INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team ReMaP

Regularity and Massive Parallelism

Rhône-Alpes

THEME 1A

Table of contents

1. Team

The ReMaP 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 project-team
Frédéric Desprez [DR INRIA]

Administrative assistants
Sylvie Boyer [INRIA, 30% on the project]
Anne-Pascale Botonnet [ENS Lyon, 30% on the project, until August 31, 2003]

INRIA staff
Frédéric Desprez [DR]
Jean-Yves L'Excellent [CR]
Gil Utard [CR (secondment), until August 31, 2003]
Frédéric Vivien [CR]

Faculty members from ENS Lyon
Eddy Caron [Assistant Professor]
Yves Robert [Professor]

Project technical staff
Ludovic Bertsch [on contract from INRIA, until February 10, 2003]
Philippe Combes [on contract from INRIA, until September 30, 2003]
Christophe Pera [on contract from ENS Lyon]

Post-doctoral fellow
Matthias Colin [CNRS, since September 1, 2003]

Ph. D. students
Pushpinder-Kaur Chouhan [MENRT grant, since September 1, 2003]
Abdou Guermouche [MENRT grant]
Arnaud Legrand [ENS grant]
Loris Marchal [ENS grant, since September 1, 2003]
Martin Quinson [MENRT grant]
Hélène Renard [MENRT grant (ACI GRID)]
Antoine Vernois [MENRT grant (ACI GRID)]

2. Overall Objectives

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

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 [61]. An enormous amount of financing has recently been put on this important subject, leading to an exponential growth of the number of projects, most of them focusing on low level software details. We believe that many of these projects

failed to study fundamental problems such as problems and algorithms complexity, and scheduling heuristics. Also they usually have not validated their theoretical results on available software platforms.

From the architectural point of view, Grid computing has different scales but is always highly heterogeneous and hierarchical. At a very large scale, thousands of PCs connected through the Internet are aggregated to solve very large applications. This form of the Grid, usually called a Peer-to-Peer (P2P) system, has several incarnations, such as SETI@home, Gnutella or XtremWeb [59]. It is already used to solve large problems (or to share files) on PCs across the world. However, as today's network capacity is still low, the applications supported by such systems are usually embarrassingly parallel. Another large-scale example is the American TeraGRID which connects several supercomputing centers in the USA and reaches a peak performance of 13.6 Teraflops. At a smaller scale but with a high bandwidth, one can mention the RNRT VTHD++ project [1] which connects several France Telecom and INRIA research centers (and the PC clusters available in those centers) and several other laboratories with a 2.5 Gb/s network. On such a platform, the network between the research centers is even faster than the network within each cluster connected to it. Many such projects exist over the world that connect a small set of machines through a fast network. Finally, at the research laboratory level, one can build a 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 [42][44][61]. Today's environments are based either on the adaptation of "classical" parallel environments or on the development of toolboxes based on Web Services.

2.1.1. Aims of the ReMaP project

In the *ReMaP* 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 *ReMaP* project are

Algorithmic Design + Middleware/Libraries + Applications

over heterogeneous architectures and the Grid

[1] Réseau à Vraiment Très Haut Débit.

3. Scientific Foundations

3.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Arnaud Legrand, Loris Marchal, Hélène Renard, Yves Robert, 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 [48][57][79][83] and the references therein. This model was introduced for homogeneous processors, and has been (straightforwardly) extended for heterogeneous computing resources. In a word, there is a limited number of computing resources, or processors, to execute the tasks. Communication delays are taken into account as follows: let task T be a predecessor of task T' in the task graph; if both tasks are assigned to the same processor, no communication overhead is incurred, the execution of T' can start immediately at the end of the execution of T; on the contrary, if T and T' are assigned to two different processors P_i and P_j, a communication delay is incurred. More precisely, if P_i completes the execution of T at time-step t, then P_j cannot start the execution of T' before time-step $t + \text{comm}(T, T', P_i, P_j)$, where $\text{comm}(T, T', P_i, P_j)$ is the communication delay, which depends upon both tasks T and T' and both processors P_i and P_j. Because memory accesses are typically several orders of magnitude cheaper than inter-processor communications, it is sensible to neglect them when T and T' are assigned to the same processor.

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

The general scheduling problem is far more complex than the traditional objective in the *macro-dataflow* model. Indeed, the nature of the scheduling problem depends on the type of tasks to be scheduled, on the platform architecture, and on the aim of the scheduling policy. The tasks may be independent (e.g., they represent jobs submitted by different users to a same system, or they represent occurrences of the same program run on independent inputs), or the tasks may be dependent (e.g., they represent the different phases of a same processing and they form a task graph). The platform may or may not have a hierarchical architecture (clusters of clusters vs. a single cluster), it may or may not be dedicated. Resources may be added to or may disappear from the platform at any time, or the platform may have a stable composition. The processing units may have the same characteristics (e.g., computational power, amount of memory, they support multi-port or only one-port communications, 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 *ReMaP* project, we investigate scheduling problems that are of practical interest in the context of large-scale distributed platforms. We assess the impact of the heterogeneity and volatility of the resources onto the scheduling strategies.

3.2. Scheduling for Sparse Direct Solvers

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

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

Because of their efficiency and robustness, direct methods (based on Gaussian factorization) are methods of choice to solve these types of problems. In this context, we are particularly interested in the multifrontal method [55][56], for symmetric positive definite, general symmetric or unsymmetric problems, with numerical pivoting to ensure numerical stability. Note that numerical pivoting induces dynamic data structures that are unpredictable symbolically or from a static analysis.

The multifrontal method is based on an elimination tree [72] which results from the graph structure corresponding to the nonzero pattern of the problem to be solved, and from the order in which variables are eliminated. This tree provides the dependency graph of the computations and is exploited to define tasks that may be executed in parallel. In this method, each node of the tree corresponds to a task (itself potentially parallel) that consists in the partial factorization of a dense matrix. This approach allows for a good locality and usage of cache memories.

In order to deal with numerical pivoting and keep an approach as much adaptative as possible to existing and newer parallel computer architectures, we are especially interested in approaches that are intrinsically dynamic and asynchronous [38][39]. 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 at the same time responsible for providing work to some others and acting as a slave for others. These algorithms are very interesting from the point of view of parallelism and in particular for the study of mapping and scheduling strategies for the following reasons:

- the associated task graphs are very irregular and dynamic,
- these algorithms are currently used inside industrial applications, and
- the evolution of high performance platforms, more heterogeneous and less predictable, requires applications to adapt using a mixture of dynamic and static approaches, as is allowed by our approach.

Note that our research in this field is strongly linked to the software platform MUMPS (see Section 5.2) which is our main platform to experiment and validate new ideas and research directions.

3.3. Providing Access to HPC Servers on the Grid

Participants: Ludovic Bertsch, Eddy Caron, Pushpinder-Kaur Chouhan, Matthias Colin, Philippe Combes, Frédéric Desprez, Christophe Pera, Martin Quinson.

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.

The first approach provides the user with a uniform view of resources. This is the case of GLOBUS [63] which provides transparent MPI communications (with MPICH-G2) between distant nodes but does not manage load balancing issues between these nodes. It's the user's task to develop a code that will take into account the heterogeneity of the target architecture. Classical batch processing can also be used on the Grid with projects like Condor-G [49] or Sun GridEngine [66]. Finally, peer-to-peer [80] or Global computing [62] 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 [73][77]) offers an easy access to available resources to a Web browser, a Problem Solving Environment (PSE), or a simple client program written in C, Fortran, or Java. It also provides more transparency by hiding the search and allocation of computing resources. We favour this second approach.

In a Grid context, this approach requires the implementation of middlewares to facilitate the client access to remote resources. In the ASP approach, a common way for clients to ask for resources to solve their problem is to submit a request to the middleware. The middleware will find the most appropriate server that will solve the problem on behalf of the client using a specific software. Several environments, usually called Network Enabled Servers (NES), have developed such a paradigm: NetSolve [41], Ninf [78], NEOS [60], OmniRPC [82], and more recently DIET developed in the ReMaP project. A common feature of these environments is that they are built on top of five components: clients, servers, databases, monitors and schedulers. Clients solve computational requests on servers found by the NES. The NES schedules the requests on the different servers using performance information obtained by monitors and stored in a database.

Designing such a NES implies to address issues linked to several well-known research domains:

- scheduling to allow clients to chain requests in a workflow mode,
- middleware and application platforms as a base to implement the necessary "glue" to broke clients requests, find the best server available, and then submit the problem and its data,
- distributed algorithms to manage the requests and the dynamic behavior of the platform.

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

3.3.1. Resource Management for NES platforms.

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

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

the target platform and schedule a whole program. Since scalability is one of our main concerns, we need to improve the scalability of the scheduling itself by using distributed scheduling techniques.

The second issue is data placement and persistence. Parallel applications usually run on very large data, unlike traditional client/servers applications. This means that the management of these data will be costly (both in transfer and storage). In the basic request submission scheme, data are sent by the client to a server. After the computation, results are returned from the server to the client. There is no other way to optimize data access in this context except choosing a server connected to the client by a high speed network or using data on-line compression [67] during transfer. However, problems are rarely solved in one step and a parallel job submission is often linked to other job submissions, in a workflow mode with dependences between the different tasks. In that case, it is possible to optimize data mapping and transfer by choosing a server for a job depending on the location of the data. Experiments have already proven that leaving data on servers and re-using them for the next job leads to substantial performance improvement.

Another research domain related to designing a NES is fault tolerance. Implementing a distributed environment without taking fault tolerance into consideration is meaningless. Indeed, the probability of a server fault or crash increases with the size of the network. NES are designed to manage resources distributed in large scale networks and several kinds of faults may occur, e.g. network or server crashes. Several research projects have already tried to address this issue in PSE environments. The CUMULVS project [70] proposes fault-tolerance of distributed simulations through heterogeneous task migration and user-directed checkpointing. Fault tolerance is managed at the application level. Other projects like the Los Alamos Message Passing Interface (LA-MPI) [65] or MPICH-V [43] provide a fault-tolerant message passing system. In the CORBA middleware fault tolerance is standardized [50].

3.3.2. Middleware and Service Offers Discovery.

Middlewares provide clients access to servers in distributed environments. They support basic functions such as remote procedure calls to communicate and higher level services such as dynamic invocation or service offers discovery to facilitate the application development. They also hide several difficulties in the application deployment and execution by providing uniform access to resources in heterogeneous distributed environments. Several norms and implementations have fixed standard interfaces to client/server middlewares: CORBA [89], Java/RMI [75], or DCOM [74]. These platforms are mainly used in general purpose architectures but not commonly in Grid computing. In Grid computing environments, using a middleware to implement a NES will have several advantages: transparent management of heterogeneity, automatic communication generation, transparent communication, fault tolerance, etc.

Middlewares have been designed to support client/server, multi-layered and distributed applications. Their services are not specialized to take Grid services needs into account and they leave several issues open: communication models, scalability, performance in exchanging large amount of data, fault tolerance, etc. Adapting these middlewares to Grid computing implies to address these issues either by adding new services to existing platforms or by modifying these platforms. Most of the current work using middlewares for Grid computing relies on using the communication middleware as a basis and on adding dedicated facilities. Thus several experiments have been driven to implement parallel algorithms on middlewares [52]. Other research projects try to extend middleware standards to add support for parallel applications [51]. Middlewares can also be used to implement a NES which will benefit from its services: for instance the ability to support multi-lingual applications.

Components technologies have been designed as a framework for general purpose distributed applications. One of their benefit is to separate non-functional code (managed by the container) from functional code written by the application developers. This provides more portable code by avoiding to mix system code which depends on the platform with code devoted to the application. Component technologies are already used in parallel platforms. This is the case of the parallel components developed in the PARIS project from INRIA. These components may be used to dynamically deploy new servers in distributed applications.

One of the issues set by NES environments is service discovery or lookup: to make the client's needs match a server's offer. Clients and servers are distributed world-wide and are dynamic: they might start or stop anytime.

Moreover, clients and servers must use a common language to describe services. Several research projects have tried to address this issue in large scale systems. Usually, propositions rely on servers distributed over the network, each recording local offers, and interconnected by a graph. In the CORBA standard, the Trading Service [86] is in charge of managing service offers. Traders may be federated by defining a federation graph and policies to forward requests when no offer is found. Thus, scalability properties of the Trading Service rely on its forwarding policies and on the federation graph definition which is given by administrators. But policies may not be uniform over the federation, which may lead to unpredictable behaviors. Moreover, dynamic properties (asked to the server) used to select offers are limited to simple data and may not support monitoring information such as the load of the network.

4. Application Domains

4.1. Applications of Sparse Direct Solvers

Our activity on sparse direct solvers and more precisely multifrontal solvers in distributed environments goes as far as making competitive software available to users. Such methods have a wide range of applications and they are at the heart of most techniques in numerical simulation: whether a model uses finite elements or differences, or requires the optimization of a complex linear or nonlinear function, one almost always ends up in solving a system of equations implying sparse matrices. There are therefore a number of application fields, among which we can list the most frequently cited by our users, i.e. the applications in which our sparse direct solver MUMPS (see Section 5.2) has been or is currently used: structural mechanical engineering (stress analysis, structural optimization, car bodies, ships, crankshaft segment, offshore platforms, CAD, CAE, rigidity of sphere packings), heat transfer analysis, thermomechanics in casting simulation, fracture mechanics, biomechanics, medical image processing, tomography, plasma physics (e.g., Maxwell's equations), critical physical phenomena, geophysics (e.g., seismic wave propagation, earthquake related problems, 3D wave propagation in inhomogeneous media for geophysical or optical problems), ad-hoc networking modeling (Markovian processes), modeling of the magnetic field inside machines, econometric models, soil-structure interaction problems, oil reservoir simulation, computational fluid dynamics (e.g., Navier-stokes, ocean/atmospheric modeling with mixed FEM, fluvial hydrodynamics, viscoelastic flows), electromagnetics, magneto-hydrodynamics, 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, aeroelasticity optical fiber modal analysis, blast furnace modeling, glaciology (models of ice flow), optimization, optimal control theory, education, astrophysics (e.g., supernova, thermonuclear reaction networks, neutron diffusion equation, quantum chaos, quantum transport), research on domain decomposition (MUMPS can for example be used inside each domain and can return the Schur complement), circuit simulations, etc.

We should notice that the current users of MUMPS include:

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

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 made by M. Memier, we optimize it while focusing on the cross-correlation problem based on a statistical operator.

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

4.4. Electronic Device Simulation

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

4.5. Biochemistry

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

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

In order to fully apprehend a model, it is necessary to explore the whole potential energy surface described by the independent variation of all its degrees of freedom. This involves the computation of many points on this surface.

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

4.6. Bioinformatics Application

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

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

In some cases, due to the lack of sufficient computing and storage resources, skilled staff or technical abilities, laboratories cannot afford such huge analyses. Grid computing may be a viable solution to the needs of the genomic research field: it can provide scientists with a transparent access to large computational and data management resources. DIET will be used as one Grid platform.

5. Software

5.1. DIET

Participants: Ludovic Bertsch, Eddy Caron, Pushpinder-Kaur Chouhan, Matthias Colin, Philippe Combes, Frédéric Desprez [correspondent], Christophe Pera, Martin Quinson, Frédéric Suter, Antoine Vernois.

5.1.1. Scilab$_{//}$

SCILAB [64] is a software for scientific computing developed in the Métalau project from INRIA. We developed a parallel version of this software funded by the ARC INRIA OURAGAN (1998-2000) [2].

The first approach of our Scilab parallelization allows the user to start other remote SCILAB sessions from the SCILAB window, make them communicate and use parallel numerical libraries.

To be able to use SCILAB as a tool for parallel computing, the first step was to provide a "regular" message passing interface for the user. This was done by including the standard PVM interface within SCILAB. This interface allows users to develop parallel programs and benefit from all the main features of SCILAB that simplify numerical computing. We choose PVM to implement the first message passing interface since it allows to dynamically spawn new processes which is not the case with MPI. Nevertheless, we also added a message passing interface based on MPI. Using MPI implies that the user has to decide at the beginning of its SCILAB session the maximum number of processes he/she will use.

In order to keep a good portability, interoperability and efficiency, SCILAB$_{//}$ also integrates interfaces to parallel linear algebra libraries like PBLAS [47], SCALAPACK [46], and the BLACS [87] communication

[2] URL: http://graal.ens-lyon.fr/~desprez/OURAGAN/.

library. Thus, the user may distribute his/her matrices and run parallel routines in order to achieve good performance. This level remains dedicated to "expert" users who have good parallel computing skills and who are familiar with the design of the SCALAPACK interface. Moreover, some numerical applications are limited by the physical memory size. To break this limit, out-of-core techniques may be employed, such as using disks as extensions of the main memory [54]. So we added an interface to the SCALAPACK out-of-core prototype.

To conclude, our development aims at providing users with a simple interface (Matlab-like) for high performance libraries (like ScaLAPACK or PETSc) accessible on parallel supercomputers and clusters. Several software layers have been used like classical message-passing libraries (like MPI or PVM) or Network Enabled Servers (like NetSolve or DIET).

5.1.2. DIET

Huge problems can now be computed over the Internet thanks to Grid Computing Environments like Globus or Legion. Because most of the current applications are numerical, the use of libraries like BLAS, LAPACK, ScaLAPACK, or PETSc is mandatory. The integration of such libraries in high level applications using languages like Fortran or C is far from being easy. Moreover, the computational power and memory needs of such applications may of course not be available on every workstation. Thus, the RPC paradigm seems to be a good candidate to build Problem Solving Environments on the Grid as explained in Section 3.3. The aim of the DIET [53] 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) [88] 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 Section 5.1.3).

The different components of our software architecture are the following:

- **Client**
  A client is an application which uses DIET to solve problems. Many kinds of clients should be able to connect to DIET from a web page, a Problem Solving Environment such as Matlab or Scilab, or from a compiled program.
- **Master Agent (MA)**
  An MA receives computation requests from clients. These requests refer to some DIET problems listed on a reference web page. Then the MA collects computational abilities from the servers and chooses the best one. The reference of the chosen server is returned to the client. A client can be connected to an MA by a specific name server or a web page which stores the various MA locations. Several MA can be deployed on the network to equilibrate the load among the clients.
- **Local Agent (LA)**
  An LA aims at transmitting requests and information between MAs and servers. The information stored on an LA is the list of requests and, for each of its subtrees, the number of servers that can solve a given problem and information about the data distributed in this subtree. Depending on the underlying network topology, a hierarchy of LAs may be deployed between an MA and the servers. No scheduling decision is made by an LA.

- **Server Daemon (SeD)**
  A SeD encapsulates a computational server. For instance it can be located on the entry point of a parallel computer. The information stored on a SeD is a list of the data available on its server (with their distribution and the way to access them), the list of problems that can be solved on it, and all information concerning its load (available memory and resources, etc). A SeD declares the problems it can solve to its parent LA. A SeD can give performance prediction for a given problem thanks to the FAST module, which is described in the next section.

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

5.1.3. FAST

FAST (*Fast Agent's System Timer*) [58] is a tool for dynamic forecasting of Network-Enabled Servers performance. This is a software package allowing client applications to get an accurate forecast of routine needs in terms of completion time, memory space, and amount of communication, as well as of current system availability. FAST relies on existing low level software packages, i.e. network and host monitoring tools, and some of our developments in modeling computation routines.

The goal of the FAST library is to provide the information needed by a scheduler. FAST models the needs of the tasks both in terms of time and memory space. Appropriate tools like NWS [88] are used to monitor the dynamic availability of system resources. FAST is also able to aggregate these two kinds of information in order to forecast the current computation time of a given task on a given machine. The goal of FAST is not to perform task placement, but to acquire the required knowledge to achieve it.

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

5.2. MUMPS

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

MUMPS (for *MUltifrontal Massively Parallel Solver*) [76] is a software package for the solution of large sparse systems of linear equations that uses a direct method (multifrontal method). It is developed in collaboration with ENSEEIHT-IRIT (Toulouse, France), CERFACS (Toulouse, France), and PARALLAB (Bergen, Norway). MUMPS is a parallel code for distributed memory architectures unique by the performance obtained and the number of functionalities available, among which we have:

- C or Fortran 90 interface,
- types of systems: symmetric positive definite, general symmetric, or unsymmetric,
- various matrix input formats: assembled or expressed as a sum of elemental matrices, centralized on one processor or pre-distributed on the processors,
- rank revealing functionalities (experimental), with computation of a null space basis,
- partial factorization and Schur complement matrix,
- real or complex arithmetic, single or double precision,
- partial threshold pivoting,
- fully asynchronous approach with overlap of computation and communication,
- distributed dynamic scheduling of the computational tasks to allow for a good load balance.

Note that the MUMPS software development was initiated by the European project PARASOL, whose results and developments were public domain. Lots of developments have been done by the authors since the end of that project, in order to enhance the software with more functionalities and integrate new results arising from our research. MUMPS is distributed free of charge and is currently being used by a few hundred of academic and industrial users for a large variety of applications (see Section 4.1).

5.3. SimGrid v2

Participant: Arnaud Legrand [correspondent].

The first version of SimGrid [84][45] was a discrete-event simulation toolkit. It provided a set of core abstractions and functionalities that can be used to easily build simulators for specific application domains and/or computing environment topologies. This allows the simulation of arbitrary performance fluctuations such as the ones observable for real resources due to background load. However, this first version lacked a number of abstractions (e.g. routing, scheduling agents). With SimGrid v2 we have added a new software layer to provide high-level abstractions and the software thus provides two interfaces:

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

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

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

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

6. New Results

6.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

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

Key words: *Algorithm design*, *heterogeneous platforms*, *scheduling strategies*, *parallelism*.

6.1.1. Steady-State Scheduling.

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

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

It would be very difficult to solve steady-state scheduling problems in a fully general framework, because it can be viewed as an extension, to heterogeneous platforms, of the NP-complete problem of software pipelining [37]. Instead, we investigate some restricted instances of the problem.

More precisely, we have considered the execution of a complex application on a heterogeneous "grid" computing platform. The complex application consists of a series of identical, independent problems to be solved. In turn, each problem consists of a set of tasks. There are dependences (precedence constraints) between these tasks, but there are no dependences from one problem to another one. A typical example is the repeated execution of the same algorithm on several distinct data samples. We showed how to determine the optimal steady-state scheduling strategy for each processor and how to build such a schedule. This result holds for a quite general framework, allowing for cycles and multiple paths in the platform graph.

6.1.2. Pipelined Execution of Macro-communication Schemes.

When analyzing the communications involved by the execution of complex applications, deployed on a heterogeneous "grid" platform, we see that such applications intensively use collective macro-communication schemes, such as scatters, personalized all-to-all or gather/reduce operations. As explained above, rather than aiming at minimizing the execution time of a single macro-communication, we focus on the steady-state operation. We assume that there is a large number of macro-communications to perform in pipeline fashion, and we aim at maximizing the throughput, i.e. the (rational) number of macro-communications which can be initiated every time-step. We target heterogeneous platforms, modeled by a graph where resources have different communication and computation speeds. The situation is simpler for series of scatters or personalized all-to-all than for series of reduce operations, because of the possibility of combining various partial reductions of the local values, and of interleaving computations with communications. In all cases, we show how to determine the optimal throughput, and how to exhibit a concrete periodic schedule that achieves this throughput.

We have extended this work to handle the case of pipelined broadcast operations. This particular macro-communication turned out to be surprisingly difficult, due to the possibility of "duplicating" the input; thereby forbidding the use of any conservation law. We have shown that achieving the best throughput may well require that the target platform is used in totality: we show that neither spanning trees nor DAGs are as powerful as general graphs. We show how to compute the best throughput using linear programming, and how to exhibit a periodic schedule, first when restricting to a DAG, and then when using a general graph. The polynomial compactness of the description comes from the decomposition of the schedule into several broadcast trees that are used concurrently to reach the best throughput. It is important to point out that a concrete scheduling algorithm based upon the steady-state operation is asymptotically optimal, in the class of all possible schedules (not only periodic solutions).

6.1.3. Divisible Loads.

A divisible task is a task that can be arbitrarily split in a linear fashion among any number of processors. This corresponds to a perfectly parallel task: any sub-task can itself be processed in parallel, and on any number of processors. On the practical side, the divisible load model provides a simple yet realistic framework to study the mapping of independent tasks on heterogeneous platforms. The granularity of the tasks can be chosen arbitrarily by the user, thereby providing a lot of flexibility in the implementation tradeoffs. Divisible load scheduling has been an active area of research for the last twenty years. A vast literature offers results and scheduling algorithms for various models for the underlying distributed computing platform. Broad surveys are available that report on accomplishments in the field. On the theoretical side, the success of the divisible load model is mostly due to its analytical tractability. Optimal algorithms and closed-form formulas exist for the simplest instances of the divisible load problem. This is in sharp contrast with the theory of task graph scheduling.

In our work, we have proposed a unified theoretical perspective that synthesizes previously published results, introduces several novel results, and raises open questions. Specifically, we discuss both one-round and multi-round algorithms, and we restrict our scope to the popular star and tree network topologies, which we study with both linear and affine cost models for communication and computation.

6.1.4. Load-Balancing for Communication-Aware Models.

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

1. Each processor may be provided with a routing table which specifies the links to be used to communicate with each other processor (hence the routing is fully static). Another hypothesis is to assume a dynamic routing, which is computed on the fly so as to optimize the network use.
2. At most one message can circulate on one link at a given time-step, so that contention for communication resources is taken into account statically. Another hypothesis is that several messages can circulate on one link at a given time-step, but the different messages share the total link bandwidth. The eXplicit Control Protocol XCP [69], for example, does enable to implement a bandwidth allocation strategy that complies with our hypotheses.

In this context, we have dealt with the mapping of iterative algorithms onto heterogeneous clusters. The application data is partitioned over the processors, which are arranged along a virtual ring. At each iteration, independent calculations are carried out in parallel, and then some communications take place between consecutive processors in the ring. The question is to determine how to slice the application data into chunks, and to assign these chunks to the processors, so that the total execution time is minimized. One major difficulty is to embed a processor ring into a network that typically is not fully connected, so that some communication links have to be shared by several processor pairs. We establish a complexity result that assesses the difficulty of this problem, and we design a practical heuristic that provides efficient mapping, routing, and data distribution schemes. We also design a greedy heuristic that assumes a fully-connected network, hence no contention (note that even with this simplification the problem remains NP-hard), thereby providing a comparative approach to assess the impact of link sharing.

6.1.5. Tasks Sharing Files.

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

Firstly, we dealt with a simple master-slave platform. The tasks depend upon (input) files which all initially reside on the master processor. The role of the master is to distribute the files to the processors, so that they can execute the tasks. The objective for the master is to select which file to send to which slave, and in which order, so as to minimize the total execution time. On the theoretical side, we established complexity results that assess the difficulty of the problem. On the practical side, we designed several new heuristics, which are shown to perform as efficiently as the best heuristics designed in the literature, although their costs are an order of magnitude lower.

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

6.2. Providing access to HPC servers on the Grid

Participants: Ludovic Bertsch, Eddy Caron, Pushpinder-Kaur Chouhan, Matthias Colin, Philippe Combes, Frédéric Desprez, Christophe Pera, Martin Quinson, Frédéric Suter, Gil Utard, Antoine Vernois.

Key words: *Numerical computing*, *computing server*, *performance forecasting*, *grid computing*.

2003 is the year of the release of version 1.0 of DIET. We stabilized all developments and wrote the *User's Manual* and *Programmer's Guide* corresponding to this version.

6.2.1. Asynchronous mode for DIET.

DIET uses the CORBA framework whose standard call is a synchronous object call. This allows the use of a simple synchronization algorithm and error management code but increases the execution time and complicates data persistence.

We defined a new API for asynchronous calls and callback/poll functions to recover a computation result. A callback mechanism allows a client to register to a SeD, and so, it configures the SeD to push results back on the client whenever necessary. The poll mechanism allows a client to get available results on a SeD. All these mechanisms are also included in the GridRPC API implementation. A high level synchronization mechanism (wait and probe functions) is based on the callback and poll mechanisms from Corba.

6.2.2. A peer-to-peer extension for DIET.

We have developed a peer-to-peer extension for DIET using JXTA [68] that allows a dynamic connection of DIET components. JXTA provides functionalities such as passing through firewall and similar network protections, or dynamically discovering other peers. These tools are mandatory to develop a Multi Agents version of DIET using Peer-to-peer technology.

The current implementation of the Multi-MA has been developed with JXTA. This is a prototype of a future powerful Multi-MA version using smart algorithms for discovery. However, connecting Corba components to JXTA is not easy. We can consider that the current JXTA Multi-MA is composed of two parts. In this extension, the client is written in Java. Once the client has received the reference of the server, it connects to it and thus uses JXTA. A part of the integration into DIET implies a cooperation between Java (JXTA) and C++ (DIET). The technology used is JNI (Java Native Interface) which allows Java to call functions written in C++. JNI is located in the Multi-MA code and the JXTA SeD code. The JXTA Multi-MA has to launch and communicate with a C++ Master Agent. The same interface appears in the SeD communication process.

6.2.3. A monitoring software for DIET.

LogService is a monitoring software for DIET. It centralizes system information collected on each Agent/SeD and offers them to concerned tools. LogService is composed of three parts. The first part (LogComponent) deals with collecting log messages on the component side (e.g. DIET agents) and sending them to the monitor core (LogCentral). The second part (LogCentral) connects components and tools by offering APIs for both sides. It gathers and merges incoming messages and offers them to connected tools. The third part (LogTool) is on the tool side (e.g. VizDiet [3]) to deliver incoming log messages from LogCentral. In this distributed approach all logs messages must be sorted, so all monitoring tools connected will receive ordered logs. A clock synchronization is done for each component (modification of messages timestamp). All logs have a tag field which indicates the type of the log. Tags can be defined using a configuration file on the LogCentral side.

6.2.4. New applications into DIET.

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

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

Bioinformatic service For the GriPPS project [4] we have developed the prototype of a DIET bioinformatic server. This server is able to run the *pattinprot* remotely. This algorithm allows a user to scan a protein databases (PROSITE) to match with one or more protein patterns (regular expressions). Based on biological criteria, *pattinprot* is able to select biologically relevant but non perfectly matching proteins.
We have also developed a web service to submit *pattinprot* jobs to DIET servers. The web interface allows users to submit *pattinprot* jobs in a simple manner without requiring any knowledge of DIET.

These two prototypes have been presented during the DIET demo at IPDPS'2003.

[3] A graphic Java tools to visualize the current state of a DIET platform.
[4] Grid Protein Pattern Scanning

6.2.5. Performance Forecasting.

One of the features missing in the FAST library was the ability to predict the performance of parallel routines. Our approach is to mix the forecast of sequential parts and the forecast of communication in analytic models of the routines obtained by source code analysis.

The current version models some parallel routines of the ScaLAPACK library. For each of them, the analysis phase consists in determining first which sequential routines of the BLAS library are called and what are their calling parameters (data sizes, coefficients, transpositions, etc.). The second phase consists in determining the communication scheme and the communication amount. FAST is then used to instantiate the model by predicting the communication time and computation time of the sequential parts.

Several experimentations allowed us to validate the accuracy of this extension, both for different grid shapes and for different data sizes on an homogeneous platform. We plan to extend this work to the ScaLAPACK functions in order to get complete performance evaluation for a dense linear algebra tool.

In order to ease the use of the FAST library in a grid middleware, we ported the library to three new operating systems (BSD, Mac OS X, and Solaris, in addition to Linux). We also improved the overall stability and internal documentation. This work eased the integration of FAST to the DIET infrastructure, which was realized at the same time.

Finally, we started to investigate how to provide more qualitative informations about the network (such as its topology) to the quantitative vision (of its end-to-end latency and bandwidth) already provided by FAST.

6.2.6. Automatic Deployment.

One of the features needed in the DIET environment is the ability automatically adapt its architecture to the changes of the target platform (addition of new servers, new clients, changes in the network performance). This adaptation has to be done at run-time without restarting the whole platform. For each DIET component, and depending on the information stored in the performance database managed by FAST (see Section 5.1.3), we can take deployment decisions to optimize the mapping on Grid resources. Moreover, the co-scheduling capability of DIET and its efficiency depend on the deployment of each scheduler.

The number of parameters that the automatic deployment must take into account is very large. We use two approaches. First, we reuse results about steady-state scheduling from our team. This implies some changes to the model. Then simulation (using SimGrid) can be used to evaluate different strategies with some fixed parameters.

We use theoretical steady-state scheduling framework to propose a new model that enables to find bottlenecks in the organization of a hierarchical NES such as the DIET middleware. This model enables to improve the overall performance of PSE by breaking these bottlenecks and therefore to perform automatic deployment or redeployment. We plan to improve the presented algorithm to take into account the data movement in the system.

We used the SimGrid simulation toolkit. It provides an excellent framework for setting up a simulation where decisions are taken by a single scheduling process. However, in grid computing systems like DIET, the scheduler is distributed on each agent without a global scheduling mechanism. Moreover, the number of schedulers is not fixed in advance. In a deployment or re-deployment case, new agents can be added at runtime and different scheduling politics can be used. Each scheduler is not static and can take a decision according to the current task or the state of the system. It is not required to know in advance where each task must be sent. The scheduler policy depends upon the measured performance of the servers. To simulate DIET in SimGrid, we made a model of each DIET process as a SimGrid process.

6.2.7. Algorithms for mixed-parallelism.

Mixed-parallelism, the combination of data- and task-parallelism is a powerful way of increasing the scalability of entire classes of parallel applications. Exploiting both types of parallelism simultaneously makes it possible to deploy these applications on platforms comprising multiple clusters, which have become increasingly popular in the last decade. However, high performance application executions are possible only if effective scheduling strategies are available. While multi-cluster platforms are predominantly heterogeneous, previous work on scheduling applications with mixed parallelism targeted only homogeneous platforms.

We developed a method for extending existing scheduling algorithms for task-parallel applications on heterogeneous platforms to the mixed-parallel case. We generated a mixed-parallel version of the popular HEFT scheduling algorithm, which we evaluated with an extensive set of simulation experiments.

6.3. Parallel Direct Solvers for Sparse Systems of Linear Equations

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

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

6.3.1. Extension of the software platform MUMPS.

Software work including maintenance, support of users and validation of new functionalities has been pursued this year to provide a reliable software platform, MUMPS, to the scientific academic and industrial community. This includes various bug fixes, the integration of research work concerning the optimization of the memory usage of the solver, the computation of matrix inertia, better parallel performance, a new mechanism to exchange workload informations between processors, an improved management of SMP platforms, improvements to the user's guide, etc.

This work has led to the release of a new version of the package (called MUMPS 4.3) in July 2003, for which we already have a few hundred users. More information on MUMPS itself is available in the software section of this report.

6.3.2. Memory-based scheduling strategies.

We are continuously working on designing improved scheduling strategies to optimize the performance of parallel direct solvers on various types of architectures. This year, after a study of the impact of classical reordering techniques on the memory usage of the parallel multifrontal method, we have started to work on the design of memory-aware scheduling strategies. The memory usage and scalability of sparse direct solvers can effectively be the bottleneck to solve very large problems. We have shown that dynamic strategies have a good potential to improve the memory usage, especially when combined with static modifications of the task dependency graph.

6.3.3. Implicit out-of-core approaches.

We have proposed a new way to improve performance of the factorization of large sparse linear systems which cannot fit in memory. Instead of rewriting a large part of the code to implement an out-of-core algorithm with explicit IO, we modify the paging mechanisms in such a way that IO are transparent. This modification is done thanks to the software tool MMUM&MMUSSEL developed by O. Cozette and G. Utard at LARIA which allows the management of the paging activity at the application level. We designed a first paging policy that is well adapted for the parallel multifrontal solver MUMPS. The results obtained are promising for the factorization step and we now plan to focus on the solution step.

Notice that the approaches described in this paragraph and in the previous one are complementary in the goal of solving problems leading to a large memory usage, possibly larger than the physical memory of the target computer.

6.3.4. Experimentation on real-life test problems.

Thanks to the software work done in the context of the MUMPS package we are able to answer to new needs of new users. We have informal collaborations around MUMPS with a number of institutions: (i) industrial teams who experiment and validate our package, (ii) research teams with whom we discuss new functionalities they would need, (iii) designers of finite element packages who integrate MUMPS as a solver for the linear systems arising, (iv) teams working in optimization, (v) physicists, chemists, etc., in various fields where robust and efficient solution methods are very critical for the results of their simulations. It should also be noted that all our research and algorithmic studies are validated on large-scale industrial problems, either coming directly from MUMPS users, or from standard collections of sparse matrices now in the public domain (Rutherford-Boeing and PARASOL).

6.3.5. Grid enabled sparse direct solvers and GRID TLSE project.

We have designed an interface allowing (sequential) sparse direct solvers to be used as an application of the middleware tools (DIET) developed in the team. This allows a solver to be called remotely from a client. One goal is to extend this work to the parallel case, allowing external users to experiment the MUMPS software platform (for example) on their favorite problems remotely.

Such work is also used in the context of the GRID TLSE project [85][40][71] coordinated by ENSEEIHT-IRIT whose goal is to design an expert site providing a one-stop shop for users of sparse matrix software. A user will be able to interrogate databases for information and references related to sparse linear algebra, and will also be able to obtain actual statistics from runs of a variety of sparse matrix solvers on his/her own problem. Each expertise request leads to a number of elementary requests on the grid for which the middleware tools developed by ReMaP are used.

A first prototype of the expert site exists and we are currently working on the specification of the final site, keeping in mind that including new services (new scenarii of expertise, new solvers with new parameters) should be as automatic as possible.

7. Contracts and Grants with Industry

7.1. Contract LaRIA/CETMEF

Participants: Abou Guermouche, Jean-Yves L'Excellent, Gil Utard.

This is a grant for the study of large solvers with the CETMEF (Centre d'Études Techniques Maritimes et Fluviales), a governemental agency for hydrography prediction. The CETMEF designed a new swell simulation model called REFONDE for computer aided design of harbour developments. It is a finite elements model which leads to solve large sparse system. For instance, the simulation model of the "Le Havre" harbour requires the factorization of a several Gigabytes system. We are using MUMPS to solve such systems and we are investigating out-of-core issues using a version with optimized paging activity. Another direction under discussion is to use the DIET platform for the solution part of the REFONDE simulation model. This work is done in collaboration with the *Laboratoire de Recherche en Informatique d'Amiens* (LaRIA).

8. Other Grants and Activities

8.1. Regional Projects

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

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

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

8.1.2. Pôle Scientifique et de Modélisation Numérique (PSMN)

This federation of laboratories aims at sharing parallel machines and experiences of parallelization of applications. We are involved in this project at different levels, from the choice of new hardware platforms to the assistance for the parallelization of applications.

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

8.2. National Contracts and Projects

8.2.1. Ministry Grant: RNRT VTHD++, 2 years, 2001-2003

E. Caron and F. Desprez participated between 1999 and 2000 to the RNTL project VTHD [5] whose aim was to connect several research centers (and their clusters) in France through a high speed network at 2.5 Gb/s. Several research projects have been completed at different levels (network management, middleware, and applications). One of the target applications was the first version of the DIET environment.

Following this project, the VTHD++ started in 2001. Our goal is now to test several protocols of quality of service and security using the last version of the DIET platform. We also study the scalability of our environment and we port several applications.

F. Desprez leads the application part of both projects.

8.2.2. Ministry Grant: RNTL GASP, 2 years, 2001-2003

F. Desprez leads the RNTL GASP project (Grid Application Service Provider) whose aim is to develop a toolbox (DIET) for the deployment of applications in an ASP mode on a Grid platform. Our partners are the ARES project from INSA Lyon, the ALGORILLE project from LORIA, the SDRP research team from LIFC, the IRCOM laboratory, the “laboratoire des Sciences de la terre” of ENS Lyon, and Sun Labs Europe.

E. Caron, J.-Y. L’Excellent, and G. Utard also participate to this project.

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

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

E. Caron also participates to this project.

8.2.4. Ministry Grant: ACI Grid CGP2P, 3 years, 2002-2005 : Calcul Global peer-to-peer

The ACI CGP2P is a national software project, which aims at providing several software components devoted to large scale peer-to-peer computation and storage. It is a collaborative project involving several computer science laboratories: LRI-LAL-ASCI (Orsay), LIFL (Lille), LaRIA (Amiens), IMAG (Grenoble), LIP (Lyon). The whole project is coordinated by F. Cappello (LRI). G. Utard is leading the subproject concerning storage.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The main goal of this project is to specify, design, implement, and evaluate a data sharing service for mutable data and integrate it into DIET. This service will be built using the generic JuxMem[6]. 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.

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

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

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

8.2.11. CNRS Grant: AS Méthodologie de programmation des grilles

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

F. Vivien participates to this project.

8.3. International Contracts and Projects

8.3.1. INRIA Associated Team

In 2003, we obtained a grant from INRIA to set an associated team with the Grid Research And Innovation Laboratory (GRAIL) of the University of California, San Diego. Our aim is to work on scheduling for heterogeneous and Grid platforms in collaboration with researchers from GRAIL. We plan to have several exchanges of researchers and students during the next 2 years and to organize a workshop at the end of the project.

The DIET software from ReMaP will be used to validate some of the scheduling heuristics and SimGrid will be used to simulate our platform.

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

This project aims at developing robust parallel preconditioners for the solution of large systems of equations. We provide direct methods and are interested in the parallelization and memory reduction aspects of these solvers.

We collaborate with the INRIA projects ALADIN and ScAlApplix, ENSEEIHT-IRIT (Toulouse, France), the University of Minnesota, the University of Indiana and the Lawrence Berkeley Laboratory (NERSC).

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

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

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

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

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

8.3.5. Franco-Bavarian Project, TUM Munich, Germany

We started a collaboration with the LRR lab. (Lehr- und Forschungseinheit Informatik X) from the University of Munich [7]. The BFHZ (Bayreisch-Französisches Hochschulzentrum) finances travel between Lyon and Munich for further coordination of our activities. One of the main goals of the present cooperation is to intensify exchange of students (*Diplomarbeiten*/*Stages MIM2*) and researchers working on projects of common interest.

At the LRR, studies are led on high performance computational biology within the framework of the ParBaum project. Researchers from the LRR have designed PAxML, which is a program for parallel and distributed phylogenetic (evolutionary) tree inference that requires a large amount of computational ressources. Thus, we work on the integration of PAxML into DIET in order to be able to calculate huge evolutionary trees containing over 1000 sequences.

We are also working on the design of a monitoring and vizualisation tool for the DIET system based on MIMO (MIddleware MOnitoring System) [81] developed at the LRR-TUM.

9. Dissemination

9.1. Scientific Missions

CoreGrid Yves Robert is co-chairing (with B. Plateau) the RTP Grilles: this is a CNRS consortium of all the major computer science teams involved in Grid computing in France. He is also coordinating the participation of the RTP to the CoreGrid proposal for a EC Network of Excellence.

Grid’5000 Yves Robert is coordinating the Grid’5000 committee, a gathering of several experts who are in charge of producing a report for the French department of education. The report will (try to) assess the interest of building and running a large-scale heterogeneous platform, distributed over ten sites all over France (with a 500 PC cluster per site, hence the name Grid’5000). The specificity of this platform is that it will be devoted to research and experiments, as opposed to the so-called production grids.

9.2. Animation Responsibilities

STIC department of CNRS. Yves Robert is co-chairing the interdisciplinary network *Calcul à hautes performances et calcul réparti* (High performance and distributed computing).

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

[7] http://wwwbode.cs.tum.edu/~stamatak/lrr-lip.html

9.3. Edition and Program Committees

Frédéric Desprez is an associate editor of *Parallel and Distributed Computing Practices* (http://www.cs.okstate.edu/~pdcp) and *Computing Letters* (COMPULETT).
F. Desprez participated to the program committees of EuroPAR'03 and Euro PVM-MPI'03.
F. Desprez has organized the workshop *Innovative Solution for the Grid* (InnoGrid) during the ICCS Conference in Melbourne, Australia.

Jean-Yves L'Excellent will be member of the program committee of SC'2004 (Supercomputing 2004), Pittsburgh, USA.

Yves Robert is an associate editor of *IEEE Transactions on Parallel and Distributed Systems*. He is a member of the editorial board of the *International Journal of High Performance Computing Applications* (Sage Press).
Y. Robert participated to the following program committees: EuroPDP'03 (European Symposium on Parallel and Distributed Processing), Genoa, Italy; HCW'03 (IEEE Heterogeneous Computing Workshop), Nice, France; Euro PVM-MPI 2003, Venezia, Italy.
Y. Robert was vice-chair (topic Algorithms) of the program committee of IPDPS'03 (IEEE International Parallel and Distributed Processing Symposium), Nice, France. He was general chair of the workshop *Scheduling and load balancing* of EuroPar'03, Klagenfurt, Austria. He will be the program chair of the HCW'2004 workshop (IEEE Heterogeneous Computing Workshop), Santa Fe, USA. He will be vice-chair (topic Algorithms) of the program committee of SC'2004 (Supercomputing 2004), Pittsburgh, USA.

Gil Utard is member of the steering committee of the RenPar conference series (*Rencontres Francophones du Parallélisme*).
G. Utard was member of the following program committees: RenPar'03, IEEE Conference on Local Computing Networks (LCN'2003), and Workshop on High Speed Local Networks (HSLN'2003).

Frédéric Vivien was member of the program committee of PPoPP'03 (ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming).

9.4. Administrative and Teaching Responsibilities

9.4.1. Administrative Responsibilities

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

9.4.2. Teaching Responsibilities

DEA d'informatique de Lyon. Y. Robert gave lectures on "Advanced algorithmics" in the Master of École normale supérieure de Lyon (http://www.ens-lyon.fr/DIF/).

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

10. Bibliography

Books and Monographs

[1] A. LEGRAND, Y. ROBERT. *Algorithmique Parallèle – Cours et exercices corrigés.* Dunod, 2003.

Doctoral dissertations and "Habilitation" theses

[2] A. LEGRAND. *Algorithmique parallèle hétérogène et techniques d'ordonnancement : approches statiques et dynamiques.* Ph. D. Thesis, École normale supérieure de Lyon, December, 2003.

[3] M. QUINSON. *Découverte automatique des caractéristiques et capacités d'une plate-forme de calcul distribué.* Ph. D. Thesis, École normale supérieure de Lyon, December, 2003.

Articles in referred journals and book chapters

[4] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT, X. S. LI. *Impact of the Implementation of MPI Point-to-Point Communications on the Performance of Two General Sparse Solvers.* in « Parallel Computing », number 7, volume 29, 2003, pages 833–847.

[5] 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 », 2003, to appear.

[6] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Scheduling strategies for mixed data and task parallelism on heterogeneous clusters.* in « Parallel Processing Letters », number 2, volume 13, 2003.

[7] O. BEAUMONT, A. LEGRAND, Y. ROBERT. *Scheduling divisible workloads on heterogeneous platforms.* in « Parallel Computing », volume 29, 2003, pages 1121-1152.

[8] O. BEAUMONT, A. LEGRAND, Y. ROBERT. *The master-slave paradigm with heterogeneous processors.* in « IEEE Trans. Parallel Distributed Systems », number 9, volume 14, 2003, pages 897-908.

[9] E. CARON, F. DESPREZ, M. QUINSON, F. SUTER. *Performance Evaluation of Linear Algebra Routines for Network Enabled Servers.* in « Parallel Computing, special issue on Cluters and Computational Grids for scientific computing (CCGSC'02) », 2003, to appear.

[10] 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 », 2003, to appear.

[11] A. GUERMOUCHE, J.-Y. L'EXCELLENT, G. UTARD. *Impact of reordering on the Memory of a Multifrontal Solver.* in « Parallel Computing », number 9, volume 29, 2003, pages 1191–1218.

[12] F. VIVIEN. *On the Optimality of Feautrier's Scheduling Algorithm.* in « Concurrency and Computation: Practice and Experience », number 11-12, volume 15, September, 2003, pages 1047-1068, special issue on Euro-Par 2002.

Publications in Conferences and Workshops

[13] O. BEAUMONT, A. LEGRAND, Y. ROBERT. *Optimal algorithms for scheduling divisible workloads on heterogeneous systems.* in « HCW'2003, the 12th Heterogeneous Computing Workshop », IEEE Computer Society Press, 2003.

[14] O. BEAUMONT, A. LEGRAND, Y. ROBERT. *Scheduling strategies for mixed data and task parallelism on heterogeneous clusters and grids.* in « PDP'2003, 11th Euromicro Workshop on Parallel, Distributed and Network-based Processing », IEEE Computer Society Press, pages 209-216, 2003.

[15] V. BOUDET, F. DESPREZ, F. SUTER. *One-Step Algorithm for Mixed Data and Task Parallel Scheduling Without Data Replication.* in « Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS'03) », April, 2003, http://dlib.computer.org/conferen/ipdps/1926/pdf/19260041b.pdf.

[16] E. CARON, F. DESPREZ, F. PETIT, V. VILLAIN. *A Hierarchical Resource Reservation Algorithm for Network Enabled Servers.* in « IPDPS'03. The 17th International Parallel and Distributed Processing Symposium », Nice - France, April, 2003.

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

[18] O. COZETTE, C. RANDRIAMARO, G. UTARD. *READ^2: put disks at network level.* in « CCGRID'03, Workshop on Parallel IO », Tokyo (Japan), Mai, 2003.

[19] S. GENAUD, A. GIERSCH, F. VIVIEN. *Load-Balancing Scatter Operations for Grid Computing.* in « Proceedings of the 12th Heterogeneous Computing Workshop (HCW'2003) », IEEE Computer Society Press, April, 2003.

[20] A. GIERSCH, Y. ROBERT, F. VIVIEN. *Scheduling tasks sharing files on heterogeneous clusters.* in « 10thEuropeanPVM/MPI: Recent Advances in Parallel Virtual Machine and Message Passing Interface », series LNCS 2840, Springer Verlag, pages 657-660, 2003.

[21] A. GIERSCH, Y. ROBERT, F. VIVIEN. *Scheduling tasks sharing files on heterogeneous master-slave platforms.* in « Proceedings of the 12-th Euromicro Conference on Parallel, Distributed and Network based Processing », IEEE Computer Society Press, 2004, To appear.

[22] A. GUERMOUCHE. *Impact de l'ordonnancement sur l'occupation mémoire d'un solveur multifrontal parallèle.* in « 15^e Rencontres Francophones en Parallélisme, La Colle sur Loup, France », M. AUGUIN, F. BAUDE, D. LAVENIER, M. RIVEILL, editors, pages 37-45, 2003.

[23] A. GUERMOUCHE, J.-Y. L'EXCELLENT, G. UTARD. *On the memory Usage of a Parallel Multifrontal Solver.* in « Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS'03) », 2003.

[24] A. LEGRAND, H. RENARD, Y. ROBERT, F. VIVIEN. *Application et équilibrage de charge pour calculs itératifs sur grappes hétérogènes.* in « 15^e Rencontres Francophones en Parallélisme, La Colle sur Loup, France », M. AUGUIN, F. BAUDE, D. LAVENIER, M. RIVEILL, editors, pages 9 - 17, October, 2003.

[25] A. LEGRAND, H. RENARD, Y. ROBERT, F. VIVIEN. *Load-balancing iterative computations on heterogeneous clusters with shared communication links.* in « PPAM-2003: Fifth International Conference on Parallel Processing and Applied Mathematics », series LNCS, Springer Verlag, 2003, To appear.

[26] A. LEGRAND, H. RENARD, Y. ROBERT, F. VIVIEN. *Mapping and load-balancing iterative computations on heterogeneous clusters.* in « 10th European PVM/MPI: Recent Advances in Parallel Virtual Machine and Message Passing Interface », series LNCS 2840, Springer Verlag, pages 586-594, 2003.

[27] H. RENARD, Y. ROBERT, F. VIVIEN. *Static load-balancing techniques for iterative computations on heterogeneous clusters.* in « Euro-Par 2003: International Conference on Parallel Processing », series LNCS 2790, Springer Verlag, pages 148-159, 2003, (Distinguished paper).

[28] A. VERNOIS. *Pérennité dans les systèmes de stockage pair à pair.* in « 15e Rencontres Francophones en Parallélisme, La Colle sur Loup, France », M. AUGUIN, F. BAUDE, D. LAVENIER, M. RIVEILL, editors, pages 153-160, October, 2003.

Internal Reports

[29] P. R. AMESTOY, I. S. DUFF, J. KOSTER, J.-Y. L'EXCELLENT. *MUltifrontal Massively Parallel Solver (MUMPS Version 4.3) Users' guide.* Users' guide, July, 2003, http://graal.ens-lyon.fr/MUMPS/doc.html.

[30] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Optimizing steady-state throughput of broadcasts on heterogeneous platforms.* Technical report, number 2003-34, LIP, ENS Lyon, France, June, 2003.

[31] O. BEAUMOUNT, H. CASANOVA, A. LEGRAND, Y. ROBERT, Y. YANG. *Scheduling Divisible Loads on Star and Tree Networks: Results and Open Problems.* Technical report, number 2003-41, LIP, ENS Lyon, September, 2003.

[32] E. CARON, P. K. CHOUHAN, A. LEGRAND. *Automatic Deployment for Hierarchical Network Enabled Server.* Technical report, number RR2003-51, Laboratoire de l'Informatique du Parallélisme (LIP), November, 2003, http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2003/RR2003-51.ps.gz.

[33] A. GIERSCH, Y. ROBERT, F. VIVIEN. *Scheduling tasks sharing files from distributed repositories.* research report, number 2003-49, LIP, ENS Lyon, October, 2003.

[34] A. LEGRAND, L. MARCHAL, Y. ROBERT. *Optimizing steady-state throughput of scatter and reduce on heterogeneous platforms.* Technical report, number 2003-33, LIP, ENS Lyon, France, June, 2003.

[35] A. LEGRAND, F. MAZOIT, M. QUINSON. *An Application-Level Network Mapper.* Technical report, number 2003-09, LIP, ENS Lyon, February, 2003.

[36] A. LEGRAND, M. QUINSON. *Automatic deployment of the Network Weather Service using the Effective Network View.* Technical report, number 2003-42, LIP, ENS Lyon, September, 2003.

Bibliography in notes

[37] V. H. ALLAN, R. B. JONES, R. M. LEE, S. J. ALLAN. *Software pipelining.* in « ACM Computing Surveys », number 3, volume 27, September, 1995, pages 367–432.

[38] 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 », number 1, volume 23, 2001, pages 15-41.

[39] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT. *Multifrontal Parallel Distributed Symmetric and Unsymmetric Solvers.* in « Comput. Methods Appl. Mech. Eng. », volume 184, 2000, pages 501–520.

[40] P. AMESTOY, M. PANTEL. *Grid-TLSE: A Web expertise site for sparse linear algebra.* June 10-13, 2003, http://www.enseeiht.fr/lima/tlse/grid_tlse.pdf, Conference : Sparse days and Grid Computing, St Girons (France).

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

[42] M. BAKER. *Cluster Computing White Paper.* 2000.

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

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

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

[46] J. CHOI, J. DEMMEL, I. DHILLON, J. DONGARRA, S. OSTROUCHOV, A. PETITET, K. STANLEY, D. WALKER, R. WHALEY. *LAPACK Working Note: ScaLAPACK: A Portable Linear Algebra Library for Distributed Memory Computers - Design Issues and Performances.* Technical report, number 95, UT, 1995.

[47] J. CHOI, J. DONGARRA, S. OSTROUCHOV, A. PETITET, D. WALKER, R. C. WHALEY. *A proposal for a set of parallel basic linear algebra subprograms* . in « Applied parallel computing: computations in physics, chemistry, and engineering science: second international workshop, PARA '95 », series Lecture Notes in Computer Science, volume 1041, Springer-Verlag, J. DONGARRA, K. MADSEN, J. WASNIEWSKI, editors, pages 107–114, 1996.

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

[49] CONDOR-G. http://www.cs.wisc.edu/condor/condorg/.

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

[51] A. DENIS, C. PÉREZ, T. PRIOL. *Achieving Portable and Efficient Parallel CORBA Objects.* in « Concurrency and Computation: Practice and Experience », 2002.

[52] D. DHOUTAUT, D. LAIYMANI. *A CORBA-Based Architecture for Parallel Applications: Experimentations with the WZ Factorization.* in « ACM Applied Computing Review », number 1, volume 10, 2002.

[53] DIET: http://graal.ens-lyon.fr/DIET.

[54] J. DONGARRA, E. D'AZEVEDO. *The Design and Implementation of the Parallel Out-of-core ScaLAPACK LU, QR, and Cholesky Factorization Routines.* Technical report, number UT-CS-97-347, Department of Computer Science, University of Tennessee, January, 1997.

[55] I. S. DUFF, J. K. REID. *The Multifrontal Solution of Indefinite Sparse Symmetric Linear Systems.* in « ACM Transactions on Mathematical Software », volume 9, 1983, pages 302-325.

[56] I. S. DUFF, J. K. REID. *The Multifrontal Solution of Unsymmetric Sets of Linear Systems.* in « SIAM Journal on Scientific and Statistical Computing », volume 5, 1984, pages 633-641.

[57] H. EL-REWINI, H. H. ALI, T. G. LEWIS. *Task Scheduling in Multiprocessing Systems.* in « Computer », number 12, volume 28, 1995, pages 27–37.

[58] FAST: http://graal.ens-lyon.fr/FAST.

[59] 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.

[60] M. FERRIS, M. MESNIER, J. MORI. *NEOS and Condor: Solving Optimization Problems Over the Internet.* in « ACM Transactions on Mathematical Sofware », number 1, volume 26, 2000, pages 1-18, http://www-unix.mcs.anl.gov/metaneos/publications/index.html.

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

[62] C. GERMAIN, G. FEDAK, V. NÉRI, F. CAPPELLO. *Global Computing Systems.* in « Lecture Notes in Computer Science », volume 2179, 2001, pages 218–227.

[63] GLOBUS. http://www.globus.org/.

[64] C. GOMEZ, editor, *Engineering and Scientific Computing with Scilab.* Birkhaüser, 1999.

[65] R. GRAHAM, E. CHOI, D. DANIEL, N. DESAI, R. MINNICH, C. RASMUSSEN, L. RISINGER, M. SUKALSKI. *A Network-Failure-Tolerance Message-Passing System For Terascale Clusters.* in « ICS'02 »,

ACM, New York, USA, June, 2002.

[66] S. GRIDENGINE. http://wwws.sun.com/software/gridware/.

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

[68] JXTA. http://www.jxta.org/.

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

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

[71] J.-Y. L'EXCELLENT. *Sparse direct solvers and grid computing.* September 22-26, 2003, http://www.enseeiht.fr/lima/tlse/bordeaux.pdf, Journées Bordelaises de formation GRID2 : Applications, Algorithmique et Ordonnancement pour la grille, Bordeaux (France).

[72] J. W. H. LIU. *The Role of Elimination Trees in Sparse Factorization.* in « SIAM Journal on Matrix Analysis and Applications », volume 11, 1990, pages 134–172.

[73] S. MATSUOKA, H. NAKADA, M. SATO, S. SEKIGUCHI. *Design Issues of Network Enabled Server Systems for the Grid.* 2000, Grid Forum, Advanced Programming Models Working Group whitepaper.

[74] MICROSOFT. *Distributed Component Object Model Technical Overview.* 1997, http://msdn.microsoft.com/library/default.asp?url=/libr
us/dndcom/html/msdn_dcomtec.asp.

[75] S. MICROSYSTEMS. *Java Remote Method Invocation.* 2003, http://java.sun.com/products/jdk/rmi.

[76] http://graal.ens-lyon.fr/MUMPS.

[77] 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 », series Lecture Notes in Computer Science, number 2536, pages 274-278, Baltimore, MD, USA, November, 2002.

[78] H. NAKADA, M. SATO, S. SEKIGUCHI. *Design and Implementations of Ninf: towards a Global Computing Infrastructure.* in « Future Generation Computing Systems, Metacomputing Issue », number 5-6, volume 15, 1999, pages 649-658.

[79] M. G. NORMAN, P. THANISCH. *Models of Machines and Computation for Mapping in Multicomputers.* in « ACM Computing Surveys », number 3, volume 25, 1993, pages 103–117.

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

[81] G. RACKL, M. LINDERMEIER, M. RUDORFER, B. SUSS. *MIMO - An Infrastructure for Monitoring and Managing Distributed Middleware Environments.* in « Middleware 2000 – IFIP/ACM International Conference on Distributed Systems Platforms », series Lecture Notes in Computer Science, volume 1795, Springer, pages 71-87, 2000.

[82] 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 », volume 2104, 2001, pages 130–136.

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

[84] SimGrid: http://grail.sdsc.edu/simgrid.

[85] *http://www.enseeiht.fr/lima/tlse/.*

[86] *Trading Object Service Specification.* Object Management Group, 97, http://www.omg.org, ORBOS/97-07-26.

[87] R. C. WHALEY. *Installing and testing the BLACSv1.1.* Technical report, Department of Computer Science, University of Tennessee, May, 1997, http://www.netlib.org/blacs/blacs_install.ps.

[88] 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 », number 5–6, volume 15, October, 1999, pages 757–768.

[89] OBJECT MANAGEMENT GROUP. *CORBA Basics.* 2003, http://www.omg.org/gettingstarted/corbafaq.htm.