A realistic network/application model for scheduling divisible loads on large-scale platforms*, in "International Parallel and Distributed Processing Symposium IPDPS'2005", IEEE Computer Society Press, 2005.

[69] J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Off-line and on-line scheduling on heterogeneous master-slave platforms*, in "PDP'2006, 14th Euromicro Workshop on Parallel, Distributed and Network-based Processing", to appear, IEEE Computer Society Press, 2006.

[70] J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *The impact of heterogeneity on master-slave on-line scheduling*, in "HCW'2006, the 15th Heterogeneous Computing Workshop", to appear, IEEE Computer Society Press, 2006.

[71] H. RENARD. *Algorithmes de redistribution de données pour anneaux de processeurs hétérogènes*, in "16e Rencontres Francophones du Parallélisme des Architectures et des Systèmes, Le Croisic, France", April6-8 2005.

[72] A. VERNOIS. *Ordonnancement conjoint du placement des données et des calculs sur la grille*, in "16e Rencontres Francophones en Parallélisme, Le Croisic, France", April 2005.

Internal Reports

[73] G. ANTONIU, M. BERTIER, L. BOUGÉ, E. CARON, F. DESPREZ, M. JAN, S. MONNET, P. SENS. *GDS: an Architecture Proposal for a Grid Data-Sharing Service*, Also available as LIP Research Report 2005-28, Technical report, n° RR-5593, Institut National de Recherche en Informatique et en Automatique (INRIA), June 2005, http://www.inria.fr/rrrt/rr-5593.html.

[74] A. BENOIT, M. ALDINUCCI. *Automatic mapping of ASSIST applications using process algrebra*, Technical report, n° TR-0016, CoreGRID, October 2005.

[75] E. CARON, P. K. CHOUHAN, H. DAIL. *Automatic Middleware Deployment Planning on Clusters*, Also available as LIP Research Report 2005-26, Technical report, n° RR-5573, Institut National de Recherche en Informatique et en Automatique (INRIA), May 2005, http://www.inria.fr/rrrt/rr-5573.html.

[76] E. CARON, P. K. CHOUHAN, H. DAIL, F. VIVIEN. *Automatic Middleware Deployment Planning on Clusters*, Revised version of LIP Research Report 2005-26, Research report, n° 2005-50, Laboratoire de l'Informatique du Parallélisme (LIP), October 2005, http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2005/RR2005-50.pdf.

[77] E. CARON, H. DAIL. *GoDIET: a tool for managing distributed hierarchies of DIET agents and servers.*, Also available as LIP Research Report 2005-06, Research report, n° RR-5520, Institut National de Recherche en Informatique et en Automatique (INRIA), March 2005, http://www.inria.fr/rrrt/rr-5520.html.

[78] E. CARON, F. DESPREZ. *DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid*, Also available as LIP Research Report 2005-23, Technical report, n° RR-5601, Institut National de Recherche en Informatique et en Automatique (INRIA), June 2005, http://www.inria.fr/rrrt/rr-5601.html.

[79] E. CARON, F. DESPREZ, C. HAMERLING, J.-Y. L'EXCELLENT, M. PANTEL, C. PUGLISI-AMESTOY. *Use of A Network Enabled Server System for a Sparse Linear Algebra Grid Application*, Also available as LIP Research Report 2005-30, Technical report, n° RR-5595, Institut National de Recherche en Informatique et en Automatique (INRIA), June 2005, http://www.inria.fr/rrrt/rr-5595.html.

[80] E. CARON, F. DESPREZ, C. TEDESCHI. *Service discovery in a peer-to-peer environments for computational grids*, Also available as LIP Research Report 2005-42, Technical report, n° RR-5679, Institut National de Recherche en Informatique et en Automatique (INRIA), September 2005, http://www.inria.fr/rrrt/rr-5679.html.

[81] E. CARON, V. GARONNE, A. TSAREGORODTSEV. *A study of meta-scheduling architectures for high throughput computing.*, Technical report, n° 2005-13, Laboratoire de l'Informatique du Parallélisme (LIP), May 2005, http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2005/RR2005-13.ps.gz.

[82] E. CARON, V. GARONNE, A. TSAREGORODTSEV. *Evaluation of Meta-scheduler Architectures and Task Assignment Policies for High Throughput Computing.*, Also available as LIP Research Report 2005-27, Technical report, n° RR-5576, Institut National de Recherche en Informatique et en Automatique (INRIA), May 2005, http://www.inria.fr/rrrt/rr-5576.html.

[83] H. DAIL, F. DESPREZ. *Experiences with hierarchical request flow management for Network-Enabled Server Environments*, Research report, n° RR-2005-07, Laboratoire de l'Informatique du Parallélisme (LIP), February 2005, http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2005/RR2005-07.pdf.

[84] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *A Study of Various Load Information Exchange Mechanisms for a Distributed Application using Dynamic Scheduling*, Also available as LIP report RR2005-02 and as an ENSEEIHT-IRIT Technical Report., Research report, n° RR-5478, INRIA, 2005, http://www.inria.fr/rrrt/rr-5478.html.

[85] A. LEGRAND, A. SU, F. VIVIEN. *Minimizing the stretch when scheduling flows of biological requests*, Also available as INRIA research report 5724., Technical report, n° LIP 2005-48, Laboratoire de l'informatique de parallélisme (LIP), École normale supérieure de Lyon, France, October 2005, http://www.inria.fr/rrrt/rr-5724.html.

[86] J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *The impact of heterogeneity on master-slave on-line scheduling*, Also available as INRIA research report 5732., Research report, n° LIP 2005-51, Laboratoire de l'informatique de parallélisme (LIP), École normale supérieure de Lyon, France, October 2005, http://www.inria.fr/rrrt/rr-5732.html.

Miscellaneous

[87] E. AGULLO. *Résolution out-of-core de systèmes linéaires creux de grande taille*, Rapport de DEA, École Normale Supérieure de Lyon, 2005.

[88] I. DJAMA. *Evaluation de stratégies de recherche de services à Grande échelle*, Technical report, Université de Franche-Comté, 2005.

[89] J.-F. PINEAU. *Ordonnancement de tâches indépendantes sur plate-forme maître-esclave hétérogène: modèles hors-ligne et à la volée*, Rapport de DEA, École Normale Supérieure de Lyon, 2005.

[90] C. TEDESCHI. *Découverte de services dans un environnement pair-à-pair pour les grilles de calcul*, Rapport de DEA, École Normale Supérieure de Lyon, 2005.

Bibliography in notes

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

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

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

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

[95] *GRID TLSE*, http://www.enseeiht.fr/lima/tlse.

[96] 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° 1, 2001, p. 15-41.

[97] 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.

[98] 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, nº CS-01-467, University of Tennessee, Knoxville, TN, July 2001, http://www.cs.utk.edu/netsolve/.

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

[100] 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.

[101] 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.

[102] H. EL-REWINI, H. H. ALI, T. G. LEWIS. *Task Scheduling in Multiprocessing Systems*, in "Computer", vol. 28, nº 12, 1995, p. 27–37.

[103] 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.

[104] M. FERRIS, M. MESNIER, J. MORI. *NEOS and Condor: Solving Optimization Problems Over the Internet*, in "ACM Transactions on Mathematical Sofware", vol. 26, nº 1, 2000, p. 1-18, http://www-unix.mcs.anl.gov/metaneos/publications/index.html.

[105] GRIPPS. *Grid Protein Pattern Scanning*, http://gripps.ibcp.fr/index.php.

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

[107] JXTA. *Project JXTA Objectives*, http://www.jxta.org/.

[108] D. KATABI, M. HANDLEY, C. ROHRS. *Congestion control for high bandwidth-delay product networks*, in "ACM SIGCOMM 2002", ACM Press, 2002, p. 89–102.

[109] 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.

[110] 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.

[111] 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, nº 2536, November 2002, p. 274-278.

[112] 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, nº 5-6, 1999, p. 649-658.

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

[114] G. ROBERTAZZI. *Ten Reasons to Use Divisible Load Theory de Thomas*, in "Computer", vol. 36, n^o 5, May 2003, p. 63-68.

[115] 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.

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

[117] A. TAKEFUSA, H. CASANOVA, S. MATSUOKA, F. BERMAN. *A Study of Deadline Scheduling for Client-Server Systems on the Computational Grid*, in "the 10th IEEE Symp. on High Performance and Dist. Comput. (HPDC'01)", August 2001.

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