INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team GRAAL

Algorithms and Scheduling for Distributed Heterogeneous Platforms

Rhône-Alpes

THEME NUM

Table of contents

1. Team

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

Head of team (until June 30, 2006)
Frédéric Desprez [Research Director (DR) Inria, HdR]

Head of team (since July 1, 2006)
Frédéric Vivien [Research Associate (CR) Inria]

Administrative assistants
Sylvie Boyer [30% on the project]
Isabelle Pera [25% on the project]

INRIA staff
Jean-Yves L'Excellent [Research Associate (CR)]
Frédéric Vivien [Research Associate (CR)]

Faculty members from ENS Lyon
Anne Benoît [Assistant Professor (MdC)]
Aurélien Bouteiller [Lecturer (ATER), until September 30, 2006]
Eddy Caron [Assistant Professor (MdC)]
Yves Robert [Professor, HdR]

Faculty members from Université Lyon 1 - UCBL
Yves Caniou [Assistant Professor (MdC)]

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

Project technical staff
Éric Boix [CNRS, 50% on the project]
Aurélien Ceyden [ENS Lyon, 50% on the project, since April 16, 2006]
Aurélia Fèvre [INRIA]
David Loureiro [INRIA, since October 1, 2006]
Nicolas Bard [CNRS, since December 1, 2006]

Post-doctoral fellows
Abdelkader Amar [until December 25, 2006]
Lionel Eyraud-Dubois [since October 17, 2006]

Ph. D. students (ENS Lyon)
Emmanuel Agullo [MENRT grant]
Raphaël Bolze [BDI CNRS]
Pushpinder-Kaur Chouhan [MENRT grant, until October 3, 2006]
Matthieu Gallet [ENS Grant, since September 1, 2006]
Jean-Sébastien Gay [Rhône-Alpes region grant]
Loris Marchal [ENS grant]
Jean-François Pineau [ENS grant]
Veronika Rehn [MENRT grant, since October 1, 2006]
Cédric Tedeschi [MENRT grant]

Ph. D. students (Univ. Franche Comté)
Sylvain Dahan [Rhône-Alpes région grant, until July 10, 2006]
Sékou Diakité [MENRT grant, since September 2006]
Bruno Del Fabbro [FAF grant, until July 31, 2006]
Suphakit Niwattanakul [Thailand grant, until August 31, 2006]

Hala Sabbah [Lecturer Liban University]

2. Overall Objectives

2.1. Overall Objectives

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

Parallel computing has spread into all fields of applications, from classical simulation of mechanical systems or weather forecast to databases, video-on-demand servers or search tools like Google. From the architectural point of view, parallel machines have evolved from large homogeneous machines to clusters of PCs (with sometime boards of several processors sharing a common memory, these boards being connected by high speed networks like Myrinet). However the need of computing or storage resources has continued to grow leading to the need of resource aggregation through Local Area Networks (LAN) or even Wide Area Networks (WAN). The recent progress of network technology has made it possible to use highly distributed platforms as a single parallel resource. This has been called Metacomputing or more recently Grid Computing [74]. An enormous amount of financing has recently been put on this important subject, leading to an exponential growth of the number of projects, most of them focusing on low level software detail. We believe that many of these projects failed to study fundamental problems such as problems and algorithms complexity, and scheduling heuristics. Also they usually have not validated their theoretical results on available software platforms.

From the architectural point of view, Grid Computing has different scales but is always highly heterogeneous and hierarchical. At a very large scale, thousands of PCs connected through the Internet are aggregated to solve very large applications. This form of the Grid, usually called a Peer-to-Peer (P2P) system, has several incarnations, such as SETI@home, Gnutella or XtremWeb [86]. 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 Grid'5000 project, which connects PC clusters spread in nine French university research centers. Many such projects exist over the world that connect a small set of machines through a fast network. Finally, at a research laboratory level, one can build an heterogeneous platform by connecting several clusters using a fast network such as Myrinet.

The common problem of all these platforms is not the hardware (these machines are already connected to the Internet) but the software (from the operating system to the algorithmic design). Indeed, the computers connected are usually highly heterogeneous (from clusters of 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 [80], [72], [74]. Today's environments are based either on the adaptation of "classical" parallel environments or on the development of toolboxes based on Web Services.

Aims of the GRAAL project.

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

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

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.

The main keywords of the GRAAL project:

Algorithmic Design + Middleware/Libraries + Applications

over Heterogeneous Architectures and the Grid

3. Scientific Foundations

3.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Anne Benoît, Lionel Eyraud-Dubois, Loris Marchal, Jean-François Pineau, Hélène Renard, Yves Robert, 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 [73], [85], [94], [96] and the references therein. This model was introduced for homogeneous processors, and has been (straightforwardly) extended to heterogeneous computing resources. In a word, there is a limited number of computing resources, or processors, to execute the tasks. Communication delays are taken into account as follows: let task T be a predecessor of task T' in the task graph; if both tasks are assigned to the same processor, no communication overhead is incurred, the execution of T' can start immediately at the end of the execution of T; on the contrary, if T and T' are assigned to two different processors P_i and P_j, a communication delay is incurred. More precisely, if P_i completes the execution of T at time-step t, then P_j cannot start the execution of T' before time-step $t + \text{comm}(T,T', P_i, P_j)$, where $\text{comm}(T,T', P_i, P_j)$ is the communication delay, which depends upon both tasks T and T' and both processors P_i and P_j. Because memory accesses are typically several orders of magnitude cheaper than inter-processor communications, it is sensible to neglect them when T and T' are assigned to the same processor.

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

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

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

3.2. Scheduling for Sparse Direct Solvers

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

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

Because of their efficiency and robustness, direct methods (based on Gaussian factorization) are methods of choice to solve these types of problems. In this context, we are particularly interested in the multifrontal method [83], [84], for symmetric positive definite, general symmetric or unsymmetric problems, with numerical pivoting in order to ensure numerical stability. Note that numerical pivoting induces dynamic data structures that are unpredictable symbolically or from a static analysis.

The multifrontal method is based on an elimination tree [90] which results (i) from the graph structure corresponding to the nonzero pattern of the problem to be solved, and (ii) from the order in which variables are eliminated. This tree provides the dependency graph of the computations and is exploited to define tasks that may be executed in parallel. In this method, each node of the tree corresponds to a task (itself potentially parallel) that consists in the partial factorization of a dense matrix. This approach allows for a good locality and usage of cache memories.

In order to deal with numerical pivoting and keep an approach that can adapt to as many computer architectures as we can, we are especially interested in approaches that are intrinsically dynamic and asynchronous [77], [78]. In addition to their numerical robustness, the algorithms retained are based on a dynamic and distributed management of the computational tasks, not so far from today's peer-to-peer approaches: each process is responsible for providing work to some other processes and at the same time acting as a slave for others. These algorithms are very interesting from the point of view of parallelism and in particular for the study of mapping and scheduling strategies for the following reasons:

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

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

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

3.3. Providing Access to HPC Servers on the Grid

Participants: Abdelkader Amar, Raphaël Bolze, Aurélien Bouteiller, Yves Caniou, Eddy Caron, Aurélien Ceyden, Pushpinder-Kaur Chouhan, Sylvain Dahan, Bruno Del Fabbro, Frédéric Desprez, Jean-Sébastien Gay, David Loureiro, Nicolas Bard, Jean-Marc Nicod, Laurent Philippe, Antoine Vernois, Frédéric Vivien.

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

A first approach provides the user with a uniform view of resources. This is the case of GLOBUS [1] which provides transparent MPI communications (with MPICH-G2) between distant nodes but does not manage load balancing issues between these nodes. It is the user's task to develop a code that will take into account the heterogeneity of the target architecture. Classical batch processing can also be used on the Grid with projects like Condor-G [2] or Sun GridEngine [3]. Finally, peer-to-peer [75] or Global computing [88] 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 [91], [92]) offers an easy access to available resources to a Web browser, a Problem Solving Environment, or a simple client program written in C, Fortran, or Java. It also provides more transparency by hiding the search and allocation of computing resources. We favor this second approach.

In a Grid context, this approach requires the implementation of middleware environments to facilitate the client access to remote resources. In the ASP approach, a common way for clients to ask for resources to solve their problem is to submit a request to the middleware. The middleware will find the most appropriate server that will solve the problem on behalf of the client using a specific software. Several environments, usually called Network Enabled Servers (NES), have developed such a paradigm: NetSolve [79], Ninf [93], NEOS [87], OmniRPC [95], and more recently DIET developed in the GRAAL project (see Section 5.1). A common feature of these environments is that they are built on top of five components: clients, servers, databases, monitors and schedulers. Clients solve computational requests on servers found by the NES. The NES schedules the requests on the different servers using performance information obtained by monitors and stored in a database.

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

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

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

4. Application Domains

4.1. Applications of Sparse Direct Solvers

Our activity on sparse direct (multifrontal) solvers in distributed environments goes as far as building competitive software available to users. Such methods have a wide range of applications and they are at the heart of many numerical methods in simulation: whether a model uses finite elements or differences, or requires the optimization of a complex linear or nonlinear function, one almost always ends up in solving a system of equations involving sparse matrices. There are therefore a number of application fields, among which we can list the most frequently cited by our users, i.e. the applications in which our sparse direct solver MUMPS (see Section 5.2) has been or is currently used: structural mechanical engineering (stress analysis, structural optimization, car bodies, ships, crankshaft segment, offshore platforms, Computer Assisted Design, Computer Assisted Engineering, rigidity of sphere packings), heat transfer analysis, thermomechanics in casting simulation, fracture mechanics, biomechanics, medical image processing, tomography, plasma physics (e.g., Maxwell's equations), critical physical phenomena, geophysics (e.g., seismic wave propagation, earthquake related problems), ad-hoc networking modeling (Markovian processes), modeling of the magnetic field inside machines, econometric models, soil-structure interaction problems, oil reservoir simulation, computational fluid dynamics (e.g., Navier-stokes, ocean/atmospheric modeling with mixed Finite Elements Methods, fluvial hydrodynamics, viscoelastic flows), electromagnetics, magneto-hydro-dynamics, modeling the structure of the optic nerve head and of cancellous bone, modeling and simulation of crystal growth processes, chemistry (chemical process modeling), vibro-acoustics, aero-acoustics, aero-elasticity, optical fiber modal analysis, blast furnace modeling, glaciology (models of ice flow), optimization, optimal control theory, astrophysics (e.g., supernova, thermonuclear reaction networks, neutron diffusion equation, quantum chaos, quantum transport), research on domain decomposition (MUMPS can for example be used on each subdomain in an iterative framework), circuit simulations, etc.

4.2. Molecular Dynamics

LAMMPS is a classical molecular dynamics (MD) code created for simulating molecular and atomic systems such as proteins in solution, liquid-crystals, polymers, zeolites, or simple Lenard-Jonesium. It was designed for distributed-memory parallel computers and runs on any parallel platform that supports the MPI message-passing library or on single-processor workstations. 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 for satellite imagery which produces 6000 $\times$ 6000-pixel images, involving important computation times as well as very large memory demand. We used the DIET architecture to solve this problem in collaboration with the Earth Science Laboratory (LST ENS Lyon).

4.4. Electronic Device Simulation

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

4.5. Biochemistry

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

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

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

4.6. Bioinformatics

Genomics acquiring programs, such as full genomes sequencing projects, are producing larger and larger amounts of data. The analysis of these raw biological data require very large computing resources. Functional sites and signatures of proteins are very useful for analyzing these data or for correlating different kinds of existing biological data. These methods are applied, for example, to the identification and characterization of the potential functions of new sequenced proteins, and to the clusterization into protein families of the sequences contained in international databanks.

The sites and signatures of proteins can be expressed by using the syntax defined by the PROSITE databank, and written as a "protein regular expression". Searching one such site in a sequence can be done with the criterion of the identity between the searched and the found patterns. Most of the time, this kind of analysis is quite fast. However, in order to identify non perfectly matching but biologically relevant sites, the user can accept a certain level of error between the searched and the matching patterns. Such an analysis can be very resource consuming.

In some cases, due to the lack of sufficient computing and storage resources, skilled staff or technical abilities, laboratories cannot afford such huge analyses. Grid computing may be a viable solution to the needs of the genomics 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.

This work continues the previous work of F. Desprez and A. Vernois about simultaneous scheduling of jobs and data replication for life science applications on the grid. From this work they designed a scheduling strategy based on the hypothesis that, as you choose a large enough time interval, the proportion of a job using a given data is always the same. As observed in execution traces of bioinformatics clusters, this hypothesis seems to correspond to the way that these clusters are generally used. However, this algorithm does not take into account the initial data distribution costs and, in its original version, the dynamicity of the submitted jobs proportions. We introduce algorithms that allow to get good performance as soon as the process starts and take care about the data redistribution when needed. We want to run a continuous stream of jobs, using linear in time algorithms that depend on the size of the data on which they are applied. Each job is submitted to a Resource Broker which chooses a Computing Element (CE) to queue the job on it. When a job is queued on a CE, it waits for the next worker node that can execute it, with a FIFO policy. The objective is to ensure the better throughput of the platform, that is, in the context of massive submission of short time jobs, to minimize the waiting time of the jobs in the CE's queues.

4.7. Cosmological Simulations

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

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

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

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

4.8. Decrypthon

The decrypthon program is a joined program between AFM, IBM and the CNRS. The main goal is to accelerate biological research by providing grid computing power to researchers who need huge computing power. In this program, we continue to have a key position, by working on making applications grid enabled. The supported applications were selected by the scientific committee of the decrypthon program. We are in position to bring our expertise to other bioinformatics teams in order to tune their applications into the decrypthon grid. We have collaborated with 6 research teams and 5 computing centers to set up and maintain the decrypthon grid. An example of these collaborations is described in more details in [26]. We also collaborate and provide support to the World Community Grid staff in order to integrate the Help Cure Muscular Dystrophy project into the world wild Desktop grid of the World Community Grid. This project has been launched in December 2006.

5. Software

5.1. DIET

Participants: Abdelkader Amar, Eric Boix, Raphaël Bolze, Aurélien Bouteiller, Yves Caniou, Eddy Caron, Pushpinder-Kaur Chouhan, Sylvain Dahan, Frédéric Desprez [correspondent], Bruno Del Fabbro, Jean-Sébastien Gay, Jean-Marc Nicod, Laurent Philippe, Cédric Tedeschi.

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) [97] sensors are placed on each node of the hierarchy to collect resource availabilities, which are used by an application-centric performance prediction tool named FAST (see below).

The different components of our scheduling architecture are the following. A **Client** is an application which uses DIET to solve problems. Many kinds of clients should be able to connect to DIET from a web page, a Problem Solving Environment such as Matlab or Scilab, or a compiled program. A **Master Agent (MA)** receives computation requests from clients. These requests refer to some DIET problems listed on a reference web page. Then the MA collects computational abilities from the servers and chooses the best one. The reference of the chosen server is returned to the client. A client can be connected to an MA by a specific name

server or a web page which stores the various MA locations. Several MAs can be deployed on the network to balance the load among the clients. A **Local Agent (LA)** aims at transmitting requests and information between MAs and servers. The information stored on a LA is the list of requests and, for each of its subtrees, the number of servers that can solve a given problem and information about the data distributed in this subtree. Depending on the underlying network topology, a hierarchy of LAs may be deployed between an MA and the servers. No scheduling decision is made by a LA. A **Server Daemon (SeD)** encapsulates a computational server. For instance it can be located on the entry point of a parallel computer. The information stored on a SeD is a list of the data available on its server (with their distribution and the way to access them), the list of problems that can be solved on it, and all information concerning its load (available memory and resources, etc). A SeD declares the problems it can solve to its parent LA. A SeD can give performance prediction for a given problem thanks to the CoRI module (Collector of Resource Information) [39].

Moreover applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via *plug-in schedulers* [39]. As the applications that are to be deployed on the grid vary greatly in terms of performance demands, the DIET plug-in scheduler facility permits the application designer to express application needs and features in order that they be taken into account when application tasks are scheduled. These features are invoked at runtime after a user has submitted a service request to the MA, which broadcasts the request to its agent hierarchy.

Master Agents can then be connected over the net (Multi-MA version of DIET), either statically of dynamically [23].

From collaboration between GRAAL project and PARIS project DIET can use *JuxMem* [16]. *JuxMem* (Juxtaposed Memory) is a peer-to-peer architecture developed by PARIS team which provides memory sharing service allowing peers to share memory data, and not only files. To illustrate how a *GridRPC* system can benefit from transparent access to data, we have implemented the proposed approach inside the DIET *GridRPC* middleware, using the *JuxMem* data-sharing service [58].

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) [37].

Seen from the user/developer point of view, the compiling and installation process of DIET should remain simple and robust. But DIET has to support this process for an increasing number of platforms (Hardware architecture, Operating System, C/C++ compilers). Additionally DIET also supports many functional extensions (sometimes concurrent) and many such extensions require the usage of one or a few external libraries. Thus the compilation and installation functionalities of DIET must handle a great number and variety of possible specific configurations. Up to the previous versions, DIET's privileged tool for such a task were the so-called GNU-autotools. DIET's autotools configuration files evolved to become fairly complicated and hard to maintain. Another important task of the packager-person of DIET is to assess that DIET can be properly compiled and installed at least for the most mainstream platforms and for a decent majority of all extension combinations. This quality assertion process should be realized with at least the frequency of the release. But, as clearly stated by the agile software development framework, the risk can be greatly reduced by developing software in short time-boxes (as short as a single cvs commit). For the above reasons, it was thus decided to move away from the GNU-autotools to cmake (refer http://www.cmake.org). Cmake offers a much simpler syntax for its configuration files (sometimes at the cost of semantics, but cmake remains an effective trade-off). Additionally, cmake integrates a scriptable regression test tool whose reports can be centralized on a so called dashboard server. The dashboard offers a synthetic view (see http://graal.ens-lyon.fr:8081/DIETcore/Dashboard/) of the current state of DIET's code. This quality evaluation is partial (compilation and linking errors and warnings) but is automatically and constantly offered to the developers. Although the very nature of DIET makes it difficult to carry distributed regression tests, we still hope that the adoption of cmake will significantly improve DIET's robustness and general quality.

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

5.1.1. Workflow support

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

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

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

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

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

5.2. MUMPS

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

MUMPS (for *MUltifrontal Massively Parallel Solver*, see http://graal.ens-lyon.fr/MUMPS) is a software package for the solution of large sparse systems of linear equations. The development of MUMPS was initiated by the European project PARASOL (Esprit 4, LTR project 20160, 1996-1999), whose results and developments were public domain. Since then, mainly in collaboration with ENSEEIHT-IRIT (Toulouse, France), lots of developments have been done, to enhance the software with more functionalities and integrate recent research work. Recent developments also involve the INRIA project ScAlAppliX, since the recruitment of Abdou Guermouche as an assistant professor at LaBRI, while CERFACS contributes to some research work.

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

- various types of systems: symmetric positive definite, general symmetric, or unsymmetric,
- several matrix input formats: assembled or expressed as a sum of elemental matrices, centralized on one processor or pre-distributed on the processors,
- partial factorization and Schur complement matrix,
- real or complex arithmetic, single or double precision,
- partial pivoting with threshold,
- fully asynchronous approach with overlap of computation and communication,
- distributed dynamic scheduling of the computational tasks to allow for a good load balance in the presence of unexpected dynamic pivoting or in multi-user environments.

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

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

From a geographical point of view, 31% of our users (approx. 1000) come from North America, 39% are Europeans, and 19% are from Asia.

The latest release is MUMPS 4.6.3, available since June 2006 (see http://graal.ens-lyon.fr/MUMPS/avail.html). It incorporates most recent features, including Matlab and Scilab interfaces and better numerical processing of symmetric indefinite matrices.

A preliminary (beta) version of the out-of-core solver has also been made available to a limited number of users, with whom we have strong collaborations: EADS, Free Field Technologies, and Samtech.

Note that a MUMPS Users' Day was organized on Tuesday 24 October 2006. This workshop gathered 30 participants, mostly from France. In addition to presentations of applications using MUMPS and presentations from the MUMPS developers, the goal was also for users to discuss their needs for the future. The discussions have been fruitful, and thanks to that day, and to technical meetings the same week, we now have a better view of the requirements of the most challenging applications relying on MUMPS. We also have a better knowledge of the research and developments that we need to carry out to satisfy these requirements. The schedule of the day as well as some of the presentations are available from the MUMPS web site.

6. New Results

6.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

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

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

Finally, we point out that we have given an invited talk on steady-state scheduling techniques at IPDPS'2006, thereby exposing (immortalizing?) the approach to a larger audience.

6.1.2. Scheduling and data redistribution strategies

In this work we are interested in the problem of scheduling and redistributing data on master-slave platforms. We consider the case where the workers possess initial loads, some of which having to be redistributed in order to balance their completion times. We examine two different scenarios. The first model assumes that the data consists of independent and identical tasks. We prove the NP-completeness in the strong sense for the general case, and we present two optimal algorithms for special platform types. Furthermore we propose three heuristics for the general case. Simulations consolidate the theoretical results. The second data model is based on Divisible Load Theory. This problem can be solved in polynomial time by a combination of linear programming and simple analytical manipulations.

The main originality of this work it to combine data redistribution and current computations. This is a very realistic but difficult scenario, and much work remains to extend it to more general platforms. For instance we could target a more decentralized environment with direct exchanges between participating workers.

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

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

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

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

Limited memory. Because we investigate the parallelization of large problems, we cannot assume that full matrix panels can be stored in the worker memories and re-used for subsequent updates (as in ScaLAPACK). The amount of memory available in each worker is expressed as a given number m_i of buffers, where a buffer can store a square block of matrix elements. The size q of these square

blocks is chosen so as to harness the power of Level 3 BLAS routines: $q = 80$ or 100 on most platforms.

We have devised efficient algorithms for resource selection (deciding which workers to enroll) and communication ordering (both for input and result messages), and we report a set of numerical experiments on various platforms at École Normale Supérieure de Lyon and the University of Tennessee.

We plan to extend this work with the study of various other algorithmic kernels, such as LU decomposition.

6.1.4. Replica placement

We have introduced and compared several policies to place replicas in tree networks, subject to server capacity and QoS constraints. In this framework, the flows of client requests are known beforehand, while the number and location of the servers are to be determined. The standard approach in the literature is to enforce that all requests of a client be served by the closest server in the tree. We introduce and study two new policies. In the first policy, all requests from a given client are still processed by the same server, but this server can be located anywhere in the path from the client to the root. In the second policy, the requests of a given client can be processed by multiple servers.

One major contribution of our work is to assess the impact of these new policies on the total replication cost. Another important goal is to assess the impact of server heterogeneity, both from a theoretical and a practical perspective. We establish several new complexity results, and provide several efficient polynomial heuristics for NP-complete instances of the problem. These heuristics are compared to an absolute lower bound provided by the formulation of the problem in terms of the solution of an integer linear program.

6.1.5. Mapping pipeline skeletons

Mapping applications onto parallel platforms is a challenging problem, that becomes even more difficult when platforms are heterogeneous —nowadays a standard assumption. A high-level approach to parallel programming not only eases the application developer's task, but it also provides additional information which can help realize an efficient mapping of the application.

This year, we have discussed the mapping of pipeline skeletons onto different types of platforms: *Fully Homogeneous* platforms with identical processors and interconnection links; *Communication Homogeneous* platforms, with identical links but processors of different speeds; and finally, *Fully Heterogeneous* platforms. We assume that a pipeline stage must be mapped on a single processor, and we establish new theoretical complexity results for different mapping policies: the mapping can be required to be one-to-one (a processor is assigned at most one stage), or interval-based (a processor is assigned an interval of consecutive stages), or fully general. We provide several efficient polynomial heuristics for interval-based mappings on *Communication-Homogeneous* platforms.

6.1.6. Scheduling communication requests

As a follow-up of our work on network resource sharing (in collaboration with the RESO team), we have investigated the problem of scheduling file transfers through a switch. This problem is at the heart of a model often used for large grid computations, where the switch represents the core of the network interconnecting the various clusters that compose the grid. We establish several complexity results, and we introduce and analyze various algorithms, from both a theoretical and a practical perspective.

6.1.7. VoroNet

In this work, we propose the design of VoroNet, an object-based peer to peer overlay network relying on Voronoi tessellations, along with its theoretical analysis and experimental evaluation. VoroNet differs from previous overlay networks in that peers are application objects themselves and get identifiers reflecting the semantics of the application instead of relying on hashing functions. Thus it provides a scalable support for efficient search in large collections of data. In VoroNet, objects are organized in an attribute space according to a Voronoi diagram. VoroNet is inspired from the Kleinberg's small-world model where each peer gets connected to close neighbors and maintains an additional pointer to a long-range node. VoroNet improves

upon the original proposal as it deals with general object topologies and therefore copes with skewed data distributions. We show that VoroNet can be built and maintained in a fully decentralized way. The theoretical analysis of the system proves that the routing in VoroNet can be achieved in a poly-logarithmic number of hops in the size of the system. The analysis is fully confirmed by our experimental evaluation by simulation.

6.1.8. Online scheduling of divisible requests

We have completed our study of the scheduling of comparisons of motifs against biological databanks. We had previously shown that this problem can be expressed within the divisible load framework. Still in the context of the minimization of the maximum stretch or, more generally, of the maximum weighted flow, we have presented some heuristics to Pareto optimally minimize the maximum weighted flow. We have shown that this problem is NP-complete in the general case, but that in some cases our heuristics build optimal schedules.

6.1.9. Parallelizing the construction of the ProDom database

ProDom is a database of protein domain families which is automatically generated from the SWISS-PROT and TrEMBL sequence databases. The size of the sequence databases is growing exponentially and building a new version of ProDom now requires more than six months on a mono-processor computer as MKDom, the algorithm used to build ProDOM, is a sequential algorithm. We began to study the properties of this algorithm and of the biological problem, in order to efficiently design a parallel algorithm.

This work is done in collaboration with the INRIA project-team Helix.

6.1.10. Automatic discovery of platform topologies

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

In the scope of this work we have contributed to the software ALNeM which is a framework to perform network measures and modelling, and which can also be used to perform simulations. In the same framework, we can therefore build a model and assess its quality. So far, we have designed and implemented some simple algorithms (mostly spanning-tree based) and we have started assessing their quality. From the results of these preliminary experiments, we will design more sophisticated modeling algorithms.

6.2. Providing access to HPC servers on the Grid

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

Participants: Raphaël Bolze, Aurélien Bouteiller, Yves Caniou, Eddy Caron, Pushpinder-Kaur Chouhan, Sylvain Dahan, Frédéric Desprez, Bruno Del Fabbro, Jean-Sébastien Gay, Jean-Marc Nicod, Laurent Philippe, Cédric Tedeschi, Frédéric Vivien.

6.2.1. Workflow scheduling

The problem of scheduling one workflow (DAG)[5] on an heterogeneous platform has been studied intensively during the last 10 years. Surprisingly the problem of scheduling multiple workflows does not appear to be fully addressed. We study many heuristics to solve this problem. We also implemented a simulator in order to classify the behaviors of these heuristics depending on the shape and size of the graphs. Some of these heuristics will be implemented within DIET and tested with the Bioinformatics applications involved in the Decrypthon program.

6.2.2. Large Scale Service Lookup

The first DIET prototype was based on an agent hierarchy with a Master Agent as a root. To further increase the overall scalability of the system, multiple 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 MAs until an agent finds the appropriate server. Two approaches have been used to implement this interconnection. One is based on the JXTA [89] Peer-to-peer platform and the other used CORBA communications to define an overlay network.

[5] Directed Acyclic Graph

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

For large scale networks, we propose a data structure which allows to distribute spanning trees among the nodes and links of the interconnection network. This structure has been called Distributed Spanning Tree (DST) as it provides a different spanning tree for each Master Agent. It decreases bottlenecks in the graph traversal algorithms. The structure proposed has been simulated and results show its efficiency in comparison with trees and random graphs: it limits the number of messages to find a server and distributes the load over the interconnection network. The DST may also be used to broadcast information to all the nodes of the network. In this case, it drastically increases the frequency of requests supported without saturation.

6.2.3. Service Discovery in Peer-to-Peer environment

The area studied is computational grids, peer-to-peer systems and their possible interactions to provide a large scale service discovery within computational grids. In order to address this issue, we first developed a new architecture called *DLPT (for Distributed Lexicographic Placement Table)*. This is a distributed indexing system based on a prefix tree that offers a totally distributed meaning of indexing and retrieving services at a large scale. The DLPT offers flexible search operations for users while offering techniques to replicate the structure for fault tolerance over dynamic platforms and a greedy algorithm partially taking into account the performance of the underlying network [42], [64].

One of the fundamental aspects of the peer-to-peer system is the fault-tolerance they provide. During a collaboration with Franck Petit from the LaRIA, we developed some fault-tolerance algorithms for peer-to-peer architecture structured as trees [41], [63].

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

6.2.4. Deployment for DIET: Software and Research

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

We worked on an automatic deployment solution for DIET [24]. The first step was to study the homogeneous case. Thus we provide an approach to determine an appropriate hierarchical middleware deployment for a homogeneous resource platform of a given size. The approach determines how many nodes should be used and in what hierarchical organization; the goal is to maximize steady-state throughput [44]. The model provides an optimal real-valued solution without resource constraints; we then apply round-up or round-down to obtain integer factors for the hierarchy definition. We also provide algorithms to modify the obtained hierarchy to limit the number of resources used to the available number. We instantiate the model for the hierarchical

scheduling system used in DIET. Our experiments on Grid'5000 validated the throughput performance model used for servers and agents and demonstrated that the automatic deployments performed well when compared to other intuitive deployments [40]. We plan to work on deployment planning and re-deployment algorithms for middleware on heterogeneous clusters and Grids.

6.2.5. Grid'5000 large scale experiments

In the current availability of Grid'5000 platform, the deployment of DIET works in three phases. The first step consists in sending one OAR request at each site, to reserve a maximum of available nodes. The second phase consists in receiving OAR information to know which nodes are given by reservation. The third phase generates an XML file with the dynamic information as well as names of nodes at each site. These files will be used by GoDIET to deploy DIET. Our main goal during this first experience is to corroborate a theoretical study of the deployment with the hardware capability of Grid'5000 platform (CPU performance, bandwidth, etc.) to design a hierarchy that achieves a good scalability and a good efficiency for DIET. From this XML file, GoDIET [40] deploys agents (or schedulers), servers and services bound to DIET as CORBA services (i.e. naming service) along with a distributed log tool designed for the visualization tools (VizDIET) [37]. In [19] we show a large deployment of DIET using 574 computing nodes and 9 agents for the scheduling of 45000 requests. The 574 servers are deployed on 8 clusters and 7 sites.

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 first proposed 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. Our approach uses a good knowledge of databank usage scheme and of the target platform. Starting with this information, we 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 validated the simulation results in the DIET environment. This prototype has been deployed on Grid'5000 for a large scale experiment.

We came to the conclusion that when the storage space available on the grid is not large enough to store all databanks that lead to very time consuming requests on all computation servers, then our approach increases the throughput of the platform. But this conclusion is effective only if the submitted requests follow precisely the usage frequencies given as an input for the static replication and scheduling algorithm. Due to particular biological experiments these schemes may punctually change. To cope with those changes, we developed a dynamic algorithm and a set of heuristics that monitor the execution platform and take decision to move data and change scheduling of requests. The main goal of this algorithm is to balance the computation load between each server. Again using the Optorsim simulator, we compared the results of the different heuristics. The conclusion of these simulations is that we have a set of heuristics that, in the case of our hypothesis, are able to reliably adapt the data placement and requests scheduling to get an efficient usage of all computation resources.

6.2.7. Parallel Job Submission Management

Actually, grids are built on a clusters hierarchy model, as used by the two projects EGEE [6] (*Enabling Grids for E-science in Europe*) and Grid'5000 (see Section 8.2.5). The production platform for the EGEE project aggregates more than one hundred sites spread over 31 countries. Grid'5000 is the French Grid for the research, which aims to own 5000 nodes spread over France (9 sites are currently participating).

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

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

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

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

We have performed several experiments, some with *Ramses* (see Section 4.7), and we plan to build a client/server for the LAMMPS software (see Section 4.2). We have undertaken some work to add performance prediction for parallel resources to DIET: communicate with batch system and simulating them with the Simbatch simulator that we have developed (see next section). Hence, we will have sufficient information to incorporate pertinent distributed scheduling algorithms into DIET.

6.2.8. Job Submission Simulations

Generally, the use of a parallel computing resource is done via a batch reservation system. The algorithms involved can greatly impact performance and consequently, be critical for the efficiency of grid computing. Unfortunately, few grid simulators take those batch reservation systems into account. They provide at best a very restricted modeling using an FCFS algorithm and few of them deals with parallel tasks. In this context we have proposed a reusable module, named Simbatch [7] [48], [68], as a built-in for the grid simulator Simgrid [8] allowing to easily model various batch schedulers.

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

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

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

6.2.9. Fault Tolerance

We presented three fault tolerant mechanisms adapted to the context of Network Enabled Servers environments. The first step to cope with failures is to detect them. Unlike much other environments, we do not rely on the network transport layer to detect failures but we introduce the first implementation of the Chandra & Toueg & Aguillera optimal failure detector in a Grid system. This is a heartbeat based failure detector, but unlike most widely used fault detection mechanisms it does not introduce dependency between the time to detect a failure and the relative speed of the previous and current heartbeat. As a consequence it allows to reach an optimal accuracy for a given heartbeat frequency. The experimental evaluation shows a very little overhead and a nearly perfect accuracy when observing very large number of processes distributed over a real Grid (Grid'5000).

Second we focused on recovering failures of the architecture. In DIET, the service discovery and scheduling architecture is distributed over a hierarchical set of agents. We introduced a distributed recovery algorithm ensuring eventual correct state of the architecture when up to $f-1$ failures occurs simultaneously, where f is the number of known ancestors. A correct state of the architecture is a state where all computing resources are available to the clients of the architecture. Experimental evaluation of this algorithm shows a large impact of failure detection time on the recovery performance. When multiple failures occur, failure detection time is added upon reconnection of a failed ancestor. A solution to this issue is to observe more than one ancestor during fault free operation to simultaneously detect multiple failures of ancestors. As long as all ancestors are not observed, it exists some worst case scenario where recovery depends on sequential detection time of many ancestors, but experiments show that in practice observing only a subset of the ancestors is sufficient as those worst cases are unlikely to happen. Moreover, we evaluated architecture recovery algorithm benefits on execution time of a typical BLAS-based numerical application. Most architecture recovery overhead is overlapped by application computation. Indeed overhead occurs only when a client submits new tasks while the system is still recovering. Compared to a non fault tolerant architecture, performance and efficiency is greatly improved as a small number of failures does not disconnect a large number of computing resources from the platform.

Finally we introduce a novel checkpoint interface to manage service recovery. This interface aims at allowing the use of both automatic and service provided checkpoint. A typical service in a GridRPC environment is a binding of a generic numerical library (such as BLAS or ScaLAPACK). Those libraries may provide their own fault tolerant management, based on explicit checkpointing. On the other hand automatic checkpointing is a key feature considering a user convenient environment. Our interface allows to automatically self checkpoint any sequential service. Parallel services may be automatically checkpointed using some third party software such as MPICH-V or LAM/MPI and using the same mechanisms as those described for service provided checkpoint. Once checkpoint data set has been build, it is stored in the *JuxMem* replicated shared memory manager, thus we are able to recover from the complete loss of the computing resource, should it be a parallel cluster.

6.3. Parallel Direct Solvers for Sparse Systems of Linear Equations

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

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

6.3.1. Extension and maintenance of the software package MUMPS

This year, we have continued our work to extend the MUMPS software package. In particular a preliminary out-of-core functionality, where computed factors are written to disk, has been made available to some of our users, in order to get their feedback before making this functionality more widely available. We have also been working on reducing the memory requirements for symmetric matrices (by using a packed format for temporary Schur complements) as well as for unsymmetric matrices and have modified the general memory management algorithms to allow for more flexibility in an out-of-core context. Performance of the factorization of symmetric matrices has been improved thanks to modifications of the algorithms (and block sizes) that use BLAS 3 kernels. The Scilab and Matlab interfaces to MUMPS have been stabilized and are now publically available in user releases.

Strong interactions with users have helped us to enhance the robustness of the recent functionalities (2x2 pivots, hybrid scheduling taking into account both memory and performance).

Part of this work was done in the context of the INRIA contract of Aurélia Fèvre ("Opération Développement Logiciel"). Furthermore, this ODL allowed to work more specifically on software issues: portability to various platforms including Grid'5000, tools for checking the performance of various functionalities from one release to the next, regression tests running each night, and code coverage with associated tests, in order to validate existing code and identify dead parts.

6.3.2. Pivoting for symmetric matrices in a parallel context

In collaboration with Stéphane Pralet (GRAAL visitor from November 2005 to February 2006, while he was postdoc at ENSEEIHT-IRIT), we have further improved the numerical behaviour of MUMPS for symmetric indefinite matrices. In our approach to parallelism, some of the columns are distributed over several processors. In that case, our pivoting strategy (partial pivoting with threshold) requires information about the magnitude of the elements in the column in order to find an acceptable pivot in the block that needs to be factored. In our parallel asynchronous approach, when electing a new pivot, it is not realistic to wait for a message from all processors involved in the column, as this would imply strong synchronizations and seriously limit the scalability of the solver. However, as a result of this lack of information on the magnitude of elements in the columns, our method could behave numerically better in the sequential case than in the parallel case (on numerically hard problems). In order to improve this behaviour on numerically hard problems, we have implemented a cheap estimation of growth factors based on [82]. These estimates are then used in the 1x1 and 2x2 pivot selections. Experiments on a set of ill-conditioned matrices confirm the simulations of a parallel behaviour done in [82]. This functionality is currently being experimented by our industrial collaborator Samtech and will be included in a future public release of MUMPS. We are not aware of any other solver for distributed-memory architectures that have similar functionalities for general LDL^T factorizations.

6.3.3. Parallel out-of-core factorization

The memory usage of sparse direct solvers can be the bottleneck to solve large-scale problems involving sparse systems of linear equations of the form Ax=b. If memory is not large enough to treat a given problem, disks must be used to store data that cannot fit in memory (*out-of-core* storage). In a previous work, we proposed a first out-of-core extension of a parallel multifrontal approach based on the solver MUMPS, where only the computed factors were written to disk during the factorization. This year we have studied in detail the minimum memory requirements of this parallel multifrontal approach and proposed several mechanisms to decrease further those memory requirements. For a given amount of memory, we have also studied the volume of disk accesses involved during the factorization of matrix A, providing insight on the extra cost that we can expect due to I/O. Furthermore, we have studied the impact of low-level I/O mechanisms, and have in particular shown the interest (and difficulty) of relying on direct I/O. Large-scale problems from applicative fields have been used to illustrate our discussions. This work is performed in the context of the PhD of Emmanuel Agullo, in collaboration with Abdou Guermouche (LaBRI and INRIA project ScAlAppliX) and Patrick Amestoy (ENSEEIHT-IRIT).

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

A collaboration with Xiaoye S. Li and Esmond G. Ng (Lawrence Berkeley National Laboratory, Berkeley, California, USA) was initiated with a first visit of Emmanuel Agullo (fifteen days) in September 2006. The goal is to compare the multifrontal factorization to the left-looking approach in an out-of-core context. A six-months visit is scheduled in 2007.

6.3.4. Parallel out-of-core solution phase

We collaborate with Tzetomila Slavova (Ph.D. at CERFACS) on the study of the performance of the out-of-core solution phases (forward and backward substitutions). In many applications, the solution phase can be invoked many times for a unique factorization phase. In an out-of-core context, the solution phase can thus become even more costly than the factorization. In a first approach, we can rely on system buffers (or pagecache) to access the disks. We have shown that this approach was not adapted because it cannot "choose" correctly data that must be kept in memory and because an unpredictable and often large cache memory is being used. Furthermore, it is important to really control the amount of memory effectively used (system buffers included). Therefore, a version with direct I/O has been introduced. We have shown that the performance was comparable with the system approach, with the advantage of effectively controlling the memory usage. In a multiprocessor environment we have also shown that the order in which the dependency tree is processed could have a very strong impact on performance, because of the irregularity of the disk accesses involved.

6.3.5. Hybrid Direct-Iterative Methods

We collaborate with Haidar Azzam (Ph.D., CERFACS) and Luc Giraud (ENSEEIHT-IRIT) on hybrid direct-iterative solvers. The substructuring methods developed in this context rely on the possibility to compute a partial factorization, with a so-called Schur complement matrix, that is typically computed by a direct solver such as MUMPS. The direct solver is called on each subdomain of a physical mesh, and the iterative approach takes care of the interface problem, based on Schur complements provided by our direct solver. We have been working on tuning this functionality and giving advice on how to best exploit the direct solver in the context of such iterative approaches. Thanks to this collaboration, we have also identified some critical points that should be addressed to optimize the memory usage when computing the Schur complement matrix: the approach will consist in increasing slightly the amount of computations, with the advantage of reducing the memory usage. As a consequence, larger subdomains will be possible, with a fixed amount of memory per subdomain.

6.3.6. Experimentation on real-life test problems

MUMPS users provide us with new challenging problems to solve and constantly help us validate and improve our algorithms. For example, Samtech S.A. or BRGM have provided huge problems that we use to assess the performance and limits of our approaches. We have informal collaborations around MUMPS with a number of institutions: (i) industrial teams which experiment and validate our package, (ii) research teams with which we discuss new functionalities they would need, (iii) designers of finite element packages who integrate MUMPS as a solver for the internal linear systems, (iv) teams working on optimization, (v) physicists, chemists, etc., in various fields where robust and efficient solution methods are critical for their simulations. In all cases, we validate all our research and 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 (Davis collection, Rutherford-Boeing and PARASOL).

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

The GRID TLSE project (see [76]), coordinated by ENSEEIHT-IRIT, is an expertise site providing a one-stop shop for users of sparse linear algebra software. This project was initially funded by the ACI Grid (2002-2005). A user can access matrices, databases, information and references related to sparse linear algebra, and can also obtain actual statistics from runs of a variety of sparse matrix solvers on his/her own problem. Each expertise request leads to a number of elementary requests on a grid of computers for which the DIET middleware developed by GRAAL is used. MUMPS is one of the packages interfaced within the project and that a user will be able to experiment through GRID TLSE. Much work has been performed this year, with the goal to be able to open the site to the public.

7. Contracts and Grants with Industry

7.1. Contract with SAMTECH, 2005-2006

INRIA and ENSEEIHT-IRIT have a contract with the company Samtech S.A. (Belgium), that develops the European finite element software package SAMCEF. The goal was to study how a parallel sparse out-of-core approach can help solving problems from customers of Samtech, for which classical parallel direct methods require too much memory, even on high-end platforms, and where iterative solvers fail to provide a correct solution. The contract is 18 months long, and relies on the use of the software package MUMPS. The new functionalities developed for this contract will be made available in a future public release of the package. We also use the work performed in the context of our research on scheduling aspects in the context of out-of-core approaches. A prototype out-of-core implementation is currently used by Samtech and allows to treat simulation problems that could not be processed before, although size of problems and performance could be further improved.

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

8. Other Grants and Activities

8.1. Regional Projects

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

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

J.-Y. L'Excellent is the correspondent of this project for GRAAL.

8.1.2. Institut des Sciences et Technique de l'Information

J.-M. Nicod and L. Philippe are involved in ISTI (Regional Institute for Information Sciences and Technologies). L. Philippe leads the "Micro-Factory of the future" project. The aim of this project is to design the information model and management part of a micro-factory composed of cells. Each cell contains a set of micro-robots which manipulate micro-products (about 10^{-5} meters).

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

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

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

8.1.4. Projet "Calcul Hautes Performances et Informatique Distribuée"

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

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

8.2. National Contracts and Projects

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

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

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

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

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

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

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

INRIA is financing Aurélia Fèvre, on contract from September 1, 2005 to August 31, 2007, as an engineer to work on the development of the MUMPS software package. While helping with various aspects of the software issues and maintenance, she has more specifically been developing a Scilab interface to the sequential version of MUMPS, allowing the use of an efficient solver for sparse systems of linear equations from within Scilab. She worked on performance optimization, and is now involved in the validation of the software, using code coverage tools.

8.2.4. Ministry Grant: ACI Grandes masses de données Grid Data Service, 2003-2006

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 [9] platform for peer-to-peer data management. The platform will serve to implement and compare multiple replication and data consistency strategies defined together by the PARIS team (IRISA) and by the REGAL team (LIP6).

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

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

ENS Lyon is involved in the GRID'5000 project [81], which aims at building an experimental Grid platform gathering nine sites geographically distributed in France (17 laboratories). Each site hosts several clusters connected through the RENATER network.

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

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

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

Yves Robert is leading the Rhône-Alpes site of this project, which comprises two other sites: Paris (LIX and LRI laboratories) and Bordeaux-Rennes (Paris and Scalapplix projects). Anne Benoit and Frédéric Vivien participate in this project, together with Lionel Eyraud, who holds a post-doctoral position since October 17. Lionel Eyraud is working on methods and tools for topology discovery.

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

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

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

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

8.2.8. ANR grant: SOLSTICE (SOlveurs et simulaTIon en Calcul Extrême), 3 years, 2007-2010

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

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

8.2.9. SEISCOPE Consortium

The SEISCOPE project focuses on wave propagation problems and seismic imaging. This year, our parallel solver MUMPS has been used extensively for finite-difference modeling of acoustic wave propagation (see [51]) and we have had many interactions with Stéphane Operto and Jean Virieux (GeoScience Azur, coordinators of the SEISCOPE project). The SEISCOPE project is supported by ANR (Agence National de la Recherche Française), and by BP, CGG, SHELL and TOTAL. Our research on out-of-core methods is also of great interest for seismic imaging and will soon be experimented in the context of this project.

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

8.3. International Contracts and Projects

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

PhD Student E. Agullo obtained a funding from Région Rhône Alpes (Explora'Doc programme) for a 6-month visit to the Lawrence Berkeley National Laboratory (California, USA) under the supervision of S. X. Li. This collaboration aims at comparing two direct out-of-core approaches (multifrontal and left-looking) for solving large sparse linear systems. A first short visit to Berkeley in September 2006 allowed the student and the team to define more accurately the objectives of the collaboration and to present each other their respective approaches. The next visit is scheduled from May/June to September/October 2007.

9. Dissemination

9.1. Scientific Missions

CoreGrid: CNRS is a partner of the CoreGrid network of excellence. The CNRS partnership involves Algorille in Nancy (E. Jeannot), ID-Imag in Grenoble (G. Huard, D. Trystram) and the Graal project (A. Benoit, Y. Caniou, E. Caron, F. Desprez, Y. Robert, F. Vivien). F. Vivien leads the CNRS contribution. He is also responsible for two tasks in the scheduling workpackage.

9.2. Animation Responsibilities

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

9.3. Edition and Program Committees

Anne Benoit co-organized the Third International Workshop on Practical Aspects of High-level Parallel Programming (PAPP 2006), University of Reading, UK, May 2006, and she is co-organizing the fourth edition of the workshop PAPP 2007, University of Beijing, China, May 2007.

A. Benoit was a member of the program committee of ICCS 2006, and she is a member of the program committee of ICCS 2007.

Yves Caniou is a member of the program committee of Heterogeneous Computing Worshop 2007 (HCW'07).

Eddy Caron was a member of the program committee of HCW 06 (Heterogeneous Computing Workshop), Rhodes Island, Greece, April 25, 2006, to be held in conjunction with IPDPS 2006 (and he will be program committee member of HCW 07). He was a member of the program committee of RenPar'2006 (Rencontres francophones du Parallélisme).

Frédéric Desprez is an associate editor of *Parallel and Distributed Computing Practices* and *Computing Letters* (COMPULETT).

F. Desprez participated to the program committees of CLADE'06, PPGaMS, Second International Summer School on Grid Computing, ICCSA'05-06, PGaMS (2nd Workshop on Programming Grids and Metasystems, Large Scale Computations on Grids (LaSCoG), Grid'06, VecPar'06, HeteroPAR'06, ISPA'06, Grid area at SC2006, High-Performance Scientific and Engineering Computing (HPCC-06) and SC2006. He is a member of the EuroPar Advisory board and of the editorial board of "Scalable Computing: Practice and Experience" (SCPE).

Aurélia Fèvre and Jean-Yves L'Excellent organized, with Patrick Amestoy, the first workshop dedicated to MUMPS users (MUMPS Users' Day, ENS Lyon, 24 October 2006), see http://graal.ens-lyon.fr/MUMPS/users_day.html.

Jean-Yves L'Excellent was a member of the program committee of ICPADS'2006 (IEEE International Conference on Parallel and Distributed Systems), Minneapolis, USA.

Yves Robert is a member of the editorial board of the *International Journal of High Performance Computing Applications* (Sage Press). In 2006 he has co-edited three special issues of journals:

- 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, February 2006.
- Two special issues of Int. J. High Performance Computing Applications, on *Scheduling techniques for large-scale distributed platforms*, edited by L. Carter, H. Casanova, F. Desprez, J. Ferrante and Y. Robert, Spring and Winter 2006. These special issues are the follow-on of the workshops organized in Aussois and San Diego.

Y. Robert was vice-chair (topic Algorithms) of the program committee of ICPADS'2006 (IEEE International Conference on Parallel and Distributed Systems), Minneapolis, USA.

Y. Robert is program chair of HiPC'2006 (IEEE Int. Conf. on High Performance Computing), Bangalore, India. He is the first editor of the proceedings (to appear in the LNCS series, Springer Verlag).

Y. Robert is vice-chair (topic Algorithms) of the program committee of IPDPS'07 (IEEE International Parallel and Distributed Processing Symposium), Long Beach, USA. He will be program chair of IPDPS'08.

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

Y. Robert gave an invited talk at IPDPS'2006 in Rhodes, Greece.

Y. Robert has been elected an IEEE Fellow (promotion 2006). In November 2006, Y. Robert has been elected as a Senior Member of Institut Universitaire de France.

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

F. Vivien was a member of the program committee of Grid 2006 (7th IEEE/ACM International Conference on Grid Computing), Barcelona, Spain, September 28-29, 2006; RenPar 2006 (17e Rencontres Francophones du Parallélisme), October 2006, Montpellier, France; HiPC 2006 (13th International Conference on High Performance Computing), December 2006, Bangalore, India; PDP 2007 (15th Euromicro Conference on Parallel, Distributed and Network-based Processing), February 7-9 2007, Naples, Italy; IPDPS 2007 (21-st IEEE International Parallel & Distributed Processing Symposium), March 26-30, 2007, Long Beach, California, USA.

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

L. Philippe is a member of the program committee of *Journées des composants*, French workshop on Components models and applications.

9.4. Administrative and Teaching Responsibilities

9.4.1. Administrative Responsibilities

Competitive selection for ENS Lyon students. Y. Robert was responsible of the practical computer science test which is part of the oral examination in the competitive selection of the students of the École normale supérieure de Lyon.

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

9.4.2. Teaching Responsibilities

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

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

Bachelor degree (Maîtrise) in computer science, L. Philippe is responsible of the Client/Server and distributed programming lecture since 2000.

Bachelor and Master degree 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.

10. Bibliography

Major publications by the team in recent years

[1] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT. *Multifrontal parallel distributed symmetric and unsymmetric solvers*, in "Comput. Methods Appl. Mech. Eng.", vol. 184, 2000, p. 501–520.

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

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

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

[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. *Constructing Memory-minimizing Schedules for Multifrontal Methods*, in "ACM Transactions on Mathematical Software", vol. 32, n^{o} 1, 2006, p. 17–32.

[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.

Year Publications

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 (I)*, Int. J. High Performance Computing Applications 20, 1, 2006.

[11] L. CARTER, H. CASANOVA, F. DESPREZ, J. FERRANTE, Y. ROBERT (editors). *Special issue on Scheduling techniques for large-scale distributed platforms (II)*, Int. J. High Performance Computing Applications 20, 4, 2006.

[12] 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.

Doctoral dissertations and Habilitation theses

[13] P. K. CHOUHAN. *Automatic Deployment for Application Service Provider Environments*, Ph. D. Thesis, École Normale Supérieure de Lyon, France, September 2006.

[14] L. MARCHAL. *Communications collectives et ordonnancement en régime permanent sur plates-formes hétérogènes*, Ph. D. Thesis, École Normale Supérieure de Lyon, France, October 2006.

[15] A. VERNOIS. *Ordonnancement et réplication de données bioinformatiques dans un contexte de grille de calcul*, Ph. D. Thesis, École Normale Supérieure de Lyon, France, October 2006.

Articles in refereed journals and book chapters

[16] G. ANTONIU, M. BERTIER, L. BOUGÉ, E. CARON, F. DESPREZ, M. JAN, S. MONNET, P. SENS. *Future Generation Grids*, V. GETOV, D. LAFORENZA, A. REINEFELD (editors). , Proceedings of the Workshop on Future Generation Grids November 1-5, 2004, Dagstuhl, Germany, vol. XVIII, CoreGrid Series, chap. GDS: An Architecture Proposal for a Grid Data-Sharing Service, Springer Verlag, 2006.

[17] O. BEAUMONT, L. MARCHAL, Y. ROBERT. *Complexity results for collective communications on heterogeneous platforms*, in "Int. Journal of High Performance Computing Applications", vol. 20, n^{o} 1, 2006, p. 5-17.

[18] A. BENOIT, B. PLATEAU, W. J. STEWART. *Memory Efficient Kronecker algorithms with applications to the modelling of parallel systems*, in "Future Generation Computer Systems, special issue on System Performance Analysis and Evaluation", vol. 22, n^{o} 7, August 2006, p. 838–847.

[19] R. BOLZE, F. CAPPELLO, E. CARON, M. DAYDÉ, F. DESPREZ, E. JEANNOT, Y. JÉGOU, S. LANTERI, J. LEDUC, N. MELAB, G. MORNET, R. NAMYST, P. PRIMET, B. QUETIER, O. RICHARD, E.-G. TALBI, T. IRENA. *Grid'5000: a large scale and highly reconfigurable experimental Grid testbed*, in "International Journal of High Performance Computing Applications", vol. 20, n^{o} 4, November 2006, p. 481-494.

[20] Y. CANIOU, E. JEANNOT. *Multi-Criteria Scheduling Heuristics for GridRPC Systems*, in "Special edition of The International Journal of High Performance Computing Applications (IJHPCA)", vol. 20, n^{o} 1, spring 2006, p. 61-76.

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

[22] E. CARON, F. DESPREZ, J.-Y. L'EXCELLENT, C. HAMERLING, M. PANTEL, C. PUGLISI-AMESTOY. *Future Generation Grids*, V. GETOV, D. LAFORENZA, A. REINEFELD (editors). , Proceedings of the Workshop on Future Generation Grids November 1-5, 2004, Dagstuhl, Germany, vol. XVIII, CoreGrid Series,

chap. Use of a Network Enabled Server System for a Sparse Linear Algebra Application, Springer Verlag, 2006.

[23] E. CARON, F. DESPREZ, C. TEDESCHI. *Enhancing Computational Grids with Peer-to-Peer technology for Large Scale Service Discovery*, in "Journal of Grid Computing", To appear, 2007.

[24] P. K. CHOUHAN, H. DAIL, E. CARON, F. VIVIEN. *Automatic Middleware Deployment Planning on Clusters*, in "International Journal of High Performance Computing Applications", vol. 20, n^{o} 4, November 2006, p. 517-530.

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

[26] N. GARNIER, A. FRIEDRICH, R. BOLZE, E. BETTLER, L. MOULINIER, C. GEOURJON, J. D. THOMPSON, G. DELEAGE, O. POCH. *MAGOS: multiple alignment and modelling server*, in "Bioinformatics", vol. 22, n^{o} 17, 2006, p. 2164-2165, http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/17/2164.

[27] A. GIERSCH, Y. ROBERT, F. VIVIEN. *Scheduling tasks sharing files on heterogeneous master-slave platforms*, in "Journal of Systems Architecture", vol. 52, n^{o} 2, 2006.

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

[29] L. MARCHAL, Y. YANG, H. CASANOVA, Y. ROBERT. *Steady-state scheduling of multiple divisible load applications on wide-area distributed computing platforms*, in "Int. Journal of High Performance Computing Applications", vol. 20, n^{o} 3, 2006, p. 365-381.

[30] H. RENARD, Y. ROBERT, F. VIVIEN. *Data redistribution algorithms for heterogeneous processor rings*, in "Int. Journal of High Performance Computing Applications", vol. 20, n^{o} 1, 2006, p. 31-43.

Publications in Conferences and Workshops

[31] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Étude mémoire d'une méthode multifrontale parallèle hors-mémoire*, in "17e Rencontres Francophones en Parallélisme (RenPar'16), Perpignan, France", 2006, p. 220–227.

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

[33] A. AMAR, R. BOLZE, A. BOUTEILLER, P. K. CHOUHAN, A. CHIS, Y. CANIOU, E. CARON, H. DAIL, B. DEPARDON, F. DESPREZ, J.-S. GAY, G. LE MAHEC, A. SU. *DIET: New Developments and Recent Results*, in "CoreGRID Workshop on Grid Middleware (in conjunction with EuroPar2006), Dresden, Germany", August 28-29 2006.

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

[35] O. BEAUMONT, A.-M. KERMARREC, L. MARCHAL, É. RIVIÈRE. *Voronet, un réseau objet-à-objet sur le modèle petit-monde*, in "CFSE'5: Conférence Française sur les Systèmes d'Exploitation", 2006.

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

[37] R. BOLZE, E. CARON, F. DESPREZ, G. HOESCH, C. PONTVIEUX. *A Monitoring and Visualization Tool and Its Application for a Network Enabled Server Platform*, in "Computational Science and Its Applications - ICCSA 2006, Glasgow, UK", M. GAVRILOVA (editor). , LNCS, vol. 3984, Springer, May 8-11 2006, p. 202–213.

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

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

[40] E. CARON, P. K. CHOUHAN, H. DAIL. *GoDIET: A Deployment Tool for Distributed Middleware on Grid'5000*, in "EXPGRID workshop. Experimental Grid Testbeds for the Assessment of Large-Scale Distributed Apllications and Tools. In conjunction with HPDC-15, Paris, France", IEEE, June 19th 2006, p. 1-8.

[41] E. CARON, F. DESPREZ, C. FOURDRIGNIER, F. PETIT, C. TEDESCHI. *A Repair Mechanism for Fault-Tolerance for Tree-Structured Peer-to-Peer Systems*, in "HiPC'2006. 12th International Conference on High Performance Computing, Bangalore. India", Y. ROBERT, M. PARASHAR, R. BADRINATH, V. K. PRASANNA (editors). , LNCS, vol. 4297, Springer-Verlag Berlin Heidelberg, December 18-21 2006, p. 171-182.

[42] E. CARON, F. DESPREZ, C. TEDESCHI. *A Dynamic Prefix Tree for the Service Discovery Within Large Scale Grids*, in "The Sixth IEEE International Conference on Peer-to-Peer Computing, P2P2006, Cambridge, UK.", A. MONTRESOR, A. WIERZBICKI, N. SHAHMEHRI (editors). , IEEE, September 6-8 2006, p. 106-113.

[43] E. CARON, C. FOURDRIGNIER, F. PETIT, C. TEDESCHI. *Mécanisme de réparations pour un système P2P de découverte de services*, in "Perpi'2006 - Conférences conjointes RenPar'17 / SympA'2006 / CFSE'5 / JC'2006, Canet en Roussillon", October 4-6 2006, p. 252-259.

[44] P. K. CHOUHAN, H. DAIL, E. CARON, F. VIVIEN. *How should you structure your hierarchical scheduler?*, in "HPDC-15. 15th IEEE International Symposium on High Performance Distributed Computing, Paris, France", IEEE, June 19-23 2006, p. 339-340 (Poster).

[45] S. DAHAN, J.-M. NICOD, L. PHILIPPE. *Utilisation du Distributed Spanning Tree en tant qu'Overlay*, in "RenPar'17, Perpignan, France", O. BEAUMONT, V. BOUDET (editors). , Université de Perpignan, October 2006, p. 60–67.

[46] B. DEL-FABBRO, D. LAIYMANI, J.-M. NICOD, L. PHILIPPE. *Design and experimentations of an efficient data management service for NES architectures*, in "2nd VLDB Workshop on Data Management in Grids (VLDB DMG 06), Seoul, Korea", J. PIERSON (editor). , September 2006, p. 62-75.

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

[48] J.-S. GAY, Y. CANIOU. *Simbatch : une API pour la simulation et la prédiction de performances de systèmes batch*, in "17e Rencontres Francophones du Parallélisme, des Architectures et des Systèmes, Perpignan", October 2006, p. 180–187.

[49] A. LEGRAND, A. SU, F. VIVIEN. *Minimizing the stretch when scheduling flows of biological requests*, in "SPAA '06: Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures, Cambridge, Massachusetts, USA", ACM Press, 2006, p. 103–112, http://doi.acm.org/10.1145/1148109.1148124.

[50] N. MARILLEAU, C. LANG, P. CHATONNAY, L. PHILIPPE. *An Agent-Based Framework for Urban Mobility Simulation*, in "Procs of the 14th IEEE Euromicro Conference on Parallel, Distributed and Network based Processing (PDP 2006), Montbéliard, France", February 2006, p. 355–361.

[51] S. OPERTO, J. VIRIEUX, P. AMESTOY, L. GIRAUD, J.-Y. L'EXCELLENT. *3D frequency-domain finite-difference modeling of accoustic wave propagation using a masively parallel direct solver: a feasable study*, in "The Society of Exploration Geophysicists (SEG 2006), New Orleans (USA)", October 1-6 2006, p. 2265–2269.

[52] L. PHILIPPE, S. DAHAN, I. DJAMA, S. DAMY, B. HERRMANN. *Evaluation of a large scale lookup algorithm for ASP based grids*, in "Procs of the 5th Int. Symposium on Parallel and Distributed Computing, Timisoara, Romania", IEEE computer society, July 2006, p. 220–229.

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

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

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

[56] C. TEDESCHI. *Découverte de services pour les grilles de calcul dynamiques large échelle*, in "Perpi'2006 - Conférences conjointes RenPar'17 / SympA'2006 / CFSE'5 / JC'2006, Canet en Roussillon, France", October 4-6 2006, p. 52-59.

Internal Reports

[57] A. AMAR, R. BOLZE, A. BOUTEILLER, A. CHIS, Y. CANIOU, E. CARON, P. K. CHOUHAN, G. LE MAHEC, H. DAIL, B. DEPARDON, F. DESPREZ, J.-S. GAY, A. SU. *DIET: New Developments and Recent Results*, Also available as LIP Research Report 2006-31, Research Report, n^{o} 6027, INRIA, 11 2006, https://hal.inria.fr/inria-00115569.

[58] G. ANTONIU, E. CARON, F. DESPREZ, M. JAN. *Towards a Transparent Data Access Model for the GridRPC Paradigm*, Also available as IRISA Research Report PI1823, Technical report, n^{o} RR-6009, INRIA, November 2006, https://hal.inria.fr/inria-00110902.

[59] A. BALLIER, E. CARON, D. EPEMA, H. MOHAMED. *Simulating Grid Schedulers with Deadlines and Co-Allocation*, Also available as LIP Research Report 2006-01 and INRIA Research Report RR-5815, Technical report, n^{o} TR-0061, CoreGRID, October 2006, http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0061.pdf.

[60] R. BOLZE, E. CARON, F. DESPREZ, G. HOESCH, C. PONTVIEUX. *A Monitoring and Visualization Tool and Its Application for a Network Enabled Server Platform*, Also available as LIP Research Report 2006-14, Technical report, n^{o} RR-5879, INRIA, April 2006, ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-5879.pdf.

[61] E. CARON, A. CHIS, F. DESPREZ, A. SU. *Plug-in Scheduler Design for a Distributed Grid Environment*, Also available as LIP Research Report 2006-41, Technical report, n^{o} RR-6030, INRIA, November 2006, https://hal.inria.fr/inria-00115949.

[62] E. CARON, P. K. CHOUHAN, H. DAIL. *GoDIET: A Deployment Tool for Distributed Middleware on Grid 5000*, Also available as LIP Research Report 2006-17, Technical report, n^{o} RR-5886, INRIA, April 2006, http://hal.inria.fr/inria-00071382.

[63] E. CARON, F. DESPREZ, C. FOURDRIGNIER, F. PETIT, C. TEDESCHI. *A Repair Mechanism for Fault-Tolerance for Tree-Structured Peer-to-Peer Systems*, Also available as LIP Research Report 2001-34, Technical report, n^{o} RR6029, INRIA, October 2006, https://hal.inria.fr/inria-00115997.

[64] E. CARON, F. DESPREZ, C. TEDESCHI. *A Dynamic Prefix Tree for the Service Discovery Within Large Scale Grids*, Also available as LIP Research Report 2001-33, Technical report, n^{o} RR-6028, INRIA, October 2006, https://hal.inria.fr/inria-00116111.

[65] J. DONGARRA, J.-F. PINEAU, Y. ROBERT, Z. SHI, F. VIVIEN. *Revisiting Matrix Product on Master-Worker Platforms*, Research report, n^{o} RR-6053, INRIA, 2006, http://hal.inria.fr/inria-00117050.

[66] A. FÈVRE, J.-Y. L'EXCELLENT, S. PRALET. *Scilab and MATLAB Interfaces to MUMPS*, Also appeared as ENSEEIHT-IRIT report TR/TLSE/06/01 and LIP report RR2006-06, Technical report, n^{o} RR-5816, INRIA, January 2006, https://hal.inria.fr/inria-00070209.

[67] M. GALLET, Y. ROBERT, F. VIVIEN. *Scheduling communication requests traversing a switch: complexity and algorithms*, Also available as an INRIA research report, Research report, n^{o} RR2006-25, LIP, ENS Lyon, 2006.

[68] J.-S. GAY, Y. CANIOU. *Simbatch: an API for simulating and predicting the performance of parallel resources and batch systems*, Research Report, n^{o} 6040, INRIA, 11 2006, https://hal.inria.fr/inria-00115880.

[69] A. LEGRAND, A. SU, F. VIVIEN. *Minimizing the stretch when scheduling flows of divisible requests*, Research report RR-6002, INRIA, 2006, http://hal.inria.fr/inria-00108524.

[70] L. MARCHAL, V. REHN, Y. ROBERT, F. VIVIEN. *Scheduling and data redistribution strategies on star platforms*, Research report, n^{o} RR-6005, INRIA, 2006, http://hal.inria.fr/inria-00108518.

[71] C. TEDESCHI. *Découverte de services pour les grilles de calcul dynamiques large échelle*, Also available as INRIA Research Report, Technical report, n^{o} RR2006-44, Laboratoire de l'Informatique du Parallélisme (LIP), November 2006, http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2006/RR2006-44.pdf.

References in notes

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

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

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

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

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

[77] 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.

[78] 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.

[79] 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/.

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

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

[82] I. S. DUFF, S. PRALET. *Towards a stable static pivoting strategy for the sequential and parallel solution of sparse symmetric indefinite systems* , Also appeared as RAL report RAL-TR-2005-007 and CERFACS report TR/PA/05/26, Rapport de recherche, n^{o} RT/TLSE/05/08, IRIT, April 2005.

[83] 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.

[84] 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.

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

[86] 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.

[87] 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.

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

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

[90] 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.

[91] 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.

[92] 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.

[93] 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.

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

[95] 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.

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

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