INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team AlGorille

Algorithms for the Grid

Lorraine

THEME NUM

Activity Report

2004

Table of contents

1. Team

Team Leader
Jens Gustedt [research director, INRIA]

Administrative Assistant
Josiane Reffort [UHP]

Staff Member
Johanne Cohen [research fellow, CNRS, until 01/09/2004]
Emmanuel Jeannot [Maître de Conférences, UHP]
Frédéric Suter [Maître de Conférences, UHP, since 01/12/04]

Post-doctoral fellows
Frédéric Suter [INRIA, 01/09/04 to 30/11/04]

Teaching Assistant
Mohamed Essaïdi [UHP, until 30/09/04]

Ph. D. Student
Yves Caniou [joint regional/INRIA grant until 30/09/04]
Frédéric Wagner [joint regional/INRIA grant]

Student Intern
Marc Thierry [Supélec Metz, second year]
Corinna Brinkmann [Universitä Dortmund]

2. Overall Objectives

Keywords: *Grid computing, algorithms, data redistribution, data distribution, 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 these distributed resources that form the Grid is one of the major challenges of information technology. It needs the implementation of specific 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 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:
Design and master the future network infrastructures and communication services platforms.

Research themes: We have identified two specific research themes:

- Transparent resource management: sequential and parallel task scheduling; migration of computations; data exchange, distribution and redistribution of data.
- 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

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

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

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 has not to be imposed to him/her, but entirely done by a "system", so called middleware. This middleware has to be able to manage all resources in a satisfactory way. Currently, numerous algorithmic problems hinder such an efficient resource management and thus the transparent use of the Grid.

By their nature, distributed applications use different types of resources; the most important being these of computing power and network connections. The management and optimization of those 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 policies to optimize their use and allow 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 on applications and middlewares. We aim at proposing new algorithms (or improve the exiting ones) *for* the resource management in middlewares. Our objective is to provide these algorithms in libraries so that they may be easily integrated. For instance we will propose algorithms to efficiently transfer data (data compression, distribution or redistribution of data) or schedule sequential or parallel tasks.

The problems that we are aiming at solving 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 a quite efficient resolution with results close to optimality.

Consequently it is mandatory 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 at 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

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

Participants: Jens Gustedt, Mohamed Essaïdi, Corinna Brinkmann.

Our approach is based on a "good" separation of the different problem levels that we encounter with Grid problems. Simultaneously this has to ensure a good data locality (a computation will use data that are "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 issues related to spatial and temporal distances as we encounter them in the Grid.

Several parallel models offering 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.*, at 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 [35]. 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 at being realistic with regard to two different aspects: algorithms and architectures. In fact, the coarseness of these models uses the common characteristic of today's parallel settings: the size of the input is orders of magnitude larger than the number of processors that are available. In contrast to the PRAM (Parallel Random Access Machine) model, the coarse grained models are able to integrate the cost of communications 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) [35], LOGP (Latency overhead gap Procs) [32], CGM (Coarse Grained Multicomputer) [34] and *PRO* (Parallel Resource Optimal Model) [7].

The assumptions on the architecture are very similar: p homogeneous processors with local memory 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 on 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 an easy design of algorithms and 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 available resources.

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 has to be constrained by other more "*natural*" parameters coming from the architecture and the problem instance.

A first solution to combine these objectives has been given in [7] with *PRO*.

Starting from this model, we try to develop high level algorithms for the Grid. It will be based upon an abstract view of the architecture and as far as possible be independent of the intermediate levels. It aims at being 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:

- 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, Frédéric Suter, Yves Caniou, Frédéric Wagner, Marc Thierry.

4.1.1. Scheduling on the Grid

Our work deals with algorithms that allocate applications divided into tasks onto remote compute servers in an client-agent-server model. A good scheduling of these tasks is a primal requirement to achieve good performance.

We have investigated the limits of the greedy algorithm MCT (Minimum Completion Time), used in the NetSolve middleware for instance (see Section 5.1). To improve it, we introduced the notion of an "*history*" allowing a better prediction of the execution time of a task on a particular server. On the basis of real experimentations, we shown the interest 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 different heuristics when scheduling an application with precedence 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 present a large gain on the *makespan*, the *sumflow* and on the quality of service over the MCT heuristic. Moreover, we study the accuracy of the HTM, observed from all undertaken experiments. We show that the HTM is able to provide very accurate and useful information, and allows a good environment management.

We also proposed algorithms minimizing the perturbation caused by the allocation of some task to a server. This optimization is done by still enforcing a good performance (response time) for that task. This system-oriented approach has first been tested through simulations. The best among all the heuristics that we studied have been integrated into NetSolve and studied on a broader scale [1]. The thesis of Yves Caniou, [13], defended in December this year, was centered around this subject.

4.1.2. Parallel Task Scheduling

Two kinds of parallelism can be exploited in most scientific applications: data- and task-parallelism. One way to maximize the degree of parallelism of a given application is to combine both kinds of parallelism. This approach is called *mixed data and task parallelism* or *mixed parallelism*. In mixed-parallel applications, several data-parallel computations can be executed concurrently in a task-parallel way. This increases scalability as more parallelism can be exploited when the maximal amount of either data- or task-parallelism has been achieved.

This capability is a key advantage for today's parallel computing platforms. Indeed, to face the increasing computation and memory demands of parallel scientific applications, a recent approach has been to aggregate multiple compute clusters either within or across institutions. Typically, clusters of various sizes are used, and different clusters contain nodes with different capabilities depending on the technology available at the time each cluster was assembled. Therefore, the computing environment is at the same time attractive because of the large computing power, and challenging because it is heterogeneous.

A number of researchers have explored mixed-parallel application *scheduling* in the context of homogeneous platforms. However, heterogeneous platforms have become prevalent and are extremely attractive for deploying applications at unprecedented scales. We propose to build on existing scheduling algorithms for heterogeneous platforms (*i.e.*, specifically designed for task-parallelism) to develop scheduling algorithms for mixed-parallelism on heterogeneous platforms.

For a certain class of scheduling heuristics (list scheduling heuristics) a generic adaptation can be used. While in task scheduling the smallest computational element is a processor, in mixed parallel scheduling the smallest element is a *configuration*. A configuration is defined as a subset of the set of the processors available within, and only within, a cluster and gives information about the size and shape of the virtual grid it represents.

The generic method consists in adapting different functions of an heuristic to handle the allocation of tasks on sets of processors.

4.1.3. Data Redistribution Between Clusters

During computations performed on clusters of machines it occurs that data has to be shifted from one cluster to an other. For instance, these two clusters may differ in the resources they offer (specific hardware, computing power, available software) and each cluster may be more adequate for a certain phase of the computation. Then the data have to be redistributed from the first cluster to the second. 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 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 [5]. Then we have to study lower bounds, approximation algorithms and heuristics. We already obtained some results on heuristics that show a good practical behavior.

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 [21].

4.1.4. 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 switch to a high speed network, since here the time to compress and uncompress the data dominates the transfer time. Then a programmer wanting to be efficient in both cases, would have to provide two different implementations of the network layer of his code, and a user of this program would have to determine which of the variants he/she has to run to be efficient in a particular case.

In [12] we presented 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 overlaps the compression and communication and automatically adapts the effort for compression to the available resources (network and CPU). It also includes another compression algorithm which favors speed against compression ratio. This allows very good performance when dealing with fast networks.

These algorithms are implemented in our library ADOC, "*Adaptive Online Compression*" which has been deposed at the *Agence de Protection des Programmes*. The ADOC library is known to be portable on the Linux, FreeBSD, MAC OS X, Solaris, AIX, IRIX operating systems.

4.2. High Performance Computing

Participants: Jens Gustedt, Mohamed Essaïdi, Corinna Brinkmann.

4.2.1. Models and Algorithms for Coarse Grained Computation

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

To be usable in a Grid context this modeling has first of all to 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 target 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 an 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 are independent, up to a certain degree, 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.

List ranking, tree contraction and graph coloring [8] algorithms already have been designed following the coarse grained setting given by the model *PRO* [7].

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* (LFMIS) problem [10].

We already are 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 terabytes).

4.2.2. External Memory Computation

In the mid-nineties several authors [31][33] 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.

In fact, the stability of our library *SSCRAP*, see Section 5.2, also showed in its extension towards external memory computing [9]. 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 gigabytes 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.

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 [7] and the library *SSCRAP* we already provide tools to better design algorithms (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 project-team GRAAL (previously REMAP), we contribute to the design of the DIET (Distributed Interactive Engineering Toolbox) middleware. The aim of the DIET project is to develop a set of tools to build computational servers.

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 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 project-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 them.

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 simultaneously.

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 sending 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.

This year we worked on several performance issues that arisen when dealing with fast networks and special kind of data. The goal is to improve performance when possible while not degrade performance on fast architecture and already compressed data. More precisely we dealt with the following issues:

- **Very fast network.** We have previously incorporated LZF (Lev-Zimpel-Free) into ADOC in order to speedup compression for fast ethernet networks. For gigabit networks, we test the bandwidth, and disable the compression for such network.
- **Compression level divergence.** If the receiver is very slow compared to the sender, the size of the fifo queue will increase. This will lead to increase the compression ratio. Then the receiver will take more time to decompress the data, which will lead in an increase of the size of its fifo queue, and so on. As we want to respect the read/write system call semantic, the receiver cannot send a message back to the sender. Our solution is to record visible bandwidth for each compression level and disable those that have a lower bandwidth than the lowest compression level.
- **Small messages.** The ADOC mechanism (thread, mutexes, protocol overhead, ...) increases the latency. This becomes critical for small messages. Therefore we have a simpler (no cost) protocol, without compression for small messages.
- **Compressed or random data.** For compressed or random data, compression is not useful as it degrades the performance. When we detect such data, we disable the compression for the next second.

6.2. Scheduling on the Grid

Participants: Yves Caniou, Emmanuel Jeannot.

The Historical Trace Manager (HTM) is a task duration predictor module embedded in the agent of a Problem Solving Environment relying on the client-agent-server model [18][17][23][24]. We improved the HTM and NetSolve – the Problem Soving Environment we use for our tests – in order to synchronize the HTM to the reality. We introduced two new scheduling heuristics relying on the HTM information: Advanced HMCT and Minimum Length. We studied the scheduling of several scenarios, including the simultaneous submissions of DAGs and independent tasks, on a real heterogeneous platform. The excellent behavior of the HTM validates its estimations of the duration of each task concurrently running in the system. It can consequently predict the contention tasks may have on each other if scheduled and executed concurrently on the same computing resource. Heuristics performance shows the relevancy of the HTM information through the experiments: their ability of behaving with a constant quality between two executions of the same experiment as well as the quality of their respective scheduling choices to optimize several criteria at the same time. We also show that heuristics which minimize the contention give generally the best results regardless the criterion. We finally compare the behavior of the heuristics previously tested in previous work to the one observed here with more precise information on the global system state due to the synchronization mechanisms. Surprisingly, in the time-shared model, it does not necessarily improve the job repartition among the servers, performance can consequently decrease and the utilization of the fastest servers can become critical.

6.3. Parallel Task Scheduling

Participant: Frédéric Suter.

This year, we proposed a generic methodology for the conversion of any heterogeneous list-scheduling algorithm for task-parallel application into an algorithm for the mixed-parallel case [4]. We have presented a case study for the popular HEFT scheduling heuristic, which we have extended to obtain the M-HEFT (Mixed-parallel HEFT) heuristic. To derive this mixed-parallel version, the priority and objective functions of HEFT have been adapted the new compute and communication units. Indeed, we now address set of processors and collective communications whereas the HEFT heuristic has been designed for sequential compute units and point-to-point communications.

We resort to simulation for evaluating our approach as it allows us to perform a statistically significant number of experiments and makes is possible to explore a wide range of platform configurations. We use the SimGrid toolkit as the foundation of our simulator. SimGrid provides base abstractions for the discrete-event simulation of parallel applications in distributed environments and was specifically designed for the evaluation of scheduling algorithms. These simulation results showed that M-HEFT achieves good performance over its competitors in the vast majority of the scenarios.

6.4. Emulation of Heterogeneity

Participants: Marc Thierry, Emmanuel Jeannot.

The French national project GridExplorer, see Section 7.2, has created a cluster (about 700 CPU's) that is dedicated to experiments that emulate real grids. This platform is mostly homogeneous: the computing nodes are formed with identical processors with the same amount of memory and the same network interface. However, simulating a grid implies to make the components of GridExplorer heterogeneous. Therefore it is interesting to degrade the performances of the different components to make such simulations possible.

We focused on degrading CPU frequency, network interface latency and bandwidth as well as address space availability. Each such degradation is independent of the other. We developed a software called "*wrekavoc*" that takes a script describing the desired heterogeneity in terms of each of these factors.

We based the degradations on the following tools:

CPU frequency the kernel module *CPUfreq* or a program highly consuming CPU resources.

Memory	the security module of the Linux kernel called *PAM*
Network	the Traffic Control tool “*TC*”.

The configuration is organized by so-called islets, *i.e.*, unions of IP address intervals. All nodes in an islet are defined to have the same CPU and memory characteristics. On top on these local characteristics, we define specific network characteristic between each pair of islets.

The network, CPU and memory characteristics are defined using either a Gaussian distribution (in terms of a mean and a variance) or a uniform distribution within a given interval.

Preliminary tests show that the performance degradation is easy to calibrate and is reproducible.

6.5. Redistribution of Data

Participants: Emmanuel Jeannot, Frédéric Suter, Frédéric Wagner.

With the emergence of grid systems aggregating power of geographically distant sites, the problem of finding an optimal way to execute large parallel data transfers over high speed networks becomes important in order to achieve high performance. In our previous work, we studied the problem of redistributing data located on one cluster to another one over a high performance network called backbone. We attempt not to request more bandwidth than available on any link when scheduling all messages [21][30].

We validated our model by comparing the redistribution time on a real platform and the simulated one [29]. In fact, since everything is done at the transport level, the model perfectly describes our algorithm as well as the brute-force approach (*i.e.*, when no scheduling is performed).

Next, we extended our work for the Δ-port model when more than one communication can take place at each node at the same time. This extend the problem to fully heterogeneous platforms where each node of each cluster can communicate at different speed. We have provided an algorithm for scheduling the messages, which gives a solution at most twice as long as the optimal one. Simulation results show that it is giving almost optimal schedules on redistribution patterns with a high number of communications, and good results in the general case.

Finally, we have extended our previous model by allowing local communications to take place during the redistribution. This allows for example to split large communications among several nodes of the cluster to use in parallel the bandwidth of several network interfaces. This problem is more complicated than the original scheduling problem, because we now need to introduce the notion of routes, as each message can now take a different path until its final destination. In our first approach we assumed that the redistribution is under steady state, *i.e.*, we have to issue an infinite number of redistributions which all have the same redistribution pattern. We proposed an approximation algorithm for the steady-state redistribution problem. This algorithm is executed in three phases: finding optimal routes using linear programming, scheduling all messages (non optimal phase) using graph theory and finally computing initialization steps (we need to issue a finite number of redistributions before reaching steady state).

One of the objectives of the ARC RedGrid was to develop a redistribution library usable by middleware. The software resulting from our algorithms has been integrated into the PaCO++ middleware developed by the PARIS project-team.

6.6. Large scale experiments

Participants: 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. The thesis of Mohamed Essaïdi [14] that has been defended in February this year, was centered around this subject.

We also investigated extensions of the communication layer towards heterogeneous architectures and tested several directions in using research environments from other groups. None of these attempts has been really satisfactory: either the libraries were not sufficiently stable or the code that was produced was not very efficient. During the year 2004, we started looking into another approach that would be only based on standard components of nowadays computing environments that are well mastered and optimized, namely POSIX shared memory segments (`shm_open`) and MPI v. 1.2.

A lot of the code of *SSCRAP* has been rewritten this year to allow for the implementation of the communication layer as described above. As a second direction we also stream-lined the programming interface a lot, in particular to allow an integration of PARCEL-6 and *SSCRAP*. PARCEL-6 is a parallel cellular language designed for complex neural networks (cortical systems) and some physical system simulations (based on local equations) that is develloped by Stéphane Vialle at Supélec, Metz campus. The integration will allow to validate *SSCRAP* on a wide range of fine grained applications and problems. For PARCEL-6 the avantage of clarifying the mapping from the fine-grained formulation (cells) to a coarse-grained real live execution and in addition in achieving portability to distributed environments.

6.7. Models and algorithms for coarse grained computation

Participants: Corinna Brinkmann, Mohamed Essaïdi, Jens Gustedt.

We continued the design of algorithms in the coarse grained setting as given by the model *PRO* [7]. In particular the internship of Corinna Brinkmann aimed for the design and implementation of simple *independent set* heuristics on graphs with an emphasis on graphs that have a low average vertex degree.

The on-going research and discussion on *PRO* and BSP-like computation in general as well as our implementation of *SSCRAP* clearly showed that there is a need for tools that are simple to use and that enforce efficiency at the same time. In [27], we present a programming paradigm and interface (called "*data handover*") that aims to handle data between parallel or distributed processes and that mixes aspects of message passing and shared memory. It is designed to overcome the potential problems in terms of efficiency of both:

- memory blow up and forced copies for message passing and
- data consistency and latency problems for shared memory.

Our approach attempts to be simple and easy to understand. It contents itself with just a handful of functions to cover the main aspects of coarse grained interoperation upon data.

6.8. Overlapping Computations and Communications with I/O

Participant: Frédéric Suter.

Several numerical computation algorithms exhibit dependences that lead to a wavefront in the computation. Depending on the data distribution chosen, pipelining communication and computation can be the only way to avoid a sequential execution of the parallel code. The computation grain has to be wisely chosen to obtain at the same time a maximum parallelism and a small communication overhead.

On the other hand, when the size of data exceeds the memory capacity of the target platform, data have to be stored on disk. The concept of Out-of-Core computation aims at minimizing the impact of the I/O needed to compute on such data. It has been applied successfully on several linear algebra applications.

In [25], we apply Out-of-Core techniques to wavefront algorithms. The originality of our approach is to overlap computation, communication, and I/O. An original strategy is proposed using several memory blocks accessed in a cyclic manner. The resulting pipeline algorithm achieves a saturation of the disk resource which is the bottleneck in Out-of-Core algorithms.

7. Other Grants and Activities

7.1. Bilateral international relations and European initiatives

We take part in the NoE "*CoreGrid*" lead by Thierry Priol from INRIA Rennes. More precisely we are part of the work package 6 on scheduling. Emmanuel Jeannot is the leader for CNRS of task 6.5: evaluation and benchmarking.

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, Knoxville.

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

We collaborate with Vandy Berten and Joel Goossens of the Université Libre de Bruxelles on scheduling problems under stochastic models.

The collaboration with Jack Dongarra of the University of Tennessee, Knoxville 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.

7.2. National initiatives

7.2.1. 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 in 2001 by a project called ARGE in the national grid initiative. ARGE had first been guided by André Schaff, and was recently handed over to Jens Gustedt and Stéphane Vialle (Supélec, Metz Campus).

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 of LRI (Université Paris-Sud), OpPALL of Prism (Université de Saint-Quentin), Opal of LAMI (Université d'Évry), without these contacts being formalized up to now.

7.2.2. ACI initiatives of the French Research Ministry

We are partners in several projects of different ACI initiatives:

- 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 théorique* of Université Henri Poincaré Nancy 1. This application is to distribute the computation of hyper-surface of potential energy of some molecules. This ACI ends in December 2004.
- GRID2 (national animation of the Grid community). This ACI ends in December 2004.
- ARGE (see above). This ACI ends in December 2004.
- Grid Explorer. We participate with a joint proposition together with Stéphane Vialle from Supélec, Metz Campus, which concerns testing the integration of *SSCRAP* and Parcel-6 as described in Section 6.6. We also work on designing a set of emulation tools for transforming an homogeneous platform into an heterogeneous one, see Section 6.4.

[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*

- In the recent (2004) initiative ACI AGIR we participate in the definition and design of a set of services for medical image processing on the grid. More precisely we are in charge of transfer with compression task and the evaluation of grid middleware.

7.2.3. INRIA New Investigation Grant

The goal of the INRIA-ARC RedGrid is to design algorithms and services for the problem of data redistribution between distant clusters. It involves the PARIS, GRAAL, and SCALAPPLIX INRIA project-teams.

AlGorille's investment in this action was to develop the software resulting from our algorithms and to integrate it into the PaCO++ middleware developed by the PARIS project-team.

8. Dissemination

8.1.1. 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 until July 2004 he was appointed member of the commission for scientific prospective, and within INPL until September 2004 he was nominated substitute member of the hiring committee in computer science.

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.

8.1.2. Scientific Expertise

In 2004, Jens Gustedt served as an external expert for the evaluation of scientific projects in regional initiatives for information science and technology in a neighboring European country.

8.1.3. Teaching activities

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

Frédéric Suter is teaching in the *Théorie des graphes* module of the IUP GMI at Henri Poincaré University.

8.1.4. Editorial activities

Since October 2001, Jens Gustedt is Editor-in-Chief of the journal *Discrete Mathematics and Theoretical Computer Science (DMTCS)*. DMTCS is an journal that is published electronically by an independent association under French law. Based on a contract with INRIA, its main web-server is located at the LORIA. This year DMTCS has known a substantial growth and has acquired a good visibility within the concerned domains of Computer Science and Mathematics.

Emmanuel Jeannot was member of the program committee of HCW'04, and inside LORIA he participates in the board of "*lettre du* LORIA".

In 2004, members of the team served as referees for the following journals and conferences:

> IEEE Transactions on Parallel and Distributed Systems, Journal of Parallel Distributed Computing, Computing and Informatics, Soft Computing, Discrete Applied Mathematics, Theory of Computing Systems, PMS'04, ESA'04, STACS'05

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

9. Bibliography

Major publications by the team in recent years

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

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

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

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

[5] 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.

[6] F. DESPREZ, F. SUTER. *Impact of Mixed-Parallelism on Parallel Implementations of Strassen and Winograd Matrix Multiplication Algorithms*, in "Concurrency and Computation:Practice and Experience", vol. 16, n^o 8, July 2004, p. 771–797.

[7] 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, June 2002, p. 106-113.

[8] A. H. GEBREMEDHIN, I. GUÉRIN LASSOUS, J. GUSTEDT, J. A. TELLE. *Graph Coloring on a Coarse Grained Multicomputers*, in "Discrete Applied Mathematics", vol. 131, n^o 1, September 2003, p. 179-198.

[9] 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", Lecture Notes in Computer Science, vol. 2668, Springer, February 2003, p. 269-278.

[10] 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", C. BLUNDO, C. LANEVE (editors)., Lecture notes in Computer Science, vol. 2841, Springer, EATCS, October 2003, p. 125-136.

[11] I. GUÉRIN LASSOUS, J. GUSTEDT. *Portable List Ranking : an Experimental Study*, in "ACM Journal of Experimental Algorithmics", vol. 7, n^o 7, July 2002.

[12] 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.

Doctoral dissertations and Habilitation theses

[13] Y. CANIOU. *Ordonnancement sur une plate-forme de métacomputing*, Thèse d'université, Université Henri Poincaré, December 2004.

[14] M. ESSAÏDI. *Echange de données pour le parallélisme à gros grain*, Thèse d'université, Université Henri Poincaré, Feb 2004.

Articles in referred journals and book chapters

[15] D. BARTH, P. BERTHOMÉ, J. COHEN. *The Eulerian stretch of a digraph and the ending guarantee of a convergence routing*, in "Journal of Interconnection Networks (JOIN)", vol. 5, n^o 2, Jan 2004, p. 93-109.

[16] M. COSNARD, E. JEANNOT, T. YANG. *Compact DAG Representation and its Symbolic Scheduling*, in "Journal of Parallel and Distributed Computing (JPDC)", vol. 64, n^o 8, aug 2004, p. 921-935.

Publications in Conferences and Workshops

[17] Y. CANIOU, E. JEANNOT. *Efficient Scheduling Heuristics for GridRPC Systems*, in "QOS and Dynamic System Workshop of ICPADS'2004 - 10th International Conference on Parallel and Distributed Systems, New-Port Beach, California, USA", IEEE, Jul 2004, p. 621–630.

[18] Y. CANIOU, E. JEANNOT. *Experimental Study of Multi-Criteria Scheduling Heuristics for GridRPC Systems*, in "Euro-Par 2004, Pisa, Italy", IEEE-ACM-IFIP, Aug 2004.

[19] F. DESPREZ, E. JEANNOT. *Improving the GridRPC Model with Data Persistence and Redistribution*, in "3rd International Symposium on Parallel and Distributed Computing - ISPDC'2004, Cork, Ireland", held with Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks - HeteroPar'04, IEEE, Jul 2004, p. 193–200.

[20] E. JEANNOT, F. WAGNER. *Message Scheduling for Data Redistribution through High Performance Networks*, in "DistRibUtIon de Données à grande Echelle - DRUIDE'2004, Le Croisic, France", May 2004, http://www.loria.fr/publications/2004/A04-R-197/A04-R-197.ps.

[21] E. JEANNOT, F. WAGNER. *Two fast and efficient message scheduling algorithms for data redistribution through a backbone*, in "18th International Parallel and Distributed Processing Symposium - IPDPS'04, Santa Fe, New Mexico", IEEE, Apr 2004.

Internal Reports

[22] D. BARTH, J. COHEN, M. LE COZ, F. QUESSETTE. *A First Approach of Grouping Problem in Stochastic Automata Network*, Rapport technique, Jun 2004.

[23] Y. CANIOU, E. JEANNOT. *Improvements and Study of the Accuracy of the Tasks Duration Predictor, New Heuristics*, Rapport de recherche, n° RR-5206, INRIA, May 2004, http://www.inria.fr/rrrt/rr-5206.html.

[24] Y. CANIOU, E. JEANNOT. *Study of the behaviour of heuristics relying on the Historical Trace*, Rapport de recherche, n° RR-5168, INRIA, Apr 2004, http://www.inria.fr/rrrt/rr-5168.html.

[25] E. CARON, F. DESPREZ, F. SUTER. *Overlapping Computations and Communications with I/O in Wavefront Algorithms*, Rapport de recherche, n° RR-5410, Dec 2004, http://www.inria.fr/rrrt/rr-5410.html.

[26] M. ESSAÏDI, I. GUÉRIN LASSOUS, J. GUSTEDT. *SSCRAP : Soft Synchronized Computing in Rounds for Adequate Parallelization*, Rapport de recherche, n° RR-5184, INRIA, May 2004, http://www.inria.fr/rrrt/rr-5184.html.

[27] J. GUSTEDT. *Data Handover : Reconciling Message Passing and Shared Memory*, Rapport de recherche, n° RR-5383, INRIA, Nov 2004, http://www.inria.fr/rrrt/rr-5383.html.

[28] J. GUSTEDT. *External Memory Algorithms using a Coarse Grained Paradigm*, Rapport de recherche, n° RR-5142, INRIA, Mar 2004, http://www.inria.fr/rrrt/rr-5142.html.

[29] E. JEANNOT, F. WAGNER. *Modelizing, Predicting and Optimizing Redistribution between Clusters on Low Latency Networks*, Rapport de recherche, n° RR-5361, Nov 2004, http://www.inria.fr/rrrt/rr-5361.html.

[30] F. WAGNER, E. JEANNOT. *Message scheduling for data redistribution through high performance networks*, Rapport de recherche, n° RR-5077, INRIA, Apr 2004, http://www.inria.fr/rrrt/rr-5077.html.

Bibliography in notes

[31] T. H. CORMEN, M. T. GOODRICH. *A Bridging Model for Parallel Computation, Communication, and I/O*, in "ACM Computing Surveys", vol. 28A, n° 4, 1996.

[32] 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", 1993, p. 1-12.

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

[34] F. DEHNE, A. FABRI, A. RAU-CHAPLIN. *Scalable parallel computational geometry for coarse grained multicomputers*, in "International Journal on Computational Geometry", vol. 6, n° 3, 1996, p. 379-400.

[35] L. G. VALIANT. *A bridging model for parallel computation*, in "Communications of the ACM", vol. 33, n° 8, 1990, p. 103-111.