INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team AlGorille

Algorithms for the Grid

Lorraine

THEME 1B

Table of contents

1. Team

AlGorille is a team at the Laboratoire lorrain de recherche en informatique et ses applications (LORIA) in common with Centre National de Recherche Scientifique (CNRS), Institut National de Recherche en Informatique et Automatique (INRIA), Université Henri Poincaré Nancy 1 (UHP), Université Nancy 2 and Institut National Polytechnique de Lorraine (INPL).

Head of Project-Team
Jens Gustedt [research director, INRIA]

Administrative Assistant
Josiane Reffort [UHP]

Staff Member
Johanne Cohen [research fellow, CNRS]
Emmanuel Jeannot [associate professor, UHP]

Teaching Assistant
Mohamed Essaïdi [UHP, since October '03]

Ph. D. Student
Yves Caniou [joint regional/INRIA grant]
Mohamed Essaïdi [Tunisian grant until 30/9/03]
Frédéric Wagner [joint regional/INRIA grant]

Student Intern
Oissem Ben Fradj [DEA Lorraine]
Lynda Gastal [DEA Univ. Versailles]
Fei Yin [DEA Lorraine]

2. Overall Objectives

Key words: *algorithms*, *Grid computing*, *data distribution*, *data redistribution*, *parallel and distributed computing*, *scheduling*.

The possible access to distributed computing resources on the Internet allows for a new type of applications that use the power of the machines and the network. The transparent and efficient access to distributed resources that form The Grid is one of the major challenges of information technology. It needs the implementation of special techniques and algorithms to make computers communicate with each other, let applications work together, allocate resources and improve the quality of service and the security of the transactions.

Challenge: The new INRIA team "Algorithms for The Grid" (AlGorille) at the LORIA tackles several problems related to the first of the major "challenges" that INRIA has identified in its strategic plan: To master digital infrastructures by being capable of programming, computing, and communicating on the Internet and on heterogeneous networks.

Research themes: We have identified two specific research themes:

- Transparent resource management: task scheduling; migration of computations; data exchange, distribution and redistribution.
- Structuring of applications for scalability: modeling of locality and granularity.

Methods: Our methodology is based upon three points (1) modeling, (2) design and (3) engineering of algorithms. These three points interact strongly to form a validation cycle.

i. With models we obtain an abstraction of the physical, technical or social reality.

ii. This abstraction allows us to design techniques for the resolution of specific problems.

iii. These techniques are implemented to validate the models with experiments and by applying them to real world problems.

3. Scientific Foundations

3.1. Transparent resource management

Participants: Johanne Cohen, Emmanuel Jeannot, Yves Caniou, Frédéric Wagner.

Key words: *data redistribution*, *parallel and distributed computing*, *scheduling*, *approximating algorithms*.

We think of the future Grid as of a medium to access resources. This access has to be as transparent as possible to a user of such a Grid and the management of these resources must not be imposed to him, but entirely done by a "system", in our language called *middleware*. The middleware must be able to manage all resources in a satisfactory way. Currently, numerous *algorithmic* problems hinder such an efficient resource management and thus the transparent utilization of the Grid.

By their nature, distributed applications use different types of resources; the most important being those of computing power and network connections. The management and optimization of these resources is essential for networking and computing on Grids. This optimization may be necessary at the level of the computation of the application, of the organization of the underlying interconnection network or for the organization of the messages between the different parts of the application. Managing these resources relates to a set of *politics* to optimize their utilization and for allowing an application to be executed under favorable circumstances.

Our approach consists of the tuning of techniques and algorithms for a transparent management of resources, be they data, computations, networks, ... This approach has to be clearly distinguished from others which are more focused towards applications and middlewares. We want to propose new algorithms (or improve the exiting ones) *for* the resource management in middlewares; the objective being to provide these algorithms in libraries such that they may be easily integrated. For example, we will propose algorithms to efficiently transfer data (data compression, distribution or redistribution of data) or for scheduling.

The problems that we are aiming to solve are quite complex. Therefore they often translate into combinatorial or graph theoretical problems where the identification of an optimal solution is known to be hard. But, the classical measures of complexity (polynomial versus NP-hard) are not very satisfactory for really large problems: even if a problem has a polynomial solution it is often infeasible in reality whereas on the other hand NP-hard problems may allow for quite efficient resolution with results close to optimality.

As a consequence it is necessary to study approximation techniques where the objective is not to impose global optimality constraints but to relax them in favor of a compromise. Thereby we hope to find *good* solutions for a *reasonable* price. But, these can only be useful if we know how to analyze and evaluate them.

3.2. Structuring of applications for scalability

Participants: Mohamed Essaïdi, Jens Gustedt, Emmanuel Jeannot.

Key words: *models for parallel and distributed computing*, *performance evaluation message passing*, *shared memory*.

Our approach is based on a "good" separation of the different problem levels that we encounter for Grid problems. Simultaneously this has to ensure a good data locality (a computation will use data that is "close") and a good granularity (the computation is divided into non preemptive tasks of reasonable size). For problems for which there is no natural data parallelism or control parallelism such a division (into data and tasks) is indispensable when tackling the difficulties that are related to spatial and temporal distances as we encounter them in the Grid.

Several parallel models that offer simplified frameworks that ease the design of algorithms and their implementation have been proposed. The best known of these provide a modeling that is called "fined grained",

i.e. on the instruction level. Their lack of realism with respect to the existing parallel architectures and their inability to predict the behavior of implementations, has triggered the development of new models that allow a switch to a *coarse grained* paradigm. In the framework of parallel and distributed (but homogeneous) computing they started with the fundamental work of Valiant [24]. Their common characteristics are:

- to maximally exploit the data that is located on a particular node by a local computation,
- to collect all requests for other nodes during the computation, and
- to only transmit these requests if the computation can't progress anymore.

The coarse grained models aim to be realistic concerning two different aspects: algorithms and architectures. In fact, the coarseness of these models uses the common characteristic of parallel settings of today: the size of the input is orders of magnitude larger than the number of processors that are available. In contrast to the PRAM model, the coarse grained models are able to integrate the communication between different processors. This allows them to give realistic predictions about the overall execution time of a parallel program. As examples we refer to BSP (Bulk Synchronous Parallel model) [24], LogP (Latency overhead gap Procs) [21], CGM (Coarse Grained Multicomputer) [23] and PRO (Parallel Resource Optimal Model) [1].

For the architecture, the assumptions are very similar: p homogeneous processors with local memory that are distributed on a point-to-point interconnection network. They also have similar models for program execution that are based on *supersteps*; an alternation of computation and communication phases. For the algorithmics, this takes the distribution of the data to the different processors into account. But, all the mentioned models do not allow the design of algorithms for the Grid since they all assume homogeneity, for the processors as well as for the interconnection network.

Our approach is algorithmic. We try to provide a modeling of a computation on the Grid that allows for an easy design of algorithms and for realistic and performing implementations. Even if there are problems for which the existing sequential algorithms may be parallelized easily, an extension to other more complex problems such as computing on large discrete structures (e.g. web graphs or social networks) is desirable. Such an extension will only be possible if we accept a paradigm change. We have to explicitly decompose data and tasks.

We are convinced that this new paradigm must

- be guided by the idea of **supersteps** (BSP). This is to enforce a concentration of the computation to the local data.
- ensure an economic use of all resources that are available.

On the other hand, we have to be careful that the model (and the design of algorithms) remains simple. The number of supersteps and the minimization thereof should by themselves not be a goal. It must be constraint by other more "natural" parameters coming from the architecture and the problem instance.

A first solution to combine these objectives has been given in [1] with PRO, *Parallel Resource Optimal model*.

Starting from this model, we try to develop high level algorithmics for The Grid. It will be based upon an abstract vision of the architecture and as far as possible be independent of the intermediate levels. It aims to be robust with respect to the different hardware constraints and should be sufficiently expressive. The applications for which our approach will be feasible are those that fulfill certain constraints, such as

- they need a lot of computing power
- they need a lot of data that is distributed upon several resources, or,
- they need a lot of temporary storage which doesn't fit into a single machine.

4. Application Domains

4.1. Evolution of scheduling policies and network protocols

Participants: Johanne Cohen, Emmanuel Jeannot, Yves Caniou, Frédéric Wagner, Fei Yin, Lynda Gastal.

4.1.1. Scheduling on the Grid

Our work concerns algorithms that allocate applications that are divided into tasks onto distant compute servers in an agent-client-server model. A good scheduling of these tasks is a primal requirement to obtain good performance.

We have investigated the limits of the greedy algorithm MCT (Minimum Completion Time) as it is e.g used in NetSolve, see Section 5.1. To improve over it, we have introduced the notion of a "history" that allows for a better prediction of the execution of a task on a particular server.

We also proposed algorithms that minimize the perturbation that is caused by the allocation of some task to a server. This optimization is done by still enforcing a good performance (response time) for the task in question by itself. This system-oriented approach has first been tested via simulations. The best among all the heuristics that we studied have been integrated to NetSolve and studied on a broad scale, see [11].

4.1.2. Redistribution of data between clusters

During computations that are done on clusters of machines it occurs that data has to be shifted from one cluster to another one. Eg. the first and the second cluster may differ in the resources they offer (special hardware, computing power, available software) and that each of the clusters may be more adequate for a different phase of the computation. So then the data must be redistributed from the first cluster to the second and such a redistribution should use the capacities of the underlying network in an efficient way.

This problem of redistribution between clusters generalizes the redistribution problem inside a parallel machine, which already is highly non trivial.

We have modeled this problem by a decomposition of the underlying bipartite graph into certain types of matchings. In general, this problem is NP-hard, as we have been able to show in [12]. So, we are forced to study lower bounds, approximation algorithms and heuristics. Our paper [12] gives some results on heuristics that show a good practical behavior.

4.1.3. Dynamic and adaptive compression of network streams

A commonly used technique to speed up transfer of large data over networks with restricted capacity during a distributed computation is data compression. But such an approach fails to be efficient if we pass to a high speed network, since here the time to compress and uncompress the data dominates the transfer time. So a programmer that wants to be efficient in both cases, would have to provide two different implementations of a network layer of his code, and the user of the program would have to determine which of the variants he has to run to be efficient in a particular case.

In [3] we have given an algorithm that avoids such an expensive and error-prone setting and provides a technique to compress data on the fly, as necessity of a particular execution requires. It covers the time for compression with communication and automatically adapts the effort for compression to the available resources (network and CPU).

This algorithms is implemented in our library AdOC, "Adaptive Online Compression" which has been deposed at the APP.

4.1.4. Evolution of network protocols

4.1.5. Ad-hoc networks

We are interested in ad-hoc networks, because we think that it is possible to propose a more efficient utilization of the communication medium. The signal processing techniques of spatial diversification have allowed to construct new types of antennas (or more precisely groups of antenna) for these types of networks. The

undirected antennas are replaced by directed ones, or, by so-called "smart" ones. Recently, a lot of research has focused on the optimization of the MAC layer for these smart antennas.

4.1.6. Optical networks

Optical networks are very interesting because they realize a very high throughput. After the WDM (Wavelength Division Multiplexing) technology which is currently used the next generation of optical networks will use DWDM (Dense Wavelength Division Multiplexing). This will probably allow for a throughput of several Tbits per second. The resource management of these high speed networks gives rise to very special constraints. The focus of our studies is on routing and load balancing.

4.2. High Performance Computing

Participants: Mohamed Essaïdi, Jens Gustedt, Emmanuel Jeannot.

Our approach is based upon a good decomposition of the problems that are to be handled on the Grid. This decomposition must simultaneously fulfill two objectives. It must ensure a good data locality (a computation should only depend on data that is "close") and a good granularity (the computation is split into non-preemptive chunks of reasonable size). Such a problem decomposition (with respect to data *and* computation) is unavoidable to attenuate the problems that are related to temporal and spatial distances that are imposed by a computation on the Grid.

4.2.1. Models and Algorithms for Coarse Grained Computation

With this work we aim to extend the coarse grained modeling (and the resulting algorithms) to hierarchically composed machines such as clusters of clusters or clusters of multiprocessors.

For being usable in a Grid context this modeling must first of all overcome a principal constraint of the existing models: the idea of an homogeneity of the processors and the interconnection network. Even if the long term goal is to go on to arbitrary architectures it would not be realistic to think to achieve this directly, but in different steps:

- Hierarchical but homogeneous architectures: These are composed of a homogeneous set of processors (or of the same computing power) interconnected with a non-uniform network or bus which is hierarchic (CC-NUMA, clusters of SMPs).
- Hierarchical heterogeneous architectures: there is no established measurable notion of efficiency or speedup. Also most certainly not any arbitrary collection of processors will be useful for computation on the Grid. Our aim is to be able to give a set of concrete indications of how to construct an extensible Grid.

In parallel, we have to work upon the characterization of architecture-robust efficient algorithms, i.e. algorithms that up to a certain degree are independent of low-level components or the underlying middleware.

The literature about fine grained parallel algorithms is quite exhaustive. It contains a lot of examples of algorithms that could be translated to our setting, and we will look for systematic descriptions of such a translation.

4.2.2. External Memory Computation

In the mid-nineties several authors, see [20][22], developed a connection between two different types of models of computation: BSP-like models of parallel computation and IO efficient external memory algorithms. Their main idea is to enforce data locality during the execution of a program by simulating a parallel computation of several processors on one single processor.

Whereas such an approach is convincing on a theoretical level, its efficient and competitive implementation is quite challenging in practice. In particular, it needs software that induces as little computational overhead as possible by itself. Up to now, it seems that this has only been provided by software specialized in IO efficient implementations.

Currently, with our library SSCRAP, see Section 5.2, we reached a level of scalability that let us hope that it could now be possible to attain high efficiency in both (or even mixed) contexts:

- SSCRAP can run hundreds of "processors" (as POSIX threads) on a single machine (main-frame or not) without losing on its performance.
- It can handle problem instances efficiently that may exceed the size of the address space of an individual hardware processor.

4.2.3. Irregular problems

Irregular data structures like sparse graphs and matrices are in wide use in scientific computing and discrete optimization. The importance and the variety of application domains are the main motivation for the study of efficient methods on such type of objects. The main approaches to obtain good results are parallel, distributed and out-of-core computation.

We follow several tracks to tackle irregular problems: automatic parallelization, design of coarse grained algorithms and the extension of these to external memory settings.

In particular we study the possible management of very large graphs, as they occur in reality. Here, the notion of "networks" appears twofold: on one side many of these graphs originate from networks that we use or encounter (Internet, Web, peer-to-peer, social networks) and on the other the handling of these graphs has to take place in a distributed Grid environment. The principal techniques to handle these large graphs will be provided by the coarse grained models. With the model *PRO* (see [1]) and the library SSCRAP we already provide tools to better design algorithm (and implement them afterwards) that are adapted to these irregular problems.

In addition we will be able to rely on certain structural properties of the relevant graphs (short diameter, small clustering coefficient, power laws). This will help to design data structures that will have good locality properties and algorithms that compute invariants of these graphs efficiently.

5. Software

5.1. Integrating Services into Middlewares

Participant: Emmanuel Jeannot.

In collaboration with the INRIA team Graal (previously ReMaP), we contribute to the elaboration of a middleware called DIET (Distributed Interactive Engineering Toolbox). It is designed to discover and to offer grid services to and for *SciLab*|| such that they may be used transparently.

More precisely we work on algorithms for scheduling, data distribution and load balancing for this environment. We also contribute with respect to models and tools that are needed to supervise such a platform and to be able to better describe its actual state.

NetSolve is a programming environment that allows to launch computations on distributed servers which are controlled by an "agent". It originates from the University of Tennessee, Knoxville, in the team of Jack Dongarra. We have worked on interfacing it with *SciLab*||. This allows users of *SciLab*|| to access the available servers via NetSolve. This is particularly useful for parallel servers with low charge.

5.2. SSCRAP

Participants: Mohamed Essaïdi, Jens Gustedt.

SSCRAP is developed to ease the implementation, test and benchmarking of algorithms that are written for the model PRO.

SSCRAP is the prototype of a C++-library that was initially developed together with Isabelle Guérin Lassous from the team ARES.

This library takes the requirements of PRO, see Section 3.2, into account, i.e the design of algorithms in alternating computation and communication steps. It realizes an abstraction layer between the algorithm as it was designed and its realization on different architectures and different modes of communication. The current version of this library is available at http://www.loria.fr/~gustedt/sscrap/, and is now able to integrate

- a layer for message passing with MPI
- a layer for shared memory with POSIX threads, and
- a layer for out-of-core management with file mapping (system call *mmap*).

All three different realizations of the communication layers are quite efficient. They let us execute programs that are otherwise unchanged within the three different contexts such that they reach or maybe outperform programs that are directly written for these contexts.

Due to the instability of the systems that we considered for passing over to heterogeneous environments, we are not yet able to use message passing and shared memory jointly at the same time.

5.3. AdOC

Participant: Emmanuel Jeannot.

The AdOC (Adaptive Online Compression) library implements the AdOC algorithm for dynamic adaptive compression of network streams.

AdOC is written in C and uses the standard library zlib for the compression part. It is realized as an additional layer above TCP and offers a service of adaptive compression for the transmission of program buffers or files. Compression is only used if it doesn't generate an additional cost, typically if the network is slow or the processor is not charged too much. It integrates overlap techniques between compression and communication as well as mechanisms that avoid superfluous copy operations. The send and receive functions have exactly the same semantics as the system calls *read* and *write* so the integration of AdOC into existing libraries and application software is straightforward. Moreover, AdOC is thread-safe.

6. New Results

6.1. AdOC

Participant: Emmanuel Jeannot.

We recently added a new compression algorithm that favors speed against compression ratio. This allows very good performances when dealing with fast network.

We also worked on portability issues. So far, AdOC is working an the following architecture/systems: Linux, FreeBSD, MAC OS X, Solaris, AIX, IRIX.

6.2. Scheduling on the Grid

Participants: Yves Caniou, Emmanuel Jeannot.

We have previously shown, on the basis of real experimentations, interests of heuristics relying on the Historical Trace Manager (HTM) to dynamically schedule independent tasks on a grid platform. The HTM is a time-shared predicting module. We have revisited these heuristics when scheduling with precendence constraints, or mixed submissions of such constraints and some independent tasks. Many experiments corresponding to many scenarios have been executed on a real testbed and they present large gain on the makespan, the sumflow and on the quality of service over the well known MCT heuristic. Moreover, we study the accuracy of the HTM, observed from all undertaken experiments. We show that it is able to provide very precise and useful information, and allows a good environment management.

6.3. Redistribution of Data

Participants: Frederic Wagner, Emmanuel Jeannot.

We have proposed and studied two fast and efficient algorithms for this problem. We prove that these algorithms are 2-approximation algorithms. Simulation results show that both algorithms perform very well compared to the optimal solution. These algorithms have been implemented using MPI. Experimental results show that both algorithms outperform a brute-force TCP based solution, when no scheduling of the messages is performed.

6.4. Large scale experiments for scalability and portability

Participants: Ouissem Ben Fradj, Mohamed Essaïdi, Jens Gustedt.

Now that the communication layer of SSCRAP can handle large numbers of POSIX threads (shared memory) or distributed processes (MPI), we were able to run large scale experiments on mainframes and clusters. These have proven the scalability of our approach as a whole, including engineering, modeling and algorithmic aspects: the algorithms that are implemented and tested show a speedup that is very close to the best possible theoretically, and these speedups are reproducible on a large variety of platforms. See [9] for some of the results.

We also started studying extensions of the communication layer towards heterogeneous architectures and tested two possible directions, see [16]. The first was the use of PM2 as a platform for a transparent mix of shared memory and message passing communication. The second was to re-implement parts in Java and to use ProActive to model communication. Both approaches only had limited success, the first because the chosen library was not stable enough, the second because of the lack of efficiency of the code that was generated by Java.

6.5. External Memory

Participant: Jens Gustedt.

In fact, the stability of SSCRAP also showed in its extension towards external memory computing, see [14]. With some relatively small add-ons to SSCRAP we were able to provide such a framework. It was tested successfully on some typical hardware, PC with some GB of free disk.

The main add-on that was integrated into SSCRAP was a consequent implementation of an abstraction between the *data* of a process execution and the memory of a processor. The programmer acts upon these on two different levels:

- with a sort of *handle* on some data array which is an abstract object that is common to all SSCRAP processors.
- with a map of its (local) part of that data into the address space of the SSCRAP processor, accessible as a conventional pointer.

Another add-on was the possibility to fix a maximal number of processors (i.e. threads) that should be executed concurrently. With these add-ons, simple environment variables `SSCRAP_MAP_MEMORY` and `SSCRAP_SERIALIZE` allow for a runtime control of the program behavior.

6.6. Models for coarse grained computation

Participant: Jens Gustedt.

We also continued the design of algorithms in the coarse grained setting as given by the model PRO [1]. In particular list ranking, tree contraction and graph coloring, see [6].

To work in the direction of understanding of what problems might be “hard” we tackled a problem that is known to be P-complete in the PRAM/NC framework, but for which not much had been known when

only imposing the use of relatively few processors: the *lexicographic first maximal independent set* problem (LFMIS), see [15].

We were able to give a work optimal algorithm in case we have about $\log n$ processors and thus to prove that the NC classification is not necessarily appropriate for today's parallel environments which consist of few processors (up to some thousands) and large amount of data (up to some TeraByte).

7. Contracts and Grants with Industry

7.1. GASP

Participants: Emmanuel Jeannot, Yves Caniou.

The goal of the RNTL GASP is to develop a hierarchical grid middleware called DIET. We participate in the design of the scheduling algorithms inside DIET. We also work on deploying DIET on heterogeneous environments.

8. Other Grants and Activities

8.1. Bilateral international relations and European initiatives

We take part in the NoE "CoreGrid" which is directed by Thierry Priol from INRIA Rennes, for which we hope to have a positive result by the end of this year 2003.

We maintain several international collaborations with other research teams. The two most fruitful are with the team of Jan Arne Telle from Bergen University, Norway, and with the team of Jack Dongarra at the University of Tennessee.

The collaboration with Bergen has been financed by a bilateral French-Norwegian grant and by some regional visiting grant for Jan Arne Telle.

The collaboration with Jack Dongarra of the Univ. of Tennessee and the Graal project of INRIA, has recently been formalized in an INRIA-NSF project which handles the aspects of the integration of our scheduling algorithms into NetSolve.

8.2. National initiatives

8.2.1. CNRS initiatives, GDR-ARP and specific initiatives

We participate at numerous national initiatives. In the GDR-ARP (architecture, networks and parallelism) we take part in TAROT[1], Grappes[2], and RGE[3].

The support for the latter has been augmented by having become a project called ARGE in the national grid initiative in 2001. ARGE had first been guided by André Schaff, and was recently handed over to Jens Gustedt and Stéphane Vialle (Supélec Metz).

Furthermore, we participate in two AS (actions spécifiques – specific initiatives) *Enabling Grid 5000* and *Programming methods for the Grid*. The first is a program that studies the possibilities of enabling a large Grid of several thousand CPUs in France. The second studies more fundamental questions related to Grid computing.

We also participate at a working group about all-optical networks together with the teams Grafcom du LRI (Université Paris-Sud), OpPALL du Prism (Université de Saint-Quentin), Opal du LAMI (Université d'Évry), without these contacts being formalized up to now.

[1] *Techniques algorithmiques, réseaux et d'optimisation pour les télécommunications*
[2] *Architecture, systèmes, outils et applications pour réseaux de stations de travail hautes performances*
[3] *Réseau Grand Est*

8.2.2. ACI initiatives of the French Research Ministry

We are partners in several projects of the ACI GRID initiative from 2001:

- GRID-ASP (client-server approach for computing on the Grid). Within this ARC we are developing an application for the DIET environment called HESP in collaboration with the *Laboratoire de Chimie thèorique* of Univ. H. Poincaré. This application is to distribute the computation of Hyper-surface of potential energy of some molecules.
- Guirlande-fr (handling distributed linguistic resources)
- GRID2 (national animation of the Grid community)
- ARGE (see above)

In the recent (2003) initiative ACI Grid Explorer we participate with a joint proposition together with Stéphane Vialle from Supélec, Metz. We also work on designing a set of emulation tools for transforming an homogeneous platform into an heterogeneous one.

8.2.3. INRIA New Investigation Grant

The goal of the INRIA-ARC Red Grid is to design algorithms and services for the problem of data redistribution between distant clusters. It involves the Paris, Graal, Scalaplix INRIA projects.

9. Dissemination

9.1.1. Leadership within scientific community

On a national level, Jens Gustedt is elected member of INRIA scientific board and a member of the INRIA steering committee VISON[4]. Locally, within LORIA he is appointed member of the commission for scientific prospective, within INPL he is nominated substitute member of the hiring committee in computer science, and within INRIA Lorraine he participated in the hiring committee for the years 1999 to 2003.

Emmanuel Jeannot is an elected member of the computer science hiring committee of UHP. He is also member of the steering committee of the réseau thématique pluridisciplinaire (RTP) (Pluri-disciplinary Thematic Network) "Calcul à hautes performances et calcul réparti" (High Performance and Distributed Computing) of the CNRS STIC Department.

9.1.2. Scientific Expertise

In 2003, Jens Gustedt served as an external expert for the evaluation of scientific projects in regional initiatives for information science and technology in a French region and in a neighboring European country, he reported the thesis of Clémence Magnien, LIX/ENS, and was a member of the habilitation committees for Zineb Habas, University of Metz, and Anne Berry, University of Clermont-Ferrand.

9.1.3. Teaching activities

Emmanuel Jeannot is teaching in the *Algorithme et programmation des systèmes distribués* module of the DEA at Univ. H. Poincaré. He is also teacher of computer science (System, Java, Data Base, C) in the IUT of Univ. H. Poincaré.

9.1.4. Editorial activities

Since October 2001, Jens Gustedt is Editor-in-Chief of the journal *Discrete Mathematics and Theoretical Computer Science (DMTCS)*.

Emmanuel Jeannot is member of the program committees of RenPar'04 and HCW'04, and inside LORIA he participates in the board of "lettre du LORIA".

In 2003, members of the team served as referees for the following journals and conferences:

Discrete Mathematics, IEEE TPDS, JPDC, Opodis, PCMAA, SODA, STACS, PaCT

[4] VISON: Vers un Intranet Sécurisé Ouvert au Nomadisme, towards an secured intranet open to nomadism

10. Bibliography

Major publications by the team in recent years

[1] A. H. GEBREMEDHIN, I. GUÉRIN LASSOUS, J. GUSTEDT, J. A. TELLE. *PRO : a Model for Parallel Resource-Optimal Computation.* in « 16th Annual International Symposium on High Performance Computing Systems and Applications, Moncton, New Brunswick, Canada », IEEE, pages 106-113, June, 2002.

[2] I. GUÉRIN LASSOUS, J. GUSTEDT. *Portable List Ranking : an Experimental Study.* in « ACM Journal of Experimental Algorithmics », number 7, volume 7, July, 2002.

[3] E. JEANNOT, B. KNUTTSON, M. BJORKMAN. *Adaptive Online Data Compression.* in « Eleventh IEEE International Symposium on High Performance Distributed Computing - HPDC 11, Edinburgh, Scotland », IEEE, July, 2002.

Articles in referred journals and book chapters

[4] K. BERTET, J. GUSTEDT, M. MORVAN. *Weak-order extensions of an order.* in « Theoretical Computer Science », number 1-3, volume 304, July, 2003, pages 249-268.

[5] M. COSNARD, E. JEANNOT, T. YANG. *Compact DAG Representation and its Symbolic Scheduling.* in « Journal of Parallel and Distributed Computing (JPDC) », December, 2003.

[6] A. H. GEBREMEDHIN, I. GUÉRIN LASSOUS, J. GUSTEDT, J. A. TELLE. *Graph Coloring on a Coarse Grained Multicomputers.* in « Discrete Applied Mathematics », number 1, volume 131, September, 2003, pages 179-198.

Publications in Conferences and Workshops

[7] D. BARTH, J. COHEN, C. DURBACH. *Algorithmes de répartition de charge pour des simulations distribuées.* in « 5ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision - ROADEF'2003, Avignon, France », February, 2003.

[8] D. BARTH, J. COHEN, L. GASTAL, T. MAUTOR, S. ROUSSEAU. *Comparison of fixed size and variable size packet models in an optical ring network : Algorithms and performances.* in « Photonics in Switching - PS'2003, Versailles, France », pages 89-91, September, 2003.

[9] W. BEN FRAJ, M. ESSAIDI, J. GUSTEDT. *Performance Implications by the Hierarchical Design of Clusters.* in « The 7th world multiconference on Systemics, Cybernetics and Informatics - SCI'2003, Orlando, Florida, USA », July, 2003.

[10] P. BERTHOMÉ, J. COHEN, T. MAUTOR. *Optimisation des ressources utilisées pour une diffusion.* in « 5ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision - ROADEF'2003, Avignon, France », February, 2003.

[11] Y. CANIOU, E. JEANNOT. *New Dynamic Heuristics in the Client-Agent-Server Model.* in « IEEE Heterogeneous Computing Workshop - HCW'03, Nice, France », April, 2003.

[12] J. COHEN, E. JEANNOT, N. PADOY. *Messages Scheduling for Data Redistribution between Clusters.* in « Algorithms, models and tools for parallel computing on heterogeneous network - HeteroPar'03, workshop of SIAM PPAM 2003, Czestochowa, Poland », September, 2003.

[13] J. GUSTEDT. *Randomized Permutations in a Coarse Grained Parallel Environment [extended abstract].* in « Fifteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures - SPAA'03, San Diego, CA, USA », ACM, ACM Press, F. M. AUF DER HEIDE, editor, pages 248-249, June, 2003.

[14] J. GUSTEDT. *Towards Realistic Implementations of External Memory Algorithms using a Coarse Grained Paradigm.* in « International Conference on Computer Science and its Applications - ICCSA'2003, Montréal, Canada », series Lecture Notes in Computer Science, volume 2668, Springer, pages 269-278, February, 2003.

[15] J. GUSTEDT, J. A. TELLE. *A work-optimal coarse-grained PRAM algorithm for Lexicographically First Maximal Independent Set.* in « Italian Conference on Theoretical Computer Science - ICTCS'03, Bertinoro, Italy », series Lecture notes in Computer Science, volume 2841, EATCS, Springer, C. BLUNDO, C. LANEVE, editors, pages 125-136, October, 2003.

Internal Reports

[16] O. BEN FREDJ. *Modélisation de l'Hétérogènéité pour le Calcul Parallèle à Gros Grain.* Stage de DEA, UHP, July, 2003.

[17] J. COHEN, E. JEANNOT, N. PADOY. *Parallel Data Redistribution Over a Backbone.* Rapport de recherche, number 4725, INRIA, February, 2003, http://www.inria.fr/rrrt/rr-4725.html.

[18] L. GASTAL. *Optimisation de la gestion de communications point à point dans un réseau optique.* Stage de DEA, Departement d'Informatique Université de Versailles Saint-Quentin, September, 2003.

[19] F. YIN. *Minimum Receiving Node Minimum Energy Broadcast in All-Wireless Networks.* Stage de DEA, UHP, September, 2003, http://www.loria.fr/publications/2003/A03-R-222/A03-R-222.ps.

Bibliography in notes

[20] T. H. CORMEN, M. T. GOODRICH. *A Bridging Model for Parallel Computation, Communication, and I/O.* in « ACM Computing Surveys », number 4, volume 28A, 1996.

[21] D. CULLER, R. KARP, D. PATTERSON, A. SAHAY, K. SCHAUSER, E. SANTOS, R. SUBRAMONIAN, T. VON EICKEN. *LogP: Towards a Realistic Model of Parallel Computation.* in « Proceeding of 4-th ACM SIGPLAN Symp. on Principles and Practises of Parallel Programming », pages 1-12, 1993.

[22] F. DEHNE, W. DITTRICH, D. HUTCHINSON. *Efficient external memory algorithms by simulating coarsegrained parallel algorithms.* in « ACM Symposium on Parallel Algorithms and Architectures », pages 106-115, 1997.

[23] F. DEHNE, A. FABRI, A. RAU-CHAPLIN. *Scalable parallel computational geometry for coarse grained multicomputers.* in « International Journal on Computational Geometry », number 3, volume 6, 1996, pages 379-400.

[24] L. G. VALIANT. *A bridging model for parallel computation.* in « Communications of the ACM », number 8, volume 33, 1990, pages 103-111.