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

2004

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

Yves Caniou [Lecturer]
Eddy Caron [Assistant Professor]
Yves Robert [Professor]

Faculty members from Université de Franche-Comté

Jean-Marc Nicod [Assistant Professor]
Laurent Philippe [Professor]

Project technical staff

Raphaël Bolze [on contract since February 01, 2004]
Holy Dail [on contract from INRIA]
Christophe Pera [on contract from ENS Lyon until January 17, 2004]

Post-doctoral fellow

Matthias Colin [CNRS, until August 31, 2004]
Alan Su [INRIA]

Ph. D. students

Pushpinder-Kaur Chouhan [MENRT grant]
Sylvain Dahan [Region grant]
Bruno Del Fabbro [FAF grant]
Abdou Guermouche [MENRT grant, until August 31, 2004]
Arnaud Legrand [ENS grant, until August 31, 2004]
Loris Marchal [ENS grant]
Suphakit Niwattanakul [Thailand grant]
Hélène Renard [MENRT grant (ACI GRID)]
Antoine Vernois [MENRT grant (ACI GRID)]

2. Overall Objectives

Keywords: *Grid computing, Programming environment, algorithmic of heterogeneous systems, distributed application, library*.

Parallel computing has spread into all fields of applications, from classical simulation of mechanical systems or weather forecast to databases, video-on-demand servers or search tools like Google. From the architectural point of view, parallel machines have evolved from large homogeneous machines to clusters of PCs (with sometime boards of several processors sharing a common memory, these boards being connected by high speed networks like Myrinet). However the need of computing or storage resources has continued to grow leading to the need of resource aggregation through Local Area Networks (LAN) or even Wide Area Networks (WAN). The recent progress of network technology has made it possible to use highly distributed platforms as a single parallel resource. This has been called Metacomputing or more recently Grid Computing [80]. An enormous amount of financing has recently been put on this important subject, leading to an exponential growth of the number of projects, most of them focusing on low level software details. 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 [92]. 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 [85][77][80]. 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,
- 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.

Table 1. The main keywords of the GRAAL project

Algorithmic Design + Middleware/Libraries + Applications
over heterogeneous architectures and the Grid

[1] Réseau à Vraiment Très Haut Débit.

3. Scientific Foundations

3.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Arnaud Legrand, Loris Marchal, 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 [78][91][104][106] and the references therein. This model was introduced for homogeneous processors, and has been (straightforwardly) extended for 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: Abdou Guermouche, 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 time of execution 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 functionalities 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 [89][90], 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 [100] 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 [82][83]. 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 dynamic,
- these algorithms are currently used inside industrial applications, and
- the evolution of high performance platforms, more heterogeneous and less predictable, requires applications to adapt using a mixture of dynamic and static approaches, as is allowed by our approach.

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, Matthias Colin, Sylvain Dahan, Holy Dail, Bruno Del Fabbro, Frédéric Desprez, Jean-Marc Nicod, Christophe Pera, Laurent Philippe, Alan Su, Antoine Vernois.

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's 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 [81] or Global computing [94] 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 [101][102]) offers an easy access to available resources to a Web browser, a Problem Solving Environment (PSE), 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 middlewares 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 [84], Ninf [103], NEOS [93], OmniRPC [105], and more recently DIET developed in the GRAAL project. 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 an NES we need to address issues linked to several well-known research domains, among which we focus on:

- scheduling to allow clients to chain requests in a workflow mode,
- 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,
- distributed algorithms to manage the requests and the dynamic behavior of the platform.

Finally, "classical" parallelism is used at the server level and between servers.

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

3.3.1. Resource Management for NES platforms.

The main function of a NES is to connect clients and servers. Due to its intermediate position, it also has to act as a resource manager. Several research topics are concerned: scheduling, data management, fault tolerance, and scalability (as well as security).

The first task done by a NES system is to find the most appropriate server available. The work has to be balanced between the different servers and several criteria can be met (execution time of one request, of the whole application, steady state, ...). Thus, scheduling plays a key role in resource management. In the case of a NES system, several models can be used. The scheduling can be on-line when requests are handled by the scheduler without any knowledge of the requests sequence and dependences, or offline if we take a picture of the target platform and schedule a whole program. Since scalability is one of our main concerns, we need to improve the scalability of the scheduling itself by using distributed scheduling techniques.

The second issue is data placement and persistence. Parallel applications usually run on very large data, unlike traditional client/servers applications. This means that the management of these data will be costly (both in transfer and storage). In the basic request submission scheme, data are sent by the client to a server. After the computation, results are returned from the server to the client. There is no other way to optimize data access in this context except choosing a server connected to the client by a high speed network or using on-line data compression [96] during transfer. However, problems are rarely solved in one step and a parallel job submission is often linked to other job submissions, in a workflow mode with dependences between the different tasks. In that case, it is possible to optimize data mapping and transfer by choosing a server for a job depending on the location of the data. Experiments have already proved that leaving data on servers and re-using them for the next job leads to substantial performance improvement.

Another research domain related to designing a NES is fault tolerance. Implementing a distributed environment without taking fault tolerance into consideration is meaningless. Indeed, the probability of a server fault or crash increases with the size of the network. NES are designed to manage resources distributed in large scale networks and several kinds of faults may occur, e.g. network or server crashes. Several research projects have already tried to address this issue in PSE environments. The CUMULVS project [98] proposes fault-tolerance of distributed simulations through heterogeneous task migration and user-directed checkpointing. Fault tolerance is managed at the application level. Other projects like the Los Alamos Message Passing Interface (LA-MPI) [95] or MPICH-V [86] provide a fault-tolerant message passing system. In the CORBA middleware fault tolerance is standardized [79].

4. Application Domains

4.1. Applications of Sparse Direct Solvers

Our activity on sparse direct solvers and more precisely multifrontal solvers in distributed environments goes as far as making competitive software available to users. Such methods have a wide range of applications and they are at the heart of most techniques in numerical 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 implying 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, CAD, CAE, 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 FEM, 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 domain and can return the Schur complement), 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 device 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 calculation 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 allow the understanding and the prediction of chemical properties.

Among the various theoretical approaches available, quantum chemistry is at a central position as all modern chemistry relies on it. This scientific domain is quite complex and involves heavy computation. The energy of a model is a function of all its degrees of freedom and the CPU time needed to compute it rapidly increases with the system size (*i.e.*, the number of atoms involved in the model).

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 Application

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, Eddy Caron, Pushpinder-Kaur Chouhan, Matthias Colin, Sylvain Dahan, Holy Dail, Frédéric Desprez [correspondent], Bruno Del Fabbro, Jean-Marc Nicod, Christophe Pera, Laurent Philippe, Alan Su, Antoine Vernois.

5.1.1. DIET

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) [108] 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:

- **Client**
  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.
- **Master Agent (MA)**
  An 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 MA can be deployed on the network to balance the load among the clients.
- **Local Agent (LA)**
  An LA aims at transmitting requests and information between MAs and servers. The information stored on an 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 an LA.

- **Server Daemon (SeD)**
  A 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..

5.1.2. *FAST*

FAST (*Fast Agent's System Timer*) http://graal.ens-lyon.fr/FAST is a tool for dynamic forecasting of Network-Enabled Servers performance. This is a software package allowing client applications to get an accurate forecast of routine needs in terms of completion time, memory space, and amount of communication, as well as of current system availability. FAST relies on existing low level software packages, i.e., network and host monitoring tools, and some of our developments in modeling computation routines.

The goal of the FAST library is to provide the information needed by a scheduler. FAST models the needs of the tasks both in terms of time and memory space. Appropriate tools like NWS [108] are used to monitor the dynamic availability of system resources. FAST is also able to aggregate these two kinds of information in order to forecast the current computation time of a given task on a given machine. The goal of FAST is not to perform task placement, but to acquire the required knowledge to achieve it.

An extension of the FAST library to handle parallel routines is under development. We combine estimations given by FAST about sequential computation routines and network availability with parallel routine models coming from analysis.

5.2. MUMPS

Participants: Jean-Yves L'Excellent [correspondent], Abdou Guermouche.

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), whose results and developments were public domain. Lots of developments have been done by the authors since the end of that project, in order to enhance the software with more functionalities and integrate new results arising from our research. MUMPS is distributed free of charge http://graal.ens-lyon.fr/MUMPS and is currently being used by several hundred academic and industrial users, from a wide range of application fields (see Section 4.1).

MUMPS uses a direct method, the multifrontal method and is currently developed in collaboration with ENSEEIHT-IRIT (Toulouse, France). MUMPS 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:

- C or Fortran 90 interface,
- 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.

5.3. SimGrid v2

Participant: Arnaud Legrand [correspondent].

The first version of SimGrid [87] footnotehttp://grail.sdsc.edu/simgrid was a discrete-event simulation toolkit. It provided a set of core abstractions and functionalities that can be used to easily build simulators for specific application domains and/or computing environment topologies. This allows the simulation of arbitrary performance fluctuations such as the ones observable for real resources due to background load. However, this first version lacked a number of abstractions (e.g. routing, scheduling agents). With SimGrid v2 we have added a new software layer to provide high-level abstractions and the software thus provides two interfaces:

SG: The original low-level toolkit does the simulation in terms of explicitly scheduling tasks on resources.

MSG: A simulator built using SG. This layer implements realistic simulations based on the foundational SG and is more application-oriented. Simulations are built in terms of communicating agents.

The scheduling algorithms with SimGrid should always be described in terms of agents that run at locations and interact by sending, receiving, and processing simulated application tasks. Agents do not have direct access to paths but can send a task to another location using a channel. In fact, a location may have many mailboxes and a channel is then simply a mailbox number. So sending a task to a location using a channel amounts to transferring the task on a particular path, depending on the emitter location and on the destination, and to put it in a particular mailbox.

SimGrid v2 enables scalable, configurable, extensible, and fast simulations for investigating novel scheduling techniques for heterogeneous and distributed platforms. SimGrid has already been used successfully and the SimGrid user community is currently undergoing a dramatic expansion. SimGrid is also used for educational purposes in a course on Parallel Algorithms and Architectures at the École normale supérieure de Lyon.

6. New Results

6.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

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

Participants: Arnaud Legrand, Loris Marchal, Hélène Renard, Yves Robert, 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, the 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.

In 2004, 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 the 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. 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.

In a different context, 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 propose a polynomial-time algorithm to solve the maximum weighted flow off-line scheduling problem on unrelated machines. We also show how to solve the maximum weighted flow off-line scheduling problem with preemption on unrelated machines.

6.1.4. Load-Balancing for Communication-Aware Models.

For all our studies we use communication models as realistic as possible. In communication-aware models, there are 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 provided with 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 [97], 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 ring 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 uni-directional 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).

6.1.5. 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).

In 2004, 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.

6.2. 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, Matthias Colin, Sylvain Dahan, Holy Dail, Frédéric Desprez, Bruno Del Fabbro, Jean-Marc Nicod, Christophe Pera, Laurent Philippe, Alan Su, Antoine Vernois.

2004 is the year of the release of version 1.1 of DIET. We stabilized all developments and wrote the *User's Manual* and *Programmer's Guide* corresponding to this version.

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” [107]. 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 feature can also be used as a more friendly configuration of DIET that provides more equal sharing with other users' processes. 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 presented a very simple algorithm for task placement refinement, but it is our goal to utilize this feature in future work.

We then presented experiments comparing the performance of the standard DIET scheduling approach against that of the SeD queue and global task scheduler approaches. 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. Specifically, we added

4 new SeDs mid-way through the execution. 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. On the other hand, it is very reassuring that in a situation such as this where no advantage can be expected of our extensions, they also do not introduce significant overheads nor slowdown the system throughput. 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. A peer-to-peer extension for DIET.

We have developed a peer-to-peer extension for DIET using JXTA http://www.jxta.org/ that allows a dynamic connection of DIET components. JXTA provides functionalities such as passing through firewall and similar network protections, or dynamically discovering other peers. These tools are mandatory to develop a Multi 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 smart algorithms for discovery. However, connecting Corba components to JXTA is not easy. We can consider that the current JXTA Multi-MA is composed of 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. 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.

6.2.4. 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 is composed of three parts. The first part (Log-Component) 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 logs messages must be sorted, so 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 type of the log. Tags can be defined using a configuration file on the LogCentral side.

6.2.5. New applications into DIET.

The integration of three different applications increase the capacities of our DIET Problem Solving Environment:

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

Sparse service In collaboration with the GRID TLSE project, DIET gives a new functionality around sparse direct solvers. This allows a solver to be called remotely from a client. A first private web access prototype is available for project members. (See also Section 6.3.)

Bioinformatics service For the GriPPS project [6] we have developed the prototype of a DIET bioinformatic server. This server is able to run the *pattinprot* algorithm remotely. This algorithm allows a user to scan a protein databases (PROSITE) to match with one or more protein patterns (regular expressions). Based on biological criteria, *pattinprot* is able to select biologically relevant but non perfectly matching proteins.
We have also developed a web service to submit *pattinprot* jobs to DIET servers. The web interface allows users to submit *pattinprot* jobs in a simple manner without requiring any knowledge of DIET.

Molecular Physics Simulation For the GridFC project of ISTI[7], we have developed a DIET server for a Kinetic Monte-Carlo simulation. This server is accessible through a web server (servlet) to facilitate its use. Interactivity will be provided with this server as the client needs to supervise the evolution of the results.

The sparse and bioinformatic service prototypes have been presented during the DIET demo at SC 2004.

6.2.6. 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 worked an algorithm that computes at the same time the mapping of data and computational requests on these data.

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 http://gripps.ibcp.fr/index.php).

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 improves the throughput of the platform.

6.2.7. Data persistence

The DIET approach consists in selecting appropriate servers to solve computational requests on behalf of the clients connected to it. The first version of our software relied only on the CPU performance of the servers.

However, for scientific applications which may 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.

If several requests share data or are data-dependent, one must keep the data in place on the remote servers to reduce the communications overhead. Data stored within the platform are called persistent.

We have designed and implemented a data management service to provide data persistence for the DIET platform. Data are managed implicitly by the client and explicitly by the infrastructure. Indeed, once a data is transmitted, the client does not have to transfer it again. It only has to state whether the data is persistent. Data might then be moved by DIET between servers for scheduling purpose.

Data and computation management have been separated. The Data Tree Manager is built upon two main entities: the Logical Data Manager (LDM))and the Physical Data Manager (PDM). The Logical Data Manager

[6] Grid Protein Pattern Scanning

[7] See Section 7.1.

provides the global knowledge of the localization of each data. It is based on a set of `LocManager` objects which are linked to the Agents (MA or LA). The Physical Data Manager is composed of a set of `DataManager` objects linked with SeDs. Each DataManager stores and provides data to the server on demand. It offers functionalities for data movement between servers and informs its `LocManager`.

6.2.8. Service localization

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 broadcasted on the network of MA until an agent finds the appropriate server.

In this context, service localization and discovery algorithms are research issues close to lookup problematics 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 as the number of clients is small. When the size of the network grows, standard broadcasting algorithms will lead to bottlenecks and Master Agent overloading.

For large scale networks, we propose a data structure which allows to distribute the graph traversal to avoid the bottlenecks. This structure has been called Distributed Spanning Tree as it provides a different spanning tree for each Master Agent.

6.3. Parallel Direct Solvers for Sparse Systems of Linear Equations

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

Participants: Abdou Guermouche, Jean-Yves L'Excellent.

6.3.1. Extension of the software platform MUMPS.

Since MUMPS version 4.3, released in July 2003, minor modifications of the package have led us to now distribute MUMPS version 4.3.2. We had many interactions with a large amount of academic and industrial users, and had continuously provided them with a significant level of support. Since that release, new software work has included the development of techniques to manage multiple and sparse right-hand sides, a new approach to scheduling (under memory constraints), various bug fixes and improvements (memory management, increased performance, better numerical stability for partial factorizations, ...). We have provided these new functionalities to a few partners that help us validating them. A major release of the package is scheduled early 2005.

6.3.2. Memory-minimizing schedules for the multifrontal method

We have worked on constructing memory-minimizing schedules for the (sequential) multifrontal method. Starting from the algorithms introduced by Liu [99], we designed new schedules to allocate and process tasks, that improve the memory usage. This generalizes two existing factorization and memory-allocation schedules, by allowing a more flexible task allocation scheme, together with a specific tree traversal. We proposed optimal algorithms for this new class of schedules, and demonstrated experimentally their benefit for some real-world matrices from sparse matrix collections, using a simulator. We also proposed variants for both the active and the total memory, and for a slightly different class of schedules (in-place algorithms) that further improve the memory consumption.

6.3.3. Information exchange mechanisms and coherence of the view in a distributed system

Since the parallel multifrontal method is based on dynamic distributed schedulers that take decisions based on the view that they have of the system, we have been working on mechanisms aiming at maintaining on each processor a view of the system as correct as possible; note that the *view* of the system consists in information about the workload, the memory occupation, ...that are of use to schedulers to map/assign some of the computational tasks. We designed a new mechanism based on exchanging load variations that improves the coherency of the load and memory information in the context of distributed applications. We studied how this approach compares to the *distributed snapshot* approach [88] introduced by Chandy and Lamport, combined with a distributed leader election (if several processors initiate a snapshot simultaneously).

6.3.4. Hybrid scheduling strategies for the parallel multifrontal method

Starting from schedulers that optimize either the makespan, or the memory usage in the parallel multifrontal factorization, we considered the problem of designing a dynamic scheduling strategy that takes into account both workload and memory information. The originality of our approach is that we base our estimations (work and memory) on a static optimistic scenario during the analysis phase and then use it during the factorization phase to constrain the dynamic decisions. The task scheduler has been redesigned to take into account these new features. Moreover performance have been improved because the new constraints allow the new scheduler to make optimal decisions that were forbidden or too dangerous in unconstrained formulations. Performance analysis shows that the memory estimation becomes much closer to the memory effectively used and that even in a constrained memory environment we decrease the factorization time with respect to the initial approach. This work is achieved in collaboration with P. Amestoy and S. Pralet from ENSEEIHT-IRIT.

6.3.5. Implicit out-of-core approaches.

In a previous work, we proposed a new way to improve the performance of the factorization of large sparse linear systems that cannot fit in memory. Instead of rewriting a large part of the code to implement an out-of-core algorithm with explicit IO, we modify the paging mechanisms in such a way that IO are transparent. We have now extended our mechanism to the parallel case. The main difference with a sequential execution concerns the scheduling aspects of the parallel multifrontal method. Whereas in the sequential execution it is possible to know the memory access pattern of the solver in advance, in the parallel execution, the paging strategy has to deal with the dynamic load balancing mechanism. Thus, we studied the paging behavior of the parallel multifrontal with several scheduling strategies (performance-based, memory-based). Preliminary results show that our approach, implicit parallel out-of-core schemes based on a modified paging policy, can give good results especially when they are used with memory-based scheduling strategies that aim at reducing the memory occupation as much as possible. In addition, this approach allows us to find the key-points for an efficient explicit out-of-core extension of the parallel multifrontal method. This work is in collaboration with O. Cozette and G. Utard from LARIA.

6.3.6. Experimentation on real-life test problems.

We have been collaborating with new users of MUMPS, that provide us new problems and help us validating and improving our algorithms. For example, SAMTECH S.A. and BRGM have recently provided us with huge challenging problems that we will use to better understand the performance and memory limits of our approach.

We have in fact informal collaborations around MUMPS with a number of institutions: (i) industrial teams who experiment and validate our package, (ii) research teams with whom 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 in optimization, (v) physicists, chemists, etc., in various fields where robust and efficient solution methods are critical in their simulation process. We aim at validating 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).

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

We use the middleware DIET in the context of the GRID TLSE project (see [8]), coordinated by ENSEEIHT-IRIT, whose goal is to design an expert site providing a one-stop shop for users of sparse matrix 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

[8] http://www.enseeiht.fr/lima/tlse/

problem. Each expertise request leads to a number of elementary requests on the grid for which the middleware tools developed by GRAAL are used.

The final expert site is now being specified and developed; we are involved in the software layers related to the use of the middleware DIET and in the definition/specification of how various sparse direct solvers shall be used in the context of TLSE. Latest events include the review meeting that occurred on September 22, 2004 in Toulouse, and a demonstration of the current status of the project at the conference SC'2004 (Supercomputing 2004), Pittsburgh, USA.

7. Other Grants and Activities

7.1. Regional Projects

7.1.1. 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.

7.1.2. 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. L. Philippe leads of the Trader project whose aim is to develop tools and experiences on services discovery in distributed systems.

7.2. National Contracts and Projects

7.2.1. Ministry Grant: RNRT VTHD++, 2 years, 2002-2004

E. Caron and F. Desprez participated between 1999 and 2000 to the RNTL project VTHD [9] whose aim was to connect several research centers (and their clusters) in France through a high speed network at 2.5 Gb/s. Several research projects have been completed at different levels (network management, middleware, and applications). One of the target applications was the first version of the DIET environment.

Following this project, the VTHD++ started in 2002. Our goal was to test several protocols of quality of service and security using the last version of the DIET platform. We also studied the scalability of our environment and we port several applications.

F. Desprez leads the application part of both projects.

7.2.2. Ministry Grant: ACI Grid ASP, 3 years, 2002-2005

F. Desprez leads the ACI Grid ASP project. This multidisciplinary project aims at porting several applications on top of the DIET platform.

E. Caron, J.-M. Nicod, and L. Philippe also participate to this project.

7.2.3. 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".

[9] URL: http://www.vthd.org.

7.2.4. 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 the most robust for my type of 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.

7.2.5. INRIA new investigation Grant: ARC INRIA RedGrid, 2 years, 2003-2004

The aim of the RedGrid project is to develop algorithms and heuristics for the redistribution of data between clusters connected in a Grid Environment. Target applications are the DIET environment, Grid Corba Components, and EPSN, a computational steering application. A library has been developed and validated on several applications.

Partners of this projects are GRAAL, PARIS from IRISA, ScAlApplix from INRIA Futurs, and ALGORILLE from LORIA.

E. Caron, F. Desprez, J.-Y. L’Excellent, Y. Robert, and F. Vivien participate to this project.

7.2.6. Ministry Grant: ACI Grandes masses de données GridExplorer, 2003-2004

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.

7.2.7. Ministry Grant: ACI Grandes masses de données Grid Data Service, 2003-2004

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[10]. 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.

7.2.8. CNRS Grant: Enabling a Nation-Wide Experimental Grid (ENWEG)

ENWEG is a study for preparing the deployment of a nation-wide experimental Grid. This project aims at identifying scientific and technical issues and propose solutions in the perspective of building an experimental Grid platform gathering nodes geographically distributed in France. This project is a specific action RTP 8 from CNRS.

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

7.2.9. CNRS Grant: AS Méthodologie de programmation des grilles

The aim of this project is to define the main research directions on grid programming. This reflection should especially take care of the relative implications of the actual works on algorithms, applications, runtime environments, network protocols, etc.

F. Vivien participates to this project.

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

ENS Lyon is involved in the GRID’5000 project, which aim is to build an experimental Grid platform gathering eight sites geographically distributed in France. Each site hosts several clusters connected through the RENATER network.

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

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.

7.3. International Contracts and Projects

7.3.1. 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 2 years and we organized a workshop in 2004.

The DIET software from GRAAL was used to validate some of the scheduling heuristics and SimGrid2 was improved and used to simulate our platform.

For the last year of the project, several other visits are planned and a final workshop will be organized in San Diego in November 2005.

Workshop *Scheduling for large-scale heterogeneous platforms*

This workshop was organized on August 27-29, 2004, in the CNRS center of Aussois. The topic was scheduling for large-scale heterogeneous platforms. The organizers were Larry Carter, Henri Casanova, and Jeanne Ferrante (University of California at San Diego), Frédéric Desprez and Yves Robert.

Here is the list of the talks:

Session : Models and objectives for scheduling

- F. Berman : Next generation scheduling
- J. Weismann : Scheduling for robustness
- M. Lauria : The Organic Grid : Self organizing computation on peer-to-peer network
- E. Jeannot : Multi criteria scheduling for heterogeneous and distributed system

Session : Scheduling algorithms

- M. Drozdowski : Scheduling multiple divisible tasks
- O. Beaumont : Pipelined broadcasts and multicasts on heterogeneous platforms
- A. Rosenberg : A pebble game for Internet computing
- F. Vivien : Load-balancing iterative computations on distributed heterogeneous platforms

Session : Robust and multi-criteria scheduling techniques

- H. Casanova : Modeling large scale platforms for the analysis and the simulation of scheduling strategies
- T. Kielmann : Fault tolerant scheduling of fine-grained tasks in grid environments
- L. Carter : Interference-aware scheduling
- D. Trystram : Dealing with disturbance while scheduling (in Grids)

Session : Scheduling tools

- F. Cappello : Hybrid preemptive scheduling of MPI applications
- K. Seymour, A. Yarkhan : Scheduling in GridSolve
- C. Perez : On the deployment and the execution of component applications on the Grid
- H. Dail : Distributed scheduling in DIET

The slides of all presentations are available at http://graal.ens-lyon.fr/aussois/. Following this workshop, a special issue of the *International Journal of High Performance Computing Applications* (IJHPCA) will be edited by the workshop organizers (the expected issue of IJHPCA if Spring 2006). We point out that a second workshop (on the same topic) will be organized in San Diego, on the week before SuperComputing'2005, in November 2005.

7.3.2. NSF-INRIA, The University of Minnesota, USA

This project that finished in 2004 aimed at developing robust parallel preconditioners for the solution of large systems of equations. We provided direct methods and were interested in the parallelization and memory reduction aspects of these solvers. This project involved collaborations with the INRIA projects ALADIN and ScAlApplix, ENSEEIHT-IRIT (Toulouse, France), the University of Minnesota, the University of Indiana and the Lawrence Berkeley Laboratory (NERSC).

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

F. Desprez is the French coordinator of a 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).

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

Y. Robert is the French coordinator of a 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).

7.3.5. STAR Project KISTI-INRIA

The GRAAL Project, with the PARIS Project-Team located at INRIA/IRISA have 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.

8. Dissemination

8.1. Scientific Missions

STIC department of CNRS. Yves Robert is co-chairing (with B. Plateau) the interdisciplinary network *Calcul à hautes performances et calcul réparti* (High performance and distributed computing). He has been nominated head of the evaluation committee of the LIFC laboratory (Besançon), and member of the evaluation committee of the LIENS laboratory (Paris).

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 (F. Desprez, Y. Robert, F. Vivien). Yves Robert is leading the CNRS contribution. Frédéric Vivien is responsible of two tasks in the scheduling workpackage.

Austrian Science Fund (FWF) F. Desprez has evaluated one 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.

8.2. Animation Responsibilities

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

8.3. Edition and Program Committees

Frédéric Desprez is an associate editor of *Parallel and Distributed Computing Practices* (http://www.cs.okstate.edu/~pdcp) and *Computing Letters* (COMPULETT).
F. Desprez participated to the program committees of ICCS04, HiPC04, ISPDC04, CLADE04, PPGaMS, EuroPVMMPI04, and the Second International Summer School on Grid Computing. He was part of the scientific committee of the DRUIDE School on Grid Computing. He is a member of the EuroPar Advisory board.
F. Desprez was vice-chair at the HiPC conference in Bangalore (algorithm track). He also gave a keynote talk at this conference. F. Desprez gave an invited talk in the workshop "Clusters and Grids for Parallel Scientific Computing", Faverges (September 2004).

Jean-Yves L'Excellent was a member of the program committee (topic Algorithms) of SC'2004 (Supercomputing 2004), Pittsburgh, USA. He was vice program chair of the International Symposium on High Performance Computational Science and Engineering (HPCSE'04), France, August 2004. He is a member of the organizing/scientific committee of the Second International Workshop on Combinatorial Scientific Computing (CSC05), Toulouse, France, June 2005.

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'04 (European Symposium on Parallel and Distributed Processing), Lugano, Switzerland; Euro PVM-MPI 2004, Bupadest, Hungary.
Y. Robert was vice-chair (topic Algorithms) of the program committee of SC'2004 (Supercomputing 2004), Pittsburgh, USA. 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.
Y. Robert gave an Abell distinguished lecture in computer engineering in Fort Collins, USA (April 2004). Y. Robert gave a seminar in the electrical engineering department of Stony Brook (November 2004). Y. Robert gave an invited talk in the workshop "Clusters and Grids for Parallel Scientific Computing", Faverges (September 2004).
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, to appear 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 January 2006.

Frédéric Vivien will be a member of the program committee of ISPA'05 (International Symposium on Parallel and Distributed Processing and Applications), Nanjing, China, November 2-5, 2005, and of the Workshop on scheduling for parallel computing, Poznan, Poland, September 13-16, 2005 to be held in conjunction with PPAM 2005.

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

8.4. Administrative and Teaching Responsibilities

8.4.1. *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.

8.4.2. *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*.

ENSEIRB, Bordeaux. F. Desprez gave several lectures around load-balancing for numerical problems and around Grid Computing for the third year of ENSEIRB (Bordeaux).

Master in Computer Science at Université de Franche Comté. Bachelor degree (Maîtrise) in computer science, L. Philippe is responsible of the Client/Server and distributed programming lecture (since 2000).

Bachelor degree (Maîtrise) in computer science, J.-M. Nicod is 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.

9. 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*, 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*, 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] A. GUERMOUCHE. *Étude et optimisation du comportement mémoire dans les méthodes parallèles de factorisation de matrices creuses*, Ph. D. Thesis, École Normale Supérieure de Lyon, July 2004.

Articles in referred journals and book chapters

[14] 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.

[15] 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.

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

[17] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Steady-state scheduling on heterogeneous clusters*, in "Int. J. of Foundations of Computer Science", to appear, 2005.

[18] E. CARON, F. DESPREZ, M. QUINSON, F. SUTER. *Performance Evaluation of Linear Algebra Routines*, in "International Journal of High Performance Computing Applications", Special issue on Clusters and Computational Grids for Scientific Computing (CCGSC'02), vol. 18, n^o 3, 2004, p. 373-390.

[19] E. CARON, F. SUTER. *Parallel Extension of a Dynamic Performance Forecasting Tool*, in "Parallel and Distributed Computing Practice (PDCP)", Special issue on selected papers of ISPDC'02. To appear, 2004.

[20] 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.

[21] 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.

[22] S. GENAUD, A. GIERSCH, F. VIVIEN. *Load-Balancing Scatter Operations for Grid Computing*, in "Parallel Computing", vol. 30, n^o 8, 2004, p. 923-946.

[23] A. GIERSCH, Y. ROBERT, F. VIVIEN. *Scheduling tasks sharing files on heterogeneous master-slave platforms*, in "Journal of Systems Architecture, special issue on Parallel, Distributed and Network-based Processing: selected papers from the 12th Euromicro Conference", to appear, 2004.

[24] 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.

[25] F. VIVIEN, N. WICKER. *Minimal enclosing parallelepiped in 3D*, in "Computational Geometry: Theory and Applications", vol. 29, n^o 3, 2004, p. 177-190.

Publications in Conferences and Workshops

[26] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Assessing the impact and limits of steady-state scheduling for mixed task and data parallelism on heterogeneous platforms*, in "HeteroPar'2004: International Conference on Heterogeneous Computing, jointly published with ISPDC'2004: International Symposium on Parallel and Distributed Computing", IEEE Computer Society Press, 2004, p. 296-302.

[27] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Complexity results and heuristics for pipelined multicast operations on heterogeneous platforms*, in "2004 International Conference on Parallel Processing (ICPP'2004)", IEEE Computer Society Press, 2004, p. 267-274.

[28] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Pipelining broadcasts on heterogeneous platforms*, in "International Parallel and Distributed Processing Symposium, IPDPS'2004", IEEE Computer Soci-

ety Press, 2004, 19b (10 pages).

[29] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Steady-state scheduling on heterogeneous clusters: why and how?*, in "6th Workshop on Advances in Parallel and Distributed Computational Models, APDCM 2004", IEEE Computer Society Press, 2004, 171a (8 pages).

[30] E. CARON, P.-K. CHOUHAN, F. DESPREZ. *Deadline scheduling with priority for client-server systems on the Grid*, in "Grid Computing 2004. IEEE International Conference On Grid Computing. Super Computing 2004, Pittsburgh, Pennsylvania", R. BUYYA (editor)., October 2004, p. 410-414.

[31] E. CARON, P.-K. CHOUHAN, A. LEGRAND. *Automatic Deployment for Hierarchical Network Enabled Server*, in "The 13th Heterogeneous Computing Workshop (HCW 2004), Santa Fe. New Mexico", April 2004, 109b (10 pages).

[32] H. CASANOVA, F. DESPREZ, F. SUTER. *From Heterogeneous Task Scheduling to Heterogeneous Mixed Parallel Scheduling*, in "Proceedings of the 10th International Euro-Par Conference (Euro-Par'04), Pisa, Italy", M. DANELUTTO, D. LAFORENZA, M. VANNESCHI (editors)., Lecture Notes in Computer Science, vol. 3149, Springer, August/September 2004, p. 230–237.

[33] G. CHUN, H. DAIL, H. CASANOVA, A. SNAVELY. *Benchmark Probes for Grid Assessment*, in "Proceedings of the 18th International Parallel and Distributed Processing Symposium - Workshop 17, Santa Fe, New Mexico", High-Performance Grid Computing Workshop, IEEE Computer Society Press, April 2004, 276a (8 pages).

[34] K. COOPER, A. DASGUPTA, K. KENNEDY, C. KOELBEL, A. MANDAL, G. MARIN, M. MAZINA, J. MELLOR-CRUMMEY, F. BERMAN, H. CASANOVA, A. CHIEN, H. DAIL, X. LIU, A. OLUGBILE, O. SIEVERT, H. XIA, L. JOHNSSON, B. LIU, M. PATEL, D. REED, W. DENG, C. MENDES, Z. SHI, A. YARKHAN, J. DONGARRA. *New Grid Scheduling and Rescheduling Methods in the GrADS Project*, in "Proceedings of NSF Next Generation Systems Program Workshop, Santa Fe, New Mexico", in conjunction with IPDPS'2004, April 2004, 199a (8 pages).

[35] O. COZETTE, A. GUERMOUCHE, G. UTARD. *Adaptive paging for a multifrontal solver*, in "Proceedings of the 18th annual international conference on Supercomputing, Saint Malo, France", ACM Press, 2004, p. 267–276.

[36] S. DAHAN, J.-M. NICOD, L. PHILIPPE. *Scalability in a GRID Server Discovery Mechanism*, in "10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems, Suzhou, China", IEEE Press, May 2004, p. 46-51.

[37] M. DAYDÉ, L. GIRAUD, M. HERNANDEZ, J.-Y. L'EXCELLENT, C. PUGLISI, M. PANTEL. *An Overview of the GRID-TLSE Project*, in "Poster Session of 6th International Meeting VECPAR'04, Valencia, Espagne", June 2004, p. 851-856.

[38] B. DEL FABBRO, D. LAIYMANI, J.-M. NICOD, L. PHILIPPE. *Data Management in Grid Applications Providers*, in "IEEE International Conference DFMA'05, Besançon, France", to appear, February 2005.

[39] B. DEL-FABBRO, D. LAIYMANI, J.-M. NICOD, L. PHILIPPE. *A Data Persistency Approach for the DIET*

Metacomputing Environment, in "International Conference on Internet Computing, Las Vegas, USA", H. R. Arabnia, O. Droegehorn, S. Chatterjee (editors)., CSREA Press, June 2004, p. 701-707.

[40] F. Desprez, E. Jeannot. *Improving the GridRPC Model with Data Persistence and Redistribution*, in "3rd International Symposium on Parallel and Distributed Computing (ISPDC), Cork, Ireland", July 2004.

[41] A. Giersch, Y. Robert, F. Vivien. *Scheduling tasks sharing files from distributed repositories*, in "Euro-Par-2004: International Conference on Parallel Processing", LNCS 3149, Springer Verlag, 2004, p. 246-253.

[42] A. Giersch, Y. Robert, F. Vivien. *Scheduling tasks sharing files on heterogeneous master-slave platforms*, in "PDP'2004, 12th Euromicro Workshop on Parallel, Distributed and Network-based Processing", IEEE Computer Society Press, 2004, p. 364-371.

[43] A. Guermouche, J.-Y. L'Excellent. *Memory-based scheduling for a Parallel Multifrontal Solver*, in "18th International Parallel and Distributed Processing Symposium (IPDPS'04)", 2004, 71a (10 pages).

[44] A. Legrand, L. Marchal, Y. Robert. *Optimizing the steady-state throughput of scatter and reduce operations on heterogeneous platforms*, in "6th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2004", IEEE Computer Society Press, 2004, 176a (8 pages).

[45] H. Renard, Y. Robert, F. Vivien. *Data redistribution algorithms for homogeneous and heterogeneous processor rings*, in "International Conference on High Performance Computing HiPC'2004", LNCS, vol. 3296, Springer Verlag, 2004, p. 123-132.

[46] K. Seymour, C. Lee, F. Desprez, H. Nakada, Y. Tanaka. *The End-User and Middleware APIs for GridRPC*, in "Workshop on Grid Application Programming Interfaces, In conjunction with GGF12, Brussels, Belgium", September 2004.

[47] A. Su, F. Berman, H. Casanova. *On the Feasibility of Running Entity-Level Simulations on Grid Platforms*, in "Proceedings of Grid Computing (Grid 2004)", in conjunction with Supercomputing 2004, nov 2004, p. 312–319.

[48] G. Utard, A. Vernois. *Data Durability in Peer-to-Peer Storage Systems*, in "Proc. 4th Workshop on Global and Peer to Peer Computing, Chicago", IEEE/ACM CCGrid Conference, April 2004, (9 pages).

[49] H. Xia, H. Dail, H. Casanova, A. A. Chien. *The MicroGrid: Using Emulation to Predict Application Performance in Diverse Grid Network Environments*, in "Challenges of Large Applications in Distributed Environments (CLADE), Honolulu, Hawaii", In conjunction with HPDC-13, IEEE Computer Society Press, June 2004, p. 52-63.

Internal Reports

[50] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet. *Hybrid scheduling for the parallel solution of linear systems*, Also available as LIP report RR2004-53 and as an ENSEEIHT-IRIT technical report., Research Report, n° RR-5404, INRIA, December 2004, http://www.inria.fr/rrrt/rr-5404.html.

[51] O. Beaumont, A. Legrand, L. Marchal, Y. Robert. *Assessing the impact and limits of steady-state*

scheduling for mixed task and data parallelism on heterogeneous platforms, Also available as INRIA Research Report RR-5198, Research Report, LIP, ENS Lyon, France, April 2004, http://www.inria.fr/rrrt/rr-5198.html.

[52] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Complexity results and heuristics for pipelined multicast operations on heterogeneous platforms*, Also available as INRIA Research Report RR-5123, Research Report, LIP, ENS Lyon, France, February 2004, http://www.inria.fr/rrrt/rr-5123.html.

[53] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Independent and Divisible Task Scheduling on Heterogeneous Star-shaped Platforms with Limited Memory*, Also available as INRIA Research Report RR-5196, Research Report, LIP, ENS Lyon, France, April 2004, http://www.inria.fr/rrrt/rr-5196.html.

[54] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Steady-State Scheduling on Heterogeneous Clusters: Why and How?*, Research Report, n^o 2004-11, LIP, ENS Lyon, France, March 2004.

[55] O. BEAUMONT, L. MARCHAL. *Pipelining broadcasts on heterogeneous platforms under the one-port model*, Research Report, n^o RR-2004-32, LIP, ENS Lyon, France, July 2004.

[56] O. BEAUMONT, L. MARCHAL, Y. ROBERT. *Broadcast Trees for Heterogeneous Platforms*, Research Report, n^o RR-2004-46, LIP, ENS Lyon, France, November 2004.

[57] E. CARON, P.-K. CHOUHAN, F. DESPREZ. *Deadline scheduling with Priority for client-server systems*, Also available as INRIA Research Report RR-5335, Research report, n^o 2004-33, Laboratoire de l'Informatique du Parallélisme (LIP), July 2004, http://www.inria.fr/rrrt/rr-5335.html.

[58] E. CARON, P.-K. CHOUHAN, A. LEGRAND. *Automatic Deployment for Hierarchical Network Enabled Server*, Also available as LIP Research Report 2003-51, Research report, n^o RR-5146, Institut National de Recherche en Informatique et en Automatique (INRIA), March 2004, http://www.inria.fr/rrrt/rr-5146.html.

[59] E. CARON, F. DESPREZ, F. PETIT, C. TEDESCHI. *Resource Localization Using Peer-To-Peer Technology for Network Enabled Servers*, Research report, n^o 2004-55, Laboratoire de l'Informatique du Parallélisme (LIP), December 2004.

[60] H. DAIL, E. CARON. *GoDIET: Un outil pour le déploiement de DIET*, Technical report, n^o 2004-49, Laboratoire de l'Informatique du Parallélisme (LIP), November 2004, http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2004/RR2004-49.ps.gz.

[61] A. GIERSCH, Y. ROBERT, F. VIVIEN. *Scheduling tasks sharing files from distributed repositories (revised version)*, Also available as LIP, ENS Lyon, research report 2004-04, Research Report, n^o 5124, INRIA, February 2004, http://www.inria.fr/rrrt/rr-5124.html.

[62] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Coherent Load Information Mechanisms for Distributed Dynamic Scheduling*, Also LIP report RR2004-25, Research report, n^o RR-5178, INRIA, May 2004, http://www.inria.fr/rrrt/rr-5178.html.

[63] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Memory-based scheduling for a Parallel Multifrontal Solver*, Also LIP report RR2004-17, Research report, n^o RR-5162, INRIA, April 2004, http://www.inria.fr/rrrt/rr-5162.html.

[64] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Optimal Memory Minimization Algorithms for the Multifrontal Method*, Submitted to ACM Transactions on Mathematical Software, Research report, n° RR-5179, INRIA, 2004, http://www.inria.fr/rrrt/rr-5179.html.

[65] A. LEGRAND, A. SU, F. VIVIEN. *Off-line scheduling of divisible requests on an heterogeneous collection of databanks*, Also available as LIP, ENS Lyon, research report 2004-51, Research report, n° 5386, INRIA, November 2004, http://www.inria.fr/rrrt/rr-5386.html.

[66] L. MARCHAL, Y. YANG, H. CASANOVA, Y. ROBERT. *A realistic network/application model for scheduling divisible loads on large-scale platforms*, Also available as INRIA Research Report RR-5197, Research Report, LIP, ENS Lyon, France, April 2004, http://www.inria.fr/rrrt/rr-5197.html.

[67] H. RENARD, Y. ROBERT, F. VIVIEN. *Data redistribution algorithms for homogeneous and heterogeneous processor rings*, Also available as LIP, ENS Lyon, research report 2004-28, Research Report, n° 5207, INRIA, May 2004, http://www.inria.fr/rrrt/rr-5207.html.

Miscellaneous

[68] P. AMESTOY, I. DUFF, L. GIRAUD, J.-Y. L'EXCELLENT, C. PUGLISI. *GRID-TLSE: A Web Site for Experimenting with Sparse Direct Solvers on a Computational Grid*, SIAM Conference on Parallel Processing for Scientific Computing, San Francisco, California, February 2004.

[69] P. R. AMESTOY, A. GUERMOUCHE, J.-Y. L'EXCELLENT, S. PRALET. *Hybrid scheduling strategies for the parallel multifrontal method*, 3rd International Workshop on Parallel Matrix Algorithms and Applications (PMAA'04), Marseille, France, October 2004.

[70] R. BOLZE, E. CARON, P. COMBES, H. DAIL, C. PERA. *DIET Tutorial*, 21-25 June 2004, Ecole thématique sur la Globalisation des Ressources Informatiques et des Données : Utilisation et Services. GridUSe 2004.

[71] E. CARON, F. DESPREZ. *DIET, tour d'horizonEcole thématique sur la Globalisation des Ressources Informatiques et des Données : Utilisation et Services. GridUSe 2004, Metz. France*, 21-25 June 2004.

[72] E. CARON, F. DESPREZ, B. DEL-FABBRO, A. VERNOIS. *Gestion de données dans les NESDistRibUtIon de Données à grande Echelle. DRUIDE 2004, Domaine du Port-aux-Rocs, Le Croisic. France*, May 2004.

[73] B. DEL-FABBRO, D. LAIYMANI, J.-M. NICOD, L. PHILIPPE. *Gestion des données dans une plate-forme de métacomputingEcole thématique sur la Globalisation des Ressources Informatiques et des Données : Utilisation et Services. GridUSe 2004, Metz. France*, 21-25 June 2004.

[74] A. GUERMOUCHE, O. COZETTE, G. UTARD. *Study of the paging activity of the parallel multifrontal method*, 3rd International Workshop on Parallel Matrix Algorithms and Applications (PMAA'04), Marseille, France, October 2004.

[75] A. GUERMOUCHE, J.-Y. L'EXCELLENT, G. UTARD. *Some Memory Issues in the Multifrontal Method*, SIAM Conference on Parallel Processing for Scientific Computing (PP04), San Francisco, California, February 2004.

[76] L. PHILIPPE. *GridRPC : normalisation des API d'accès aux applications de grillesEcole thématique sur la*

Globalisation des Ressources Informatiques et des Données : Utilisation et Services. GridUSe 2004, Metz. France, 21-25 June 2004.

Bibliography in notes

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

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

[79] *CORBA 3.0 - Fault Tolerant chapter*, http://www.omg.org/cgi-bin/doc?formal/02-06-27.

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

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

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

[83] 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.

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

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

[86] G. BOSILCA, A. BOUTEILLER, F. CAPPELLO, S. DJAILALI, G. FEDAK, C. GERMAIN, P. HERAULT, O. LODYGENSKY, F. MAGNIETTE, V. NERI, A. SELIKHOV. *MPICH-V: Toward a Scalable Fault Tolerant MPI for Volatile Nodes*, in "Supercomputing'2002", 2002.

[87] H. CASANOVA, A. LEGRAND, L. MARCHAL. *Scheduling Distributed Applications: the SimGrid Simulation Framework*, in "Proceedings of the third IEEE International Symposium on Cluster Computing and the Grid (CCGrid'03)", May 2003.

[88] K. M. CHANDY, L. LAMPORT. *Distributed snapshots: Determining global states of distributed systems*, in "ACM Transactions on Computer Systems", vol. 3(1), 1985, p. 63–75.

[89] 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.

[90] 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.

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

[92] 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.

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

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

[95] R. GRAHAM, E. CHOI, D. DANIEL, N. DESAI, R. MINNICH, C. RASMUSSEN, L. RISINGER, M. SUKALSKI. *A Network-Failure-Tolerance Message-Passing System For Terascale Clusters*, in "ICS'02, New York, USA", ACM, June 2002.

[96] E. JEANNOT, B. KNUTSSON, M. BJORKMANN. *Adaptive Online Data Compression*, in "High Performance Distributed Computing (HPDC'11), Edinburgh, Scotland", IEEE, july 2002.

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

[98] J. KOHL, P. PAPADOPOULOS. *Efficient and Flexible Fault Tolerance and Migration of Scientific Simulations Using CUMULVS*, in "2nd SIGMETRICS Symposium on Parallel and Distributed Tools, Welches, OR", August 1998.

[99] J. W. H. LIU. *On the storage requirement in the out-of-core multifrontal method for sparse factorization*, in "ACM Transactions on Mathematical Software", vol. 12, 1986, p. 127-148.

[100] 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.

[101] 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.

[102] H. NAKADA, S. MATSUOKA, K. SEYMOUR, J. DONGARRA, C. LEE, H. CASANOVA. *GridRPC: A Remote Procedure Call API for Grid Computing*, in "Grid 2002, Workshop on Grid Computing, Baltimore, MD, USA", Lecture Notes in Computer Science, n° 2536, November 2002, p. 274-278.

[103] H. NAKADA, M. SATO, S. SEKIGUCHI. *Design and Implementations of Ninf: towards a Global Computing Infrastructure*, in "Future Generation Computing Systems, Metacomputing Issue", vol. 15, n° 5-6, 1999, p. 649-658.

[104] 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.

[105] 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.

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

[107] 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.

[108] 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.