CEPAGE
Chercher et Essaimer dans les Plates-formes À Grande Echelle
Searching and dissemination of information on large-scale platforms
Project-Team proposal – CR Bordeaux - Sud-Ouest

Olivier Beaumont Nicolas Bonichon Philippe Duchon
Lionel Eyraud-Dubois Cyril Gavoille Nicolas Hanusse David Ilcinkas
Ralf Klasing

January 22, 2008

1 Composition

- **Project-Team Leader**
  - Olivier Beaumont, assistant professor, ENSEIRB
- **Staff members**
  - Nicolas Bonichon, assistant professor, Univ. Bordeaux 1
  - Lionel Eyraud, CR2 INRIA, from 09/07
  - Cyril Gavoille, Professor, Univ. Bordeaux 1
  - Nicolas Hanusse, CR1 CNRS, LaBRI
  - David Ilcinkas, CR2 CNRS, LaBRI
  - Ralf Klasing, CR1 CNRS, LaBRI
- **External Collaborator**
  - Philippe Duchon, assistant professor (delegation CNRS) ENSEIRB
- **Postdoctoral Students**
  - Christopher Thraves Caro, (should arrive 15/02), ANR Alpage
- **PhD Students**
  - Youssou Dieng, PhD Student, Univ. Bordeaux 1, 3rd year
  - Hejer Rejeb, PhD Student, Univ. Bordeaux 1, 1st year
  - Hubert Larchevêque, PhD Student, BDI Région/CNRS, 1st year
  - Radu Tofan, PhD Student, INRIA CORDIS, 1st year

Contents

2 Overall objective

The development of interconnection networks has led to the emergence of new types of computing platforms. These platforms are characterized by heterogeneity of both processing and communication resources, geographical dispersion, and instability in terms of the number and performance of participating resources. These characteristics restrict the nature of the applications that can perform well on these platforms. Due to middleware and application deployment times, applications must be long-running and involve large amounts of data; also, only loosely-coupled applications may currently be executed on unstable platforms.

The new algorithmic challenges associated with these platforms have been approached from two different directions. On the one hand, the parallel algorithms community has largely concentrated on the problems associated with heterogeneity and large amounts of data. On the other hand, the distributed systems community has focused on scalability and fault-tolerance issues. The success of file sharing applications demonstrates the capacity of the resulting algorithms to manage huge volumes of data and users on large unstable platforms. Algorithms developed within this context are completely distributed and based on peer-to-peer (P2P for short) communication.

The goal of our project is to establish a link between these two directions, by gathering researchers from the distributed algorithms and data structures, parallel and randomized algorithms communities. More precisely, the objective of our project is to extend the application field that can be executed on large scale distributed platforms. Indeed, whereas protocols designed for P2P file exchange are actually distributed, computationally intensive applications executed on large scale platforms (BOINC [1], WCG [2] or XTremWeb [3]) mostly rely on a client-server model, where no direct communication between peers is allowed. This characteristic strongly influences the set of applications that can be executed, as underlined in the call for project proposals of WCG:

> Projects must meet three basic technological requirements, to ensure benefits from grid computing:
> 1. Projects should have a need for millions of CPU hours of computation to proceed. However, humanitarian projects with smaller CPU hour requirements are able to apply.
> 2. The computer software algorithms required to accomplish the computations should be such that they can be subdivided into many smaller independent computations.
> 3. If very large amounts of data are required, there should also be a way to partition the data into sufficiently small units corresponding to the computations.

Given these constraints, applications using large data sets should be such that they can be arbitrarily split into small pieces of data (such as Seti@home [4]) and computationally intensive applications should be such that they can be arbitrarily split into small pieces of work (such as Folding@home [5] or Monte Carlo simulations).
These constraints are both related to security and algorithmic issues. Security is of course an important issue, since executing non-certified code on non-certified data on a large scale, open, distributed platform is clearly unacceptable. Nevertheless, we believe that external techniques,

[1] http://boinc.berkeley.edu/
[2] http://www.worldcommunitygrid.org/
[3] http://www.lri.fr/ fedak/XtremWeb/
[4] http://setiathome.berkeley.edu/
[5] http://folding.stanford.edu/

such as Sandboxing, certification of data and code through hashcode mechanisms, should be used to solve these problems. Therefore, the focus of our project is on algorithmic issues and in what follows, we assume a cooperative environment of well-intentioned users, and we assume that security and cooperation can be enforced by external mechanisms. Our goal is to demonstrate that gains in performances and extension of the application field justify these extra costs but that, just as operating systems do for multi-users environments, security and cooperation issues should not affect the design of efficient algorithms nor reduce the application field.

We will concentrate on the design of new services for computationaly intensive applications, consisting of mostly independent tasks sharing data, with application to distributed storage, molecular dynamics and distributed continuous integration, that will be described in more details in Section 7.

Most of the research (including ours) currently carried out on these topics relies on a centralized knowledge of the whole (topology and performances) execution platform, whereas recent evolutions in computer networks technology yield a tremendous change in the scale of these networks. The solutions designed for scheduling and managing compact data structures must be adapted to these systems, characterized by a high dynamism of their entities (participants can join and leave at will), a potential instability of the large scale networks (on which concurrent applications are running), and the increasing probability of failure.

P2P systems have achieved stability and fault-tolerance, as witnessed by their wide and intensive usage, by changing the view of the networks: all communication occurs on a logical network (fixed even though resources change over time), thus abstracting the actual performance of the underlying physical network. Nevertheless, disconnecting physical and logical networks leads to low performance and a waste of resources. Moreover, due to their original use (file exchange), those systems are well suited to exact search using Distributed Hash Tables (DHT's) and are based on fixed regular virtual topologies (Hypercubes, De Bruijn graphs...). In the context of the applications we consider, more complex queries will be required (finding the set of edges used for content distribution, finding a set of replicas covering the whole database) and, in order to reach efficiency, unstructured virtual topologies must be considered.

The overall goal of the project is to provide a set of basic tools and services that will be used to program more complex applications than those actually deployed on large scale platforms. More specifically, our approach consists in selecting a few applications and extracting a few services that may be of general use. Our project is based on the following observations:

- Solutions offered by scheduling and parallel computing communities are much too centralized to be implemented in practice on large scale platforms
- Services offered by P2P and distributed systems communities are much too poor (mostly restricted to exact search and content distribution) to be used as building blocks for scientific applications

Therefore, scientific applications on large scale platforms consists in applications that can be arbitrarily split into small tasks that operate on small amount of data. These applications are all based on a client-server paradigm where the server is responsible for storing all data produced by the algorithm and schedule all tasks. In order to extend the applicative field, we consider the following applications, described into more details in the Section 6 and 7 of the proposal

- In order to remove the constraint on the size of the tasks, we propose a distributed mechanism for building dynamically small heterogeneous clusters that will be responsible for larger tasks that cannot fit in the memory of a single node. This is the subject of the PhD thesis started last September by Hubert Larchevêque. This is done in collaboration

with Juan Elezgaray (IECB) for a parallel molecular dynamics code where each task involves a large amount of data.

- In order to remove the constraint on the client server protocol, we propose a mechanism to store intermediate data produced by the application in a distributed way and to use DHT-like mechanisms to retrieve data without any central server. We have successfully applied this technique to the same molecular dynamics code and we have proved that our implementation scales well up to 250 nodes on GRID'5000 platform (whereas previous parallel solutions based on domain decomposition did not scale up to 16 processors).

- We also propose the design of a new service called "dating service" to match randomly requests and offers on large scale platforms. This service has been applied to organize communications (where offers and requests correspond to incoming and outgoing bandwidths). We are currently working with Yahoo! to balance the load between disks that handle requests and where the popularity of a file (and therefore the load associated to its storage) varies over time. Another interest for CEPAGE is to benefit from the possibility of performing experiments on the large scale platform provided by Yahoo!.

- At last, we also consider distributed build applications. It corresponds to complex queries where a task looks for a distant node that holds a given set of files required for the execution of the task (or the node that may be able to get these resources quickly) where to be processed. This corresponds to the case where moving a task is much cheaper than moving associated files).

We plan to prove both formally (time and space complexities) and through experiments on large scale platforms (see Section 4.4) that these services can be efficiently implemented, using GRID'5000 platform. Turning them into a programming language or a library may be considered as the objective of the project, but it seems overly ambitious and too far from the group expertise at this step to claim that we will be able to do it in near future. Nevertheless, we strongly believe that such general use services are needed, since actual middlewares and languages for large scale platforms are either too close from those used for parallel programming (Globus, ...) inducing huge costs to hide platform heterogenity and dynamism, or too poor to program complex applications (DHTs, ...) inducing unacceptable constraints on the applicative field.

In this context, the main scientific challenges of our project are

- **Models:**
  - At a low level, to understand the underlying physical topology and to obtain both realistic and instanciable models. This requires expertise in graph theory (all the members of the project) and platform modelling (Olivier Beaumont, Nicolas Bonichon, Lionel Eyraud and Ralf Klasing). The obtained results will be used to focus the algorithms designed in Sections 5 and 6.
  - at a higher level, to derive models of the dynamism of targeted platforms, both in terms of participating resources and resource performances (Olivier Beaumont, Philippe Duchon). Our goal is to derive suitable tools to analyze and prove algorithm performances in dynamic conditions rather than to propose stochastic modeling of evolutions (Section 3).

- **Overlays and distributed algorithms:**
  - to understand how to augment the logical topology in order to achieve the good properties of P2P systems. This requires knowledge in P2P systems and small-world networks (Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Nicolas Hanusse, Cyril Gavoille). The obtained results will be used for developing the algorithms designed in Sections 5 and 6.
  - to build overlays dedicated to specific applications and services that achieve good performances (Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Lionel Eyraud, Ralf Klasing). The set of applications and services we target will be described in more details in Section 6 and 7.
  - to understand how to dynamically adapt scheduling algorithms (in particular collective communication schemes) to changes in network performance and topology, using randomized algorithms (Olivier Beaumont, Nicolas Bonichon, Nicolas Hanusse, Philippe Duchon, Ralf Klasing) (Section 6).
- **Compact and distributed data structures:**
  - to understand how to dynamically adapt compact data structures to changes in network performance and topology (Nicolas Hanusse, Cyril Gavoille) (Section 5)
  - to design sophisticated labeling schemes in order to answer complex predicates using local labels only (Nicolas Hanusse, Cyril Gavoille) (Section 5)

We will detail in Section 7 how the various expertises in the team will be employed for the considered applications. The main focus of the project is therefore on distributed algorithms and distributed systems. Although we are not experts in the design of distributed systems, we believe that the scientific domain is now mature enough so that we can use already developped softwares as building blocks in order to deal, at low level, with platform dynamism. All the algorithms, services and softwares we develop are based on well known implementation of distributed data structures such as DHTs, skip lists, skip graphs, epidemic algorithms... On the other hand, we consider that this domain may greatly benefit from the expertises of team members in platform modeling, compact data structures, parallel and randomized algorithms.

We therefore tackle several problems related to the major challenges that INRIA identified in its strategic plan (2008-2012). In particular, the project deals with the modeling of large scale dynamic applications (”Modeling, simulation of optimization of complex dynamic systems” axis) and with the development of new services for self-organization on large scale platforms and new grid applications (”Information, Communication and ubiquitous systems” axis).

3 Goal and context

3.1 General context

The recent evolutions in computer networks technology, as well as their diversification, yield a tremendous change in the use of these networks: applications and systems can now be designed at a much larger scale than before. This scaling evolution is dealing with the amount of data, the number of computers, the number of users, and the geographical diversity of these users. This race towards *large scale* computing has two major implications. First, new opportunities are offered to the applications, in particular as far as scientific computing, data bases, and file sharing are concerned. Second, a large number of parallel or distributed algorithms developed for average size systems cannot be run on large scale systems without a significant degradation of their performances. In fact, one must probably relax the constraints that the system should satisfy in order to run at a larger scale. In particular the coherence protocols designed for the distributed applications are too demanding in terms of both message and time complexity, and must therefore be adapted for running at a larger scale. Moreover, most distributed systems deployed nowadays are characterized by a high dynamism of their entities (participants can join and leave at will), a potential instability of the large scale networks (on which concurrent applications are running), and an increasing individual probability of failure. Therefore, as the size of the system increases, it becomes necessary that it adapts automatically to the changes of its components, requiring self-organization of the system to deal with the arrival and departure of participants, data, or resources.

As a consequence, it becomes crucial to be able to understand and model the behavior of large scale systems, to efficiently exploit these infrastructures, in particular w.r.t. designing dedicated algorithms handling a large amount of users and/or data.

3.2 Limitations of parallel processing solutions

In the case of parallel computation solutions, some strategies have been developed in order to cope with the intrinsic difficulty induced by resource heterogeneity. It has been proved that changing the metric (from makespan minimization to throughput maximization) simplifies most scheduling problems, both for collective communications and parallel processing. This restricts the use of target platforms to simple and regular applications, but due to the time needed to develop and deploy applications on large scale distributed platforms, the risk of failures, the intrinsic dynamism of resources, it is unrealistic to consider tightly coupled applications involving many tight synchronizations. Nevertheless, (1) it is unclear how the current models can be adapted to large scale systems, and (2) the current methodology requires the use of (at least partially) centralized subroutines that cannot be run on large scale systems. In particular, these subroutines assume the ability to gather all the information regarding the network at a single node (topology, resource performance, etc.). This assumption is unrealistic in a general purpose large size platform, in which the nodes are unstable, and whose resource characteristics can vary abruptly over time. Moreover, the proposed solutions for small to average size, stable, and dedicated environments do not satisfy the minimal requirements for self-organization and fault-tolerance, two properties that are unavoidable in a large scale context. Therefore, there is a strong need to design efficient and decentralized algorithms. This requires in particular to define new metrics adapted to large scale dynamic platforms in order to analyze the performance of the proposed algorithms.

3.3 Limitations of P2P strategies

As already noted, P2P file sharing applications have been successfully deployed on large scale dynamic platforms. Nevertheless, since our goal is the design of efficient algorithms in terms of actual performance and resource consumption, we need to concentrate on specific P2P environments. Indeed, P2P protocols are mostly designed for file sharing applications, and are not optimized for scientific applications, nor are they adapted to sophisticated database applications. This is mainly due to the primitive goal of designing file sharing applications, where anonymity is crucial, exact queries only are used, and all large file communications are made at the IP level.

Unfortunately, the context strongly differs for the applications we consider in our project, and some of the constraints appear to be in contradiction with performance and resource consumption optimization. For instance, in these systems, due to anonymity, the number of neighboring nodes in the overlay network (i.e. the number of IP addresses known to each peer) is kept relatively low, much lower than what the memory constraints on the nodes actually impose. Such a constraint induces longer routes between peers, and is therefore in contradiction with performance. In those systems, with the main exception of the LAND overlay, the overlay network (induced by the connections of each peer) is kept as far as possible separate from the underlying physical network. This property is essential in order to cope with malicious attacks, i.e. to ensure that even if a geographic site is attacked and disconnected from the rest of the network, the overall network will remain connected. Again, since actual communications between peers occur between peers connected in the overlay network, communications between two close nodes (in the physical network) may well involve many wide area messages, and therefore such a constraint is in contradiction with performance optimization. Fortunately, in the case of file sharing applications, only queries are transmitted using the overlay network, and the communication of large files is made at IP level. On the other hand, in the case of more complex communication schemes, such as broadcast or multicast, the communication of large files is done using the overlay network, due to the lack of support, at IP level, for those complex operations. In this case, in order to achieve good results, it is crucial that virtual and physical topologies be as close as possible.

3.4 Targeted platforms

Our aim is to target large scale platforms. From parallel processing, we keep the idea that resource heterogeneity dramatically complicates scheduling problems, what imposes to restrict ourselves to simple applications. The dynamism of both the topology and the performance reinforces this constraint. We will also adopt the throughput maximization objective, though it needs to be adapted to more dynamic platforms and resources.

From previous work on P2P systems, we keep the idea that there is no centralized large server and that all participating nodes play a symmetric role (according to their performance in terms of memory, processing power, incoming and outgoing bandwidths, etc.), which imposes the design of self-adapting protocols, where any kind of central control should be avoided as much as possible.

Since dynamism constitutes the main difficulty in the design of algorithms on large scale dynamic platforms, we will consider several layers in dynamism:

- **Stable:** In order to establish the complexity induced by dynamism, we will first consider fully heterogeneous (in terms of both processing and communication resources) but fully stable platforms (where both topology and performance are constant over time).
- **Semi-stable:** In order to establish the complexity induced by fault-tolerance, we will

then consider fully heterogeneous platforms where resource performance varies over time, but topology is fixed.

- **Unstable:** At last, we will target systems facing the arrival and departure of participants, data or resources.

4 Models, theoretical and practical validation

4.1 Modeling platform dynamics

Modeling the platform dynamics in a satisfying manner, in order to design and analyze efficient algorithms, is a major challenge. In a semi-stable platform, the performance of individual nodes (be they computing or communication resources) will fluctuate; in a fully dynamic platform, which is our ultimate target, the set of available nodes will also change over time, and algorithms must take these changes into account if they are to be efficient.

There are basically two ways one can model such evolution: one can use a *stochastic process*, or some kind of *adversary model*.

In a stochastic model, the platform evolution is governed by some specific probability distribution. One obvious advantage of such a model is that it can be simulated and, in many well-studied cases, analyzed in detail. The two main disadvantages are that it can be hard to determine how much of the resulting algorithm performance comes from the specifics of the evolution process, and that estimating how realistic a given model is – none of the current project participants are metrology experts.

In an adversary model, it is assumed that these unpredictable changes are under the control of an adversary whose goal is to interfere with the algorithms efficiency. Major assumptions on the system's behavior can be included in the form of restrictions on what this adversary can do (like maintaining such or such level of connectivity). Such models are typically more general than stochastic models, in that many stochastic models can be seen as a probabilistic specialization of a nondeterministic model (at least for bounded time intervals, and up to negligible probabilities of adopting "forbidden" behaviors).

Since we aim at proving guaranteed performance for our algorithms, we want to concentrate on suitably restricted adversary models. The main challenge in this direction is thus to describe sets of restricted behaviors that both capture realistic situations and make it possible to prove such guarantees.

4.2 Models for platform topology and parameter estimation

On the other hand, in order to establish complexity and approximation results, we also need to rely on a precise theoretical model of the targeted platforms.

- At a lower level, several models have been proposed to describe interference between several simultaneous communications. In the 1-port model, a node cannot simultaneously send to (or/and receive from) more than one node. Most of the steady state scheduling results have been obtained using this model. On the other hand, some authors propose to model incoming and outgoing communication from a node using fictitious incoming and outgoing links, whose bandwidths are fixed. The main advantage of this model, although it might be slightly less accurate, is that it does not require strong synchronization and that many scheduling problems can be expressed as multi-commodity flow problems, for which decentralized efficient algorithms are known. Another important issue is to model the bandwidth actually allocated to each communication when several communications compete for a WAN link.

- At a higher level, proving good approximation ratios on general graphs may be too difficult, and it has been observed that actual platforms often exhibit a simple structure. For instance, many real life networks satisfy small-world properties, and it has been proved, for instance, that greedy routing protocols on small world networks achieve good performance. It is therefore of interest to prove that logical (given by the interactions between hosts)

and physical platforms (given by the network links) exhibit some structure in order to derive efficient algorithms.

4.3 Theoretical validation

In order to analyze the performance of the proposed algorithms, we first need to define a metric adapted to the targeted platform. In particular, since resource performance and topology may change over time, the metric should also be defined from the optimal performance of the platform at any time step. For instance, if throughput maximization is concerned, the objective is to provide for the proposed algorithm an approximation ratio with respect to

$$\int_{\text{SimulationTime}} \text{OptThroughput}(t)$$

or at least

$$\min_{\text{SimulationTime}} \text{OptThroughput}(t).$$

For instance, Awerbuch and Leighton [AL93,AL94] developed a very nice distributed algorithm for computing multi-flows. The algorithm proposed in [AL94] consists in associating queues and potential to each commodity at each node for all incoming or outgoing edges. These regular queues store the flow that did not reach its destination yet. Using a very simple and very natural framework, flow goes from high potential areas (the sources) to low potential areas (the sinks). This algorithm is fully decentralized since nodes make their decisions depending on their state (the size of their queues), the state of their neighbors (the size of their queues), and the capacity of neighboring links.

The remarkable property about this algorithm is that if, at any time step, the network is able to ship $(1 + \epsilon)d_i$ flow units for each capacity at each time step, then the algorithm will ship at least d_i units of flow at steady state. The proof of this property is based on the overall potential of all the queues in the network, which remains bounded over time.

It is worth noting that this algorithm is quasi-optimal for the metrics we defined above, since the overall throughput can be made arbitrarily close to

$$\min_{\text{SimulationTime}} \text{OptThroughput}(t).$$

In this context, the approximation result is given under an adversary model, where the adversary can change both the topology and the performances of communication resources between any two steps, provided that the network is able to ship $(1 + \epsilon)d_i$.

4.4 General framework for validation

4.4.1 Low level modeling of communications

In the context of large scale dynamic platforms, it is unrealistic to determine precisely the actual topology and the contention of the underlying network at application level. Indeed, existing tools such as Alnem [LMQ03] are very much based on quasi-exhaustive determination of

[AL93] Baruch Awerbuch and Frank Thomson Leighton. A simple local-control approximation algorithm for multicommodity flow. In *IEEE Symposium on Foundations of Computer Science*, pages 459–468, 1993.

[AL94] Baruch Awerbuch and Tom Leighton. Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks. In *IEEE Symposium on Foundations of Computer Science*, pages 487–496, 1994.

[LMQ03] A. Legrand, F. Mazoit, and M. Quinson. An application-level network mapper. Research Report RR-2003-09, LIP, ENS Lyon, France, feb 2003.

interferences, and it takes several days to determine the actual topology of a platform made up of a few tens of nodes. Given the dynamism of the platforms we target, we need to rely on less sophisticated models, whose parameters can be evaluated at runtime.
Therefore, we propose to model each node by an incoming and an outgoing bandwidth and to neglect interference that appears at the heart of the network (Internet), in order to concentrate on local constraints. We are currently implementing a script, based on Iperf[6] to determine the achieved bit-rates for one-to-one, one-to-many and many-to-one transfers, given the number of TCP connections, and the maximal size of the TCP windows. The next step will be to build a communication protocol that enforces a prescribed sharing of the network resources. In particular, if in the optimal solution, a node P_0 must send data at rate x_i^{OUT} to node P_i and receive data at rate y_j^{IN} from node P_j, the goal is to achieve the prescribed bitrates, provided that all capacity constraints are satisfied at each node. Our aim is to implement using Java RMI a protocol able to both evaluate the parameters of our model (incoming and outgoing bandwidths) and to ensure a prescribed sharing of communication resources.

4.4.2 Simulation

Once low level modeling has been obtained, it is crucial to be able to test the proposed algorithms. To do this, we will first rely on simulation rather than direct experimentation. Indeed, in order to be able to compare heuristics, it is necessary to execute those heuristics on the same platform. In particular, all changes in the topology or in the resource performance should occur at the same time during the execution of the different heuristics. In order to be able to replicate the same scenario several times, we need to rely on simulations. Moreover, the metric we have tentatively defined for providing approximation results in the case of dynamic platforms requires to compute the optimal solution at each time step, which can be done off-line if all traces for the different resources are stored. Using simulation rather than experiments can be justified if the simulator itself has been proved valid. Moreover, the modeling of communications, processing and their interactions may be much more complex in the simulator than in the model used to provide a theoretical approximation ratio, such as in SimGrid. In particular, sophisticated TCP models for bandwidth sharing have been implemented in SimGRID.

At a higher level, the derivation of realistic models for large scale platforms is out of the scope of our project, as explained in Section 8. Therefore, in order to obtain traces and models, we will collaborate with MESCAL, GANG and ASAP projects. We already worked on these topics with the members of GANG in the ACI Pair-A-Pair (ACI Pair-A-Pair finished in 2006, but we have proposed a follow-up, with the members of GANG and Cepage projects to ANR Blanche program). On the other hand, we also need to rely on an efficient simulator in order to test our algorithms. We have not yet chosen the discrete event simulator we will use for simulations. One attractive possibility would be to adapt SimGRID, developed in the Mescal project, to large scale dynamic environments. Indeed, a parallel version of SimGrid, based on activations is currently under development. This version will be able to deal with platforms containing more than 10^5 resources. SimGrid has been developed by Henri Casanova (U.C. San Diego) and Arnaud Legrand during his PhD (under the co-supervision of O. Beaumont).

4.5 Practical validation and scaling

Finally, we propose several applications that will be described in detail in Section 7. These applications cover a large set of fields (molecular dynamics, distributed storage, continuous integration, distributed databases...). All these applications will be developed and tested with

[6] (`http://dast.nlanr.net/Projects/Iperf/`)

an academic or industrial partner. In all these collaborations, our goal is to prove that the services that we propose in Section 6.2 can be integrated as steering tools in already developed software. Our goal is to assert the practical interest of the services we develop and then to integrate and to distribute them as a library for large scale computing.

In order to test our algorithms, we propose to implement these services using Java RMI. The main advantages of Java RMI in our context are the ease of use and the portability. Multithreading is also a crucial feature in order to schedule concurrent communications and it does not interfere with ad-hoc routing protocols developed in the project.
A prototype has already been developed in the project as a steering tool for molecular dynamic simulations (see Section 7.1). In order to test their scalability, ll the applications we proposed will first be validated on the GRID 5000 platform or the partner's platform. GRID'5000 is a very valuable and unique platform in the context of the deployment of large scale applications, since it is actually possible to test applications and to validate the usefulness of proposed services at large scale (thousands of nodes). To validate our solutions at a larger scale (10^5 nodes), we will rely on the tools developped within the INRIA ANDT project Aladdin to emulate larger platforms using GRID'5000.

5 Efficient queries and compact data structures

The optimization schemes for content distribution processes or for handling standard queries require a good knowledge of the physical topology or performance (latencies, throughput, ...) of the network. Assuming that some rough estimate of the physical topology is given, former theoretical results described in Section 5.1 show how to pre-process the network so that local computations are performed efficiently. Due to the dynamism of large distributed platforms, some requirements on the coding of local data structures and the updating mechanism are needed. This last process is done using the maintenance of light virtual networks, so-called *overlay networks* (see Section 5.2). In our approach, we focus on:

- *Compression.*

  The emergence of huge distributed networks does not allow the topology of the network to be totally known to each node without any compression scheme. There are at least two reasons for this:

  – In order to guarantee that local computations are done efficiently, that is avoiding external memory requests, it may be of interest that the coding of the underlying topology can be stored within *fast memory* space.

  – The dynamism of the network implies many basic message communications to update the knowledge of each node. The smaller the message size is, the better the performance.

  The compression of any topology description should not lead to an extra cost for standard requests: distance between nodes, adjacency tests, ... Roughly speaking, a decoding process should not be necessary.

- *Routing tables.*

  Routing queries and broadcasting information on large scale platforms are tasks involving many basic message communications. The maximum performance objective imposes that basic messages are routed along paths of cost as low as possible. On the other hand, local routing decisions must be fast and the algorithms and data structures involved must support a certain amount of dynamism in the platform.

- *Local computations.*

  Although the size of the data structures is less constrained in comparison with P2P systems (due to security reasons), however, even in our collaborative framework, it is unrealistic that each node manages a complete view of the platform with the full resource characteristic. Thus, a node has to manage data structures concerning only a fraction of the whole system. In fact, a partial view of the network will be sufficient for many tasks: for instance, in order to compute the distance between two nodes (distance labeling).

- *Overlay and small world networks.*

  The processes we consider can be highly dynamic. The preprocessing usually assumed takes polynomial time. Hence, when a new process arrives, it must be dealt with in an *on-line* fashion, i.e., we do not want to totally re-compute, and the (partial) re-computation has to be simple.

  In order to meet these requirements, *overlay networks* are normally implemented. These are light virtual networks, i.e., they are sparse and a local change of the physical network

will only lead to a small change of the corresponding virtual network. As a result, small address books are sufficient at each node.

A specific class of overlay networks are *small-world* networks. These are efficient overlay networks for (greedy) routing tasks assuming that distance requests can be performed easily.

Of course, the main difficulty is to adapt the maintenance of local data structures to the dynamism of the network.

5.1 Compression and short data structures

5.1.1 Routing with short tables

There are several techniques to manage sub-linear size routing tables (in the number of nodes of the platform) while guaranteeing almost shortest paths (cf. [Gav01] for a survey of routing techniques).

Some techniques provide routes of length at most $1 + \epsilon$ times the length of the shortest one while maintaining a poly-logarithmic number of entries per routing table [AGGM05,CGMZ05,Sli05]. However, these techniques are not universal in the sense that they apply only on some class of underlying topologies. Universal schemes exist. Typically they achieve $O(\sqrt{n})$-entry local routing tables for a stretch factor of 3 in the worst case [AGM+04,TZ01]. Some experiments have shown that such methods, although universal, work very well in practice, in average, on realistic scale-free or existing topologies [KFY04].

While the fundamental question is to determine the best stretch-space trade-off for universal schemes, the challenge for platform routing would be to design specific schemes supporting reasonable dynamic changes in the topology or in the metric, at least for a limited class of relevant topologies. In this direction [BLTT97] have constructed (in polynomial time) network topologies for which nodes can be labeled once such that whatever the link weights vary in time, shortest path routing tables with compacity k can be designed, i.e., for each routing table the set of destinations using the same first outgoing edge can be grouped in at most k ranges of consecutive labels.

[Gav01] Cyril Gavoille. Routing in distributed networks: Overview and open problems. *ACM SIGACT News - Distributed Computing Column*, 32(1):36–52, March 2001.

[AGGM05] Ittai Abraham, Cyril Gavoille, Andrew V. Goldberg, and Dahlia Malkhi. Routing in networks with low doubling dimension. Technical Report MSR-TR-2005-175, Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052 - http://www.research.microsoft.com, December 2005.

[CGMZ05] T.-H. Hubert Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. In 16^{th} *Symposium on Discrete Algorithms (SODA)*, pages 762–771. ACM-SIAM, January 2005.

[Sli05] Aleksandrs Slivkins. Distance estimation and object location via rings of neighbors. In 24^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 41–50. ACM Press, July 2005. Appears earlier as Cornell CIS technical report TR2005-1977.

[AGM+04] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In 16^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 20–24. ACM Press, July 2004.

[TZ01] Mikkel Thorup and Uri Zwick. Compact routing schemes. In 13^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 1–10. ACM Press, July 2001.

[KFY04] Dmitri Krioukov, Kevin Fall, and Xiaowei Yang. Compact routing on internet-like graphs. *IEEE INFOCOM*, 2004. To appear.

[BLTT97] Hans Leo Bodlaender, Jan van Leeuwen, Richard B. Tan, and Dimitrios M. Thilikos. On interval routing schemes and treewidth. *Information and Computation*, 139:92–109, November 1997.

One other aspect of the problem would be to model a realistic typical platform topology. Natural parameters (or characteristic) for this are its low dimensionality: low Euclidean or near Euclidean networks, low growing dimension, or more generally, low doubling dimension.

5.1.2 Succinct representation of underlying topologies

In order to optimize applications the platform topology itself must be discovered, and thus represented in memory with some data structures. The size of the representation is an important parameter, for instance, in order to optimize the throughput during the exploration phase of the platform.

Classical data structures for representing a graph (matrix or list) can be significantly improved when the targeted graph falls in some specific classes or obeys to some properties: the graph has bounded genus (embeddable on surface of fixed genus), bounded tree-width (or c-decomposable), or embeddabble into a bounded page number [GH05,GH99]. Typically, planar topologies with n nodes (thus embeddable on the plane with no edge crossings) can by efficiently coded in linear time with at most $5n + o(n)$ bits supporting adjacency queries in constant time. This improves the classical adjacency list within a non negligible $\log n$ factor on the size (the size is about $6n \log n$ bits for edge list), and also on the query time [BGH+04,BGH03a,BGH03b].

5.1.3 Local data structures and other queries

The basic routing scheme and the overlay networks must also allow us to route other queries than routing driven by applications. Typically, divide-and-conquer parallel algorithms require to compute many nearest common ancestor (NCA) queries in some tree decomposition. In a large scale platform, if the current tree structure is fully or partially distributed, then the physical location of the NCA in the platform must be optimized. More precisely, the NCA computation must be performed from distributed pieces of information, and then addressed via the routing overlay network (cf. [AGKR04] for distributed NCA algorithms).

Recently, a theory of localized data structures has been developed (initialized by [Pel00];

[GH05] Cyril Gavoille and Nicolas Hanusse. On compact encoding of pagenumber k graphs. *Discrete Mathematics & Theoretical Computer Science*, 2005. To appear.

[GH99] Cyril Gavoille and Nicolas Hanusse. Compact routing tables for graphs of bounded genus. In Jiří Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, 26^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1644 of Lecture Notes in Computer Science, pages 351–360. Springer, July 1999.

[BGH+04] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, Dominique Poulalhon, and Gilles Schaeffer. Planar graphs, via well-orderly maps and trees. In 30^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 3353 of Lecture Notes in Computer Science. Springer, June 2004. 270-284.

[BGH03a] Nicolas Bonichon, Cyril Gavoille, and Nicolas Hanusse. Canonical decomposition of outerplanar maps and application to enumeration, coding and generation. In 29^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 2880 of Lecture Notes in Computer Science, pages 81–92. Springer-Verlag, June 2003.

[BGH03b] Nicolas Bonichon, Cyril Gavoille, and Nicolas Hanusse. An information-theoretic upper bound of planar graphs using triangulation. In 20^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of Lecture Notes in Computer Science, pages 499–510. Springer, February 2003.

[AGKR04] Stephen Alstrup, Cyril Gavoille, Haim Kaplan, and Theis Rauhe. Nearest common ancestors: A survey and a new algorithm for a distributed environment. *Theory of Computing Systems*, 37:441–456, 2004.

[Pel00] David Peleg. Informative labeling schemes for graphs. In 25^{th} *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 1893 of Lecture Notes in Computer Science, pages 579–588. Springer, August 2000.

see [GP03] for a survey). One associates with each node a label such that some given function (or predicate) of the node can be extracted from two or more labels. Theses labels are usually joined to the addresses or inserted into a global database index.

In relation with the project, queries involving the flow computation between any sink-target pair of a capacitated network is of great interest [KKKP04]. Dynamic labeling schemes are also available for tree models [Kor05,KP03], and need further work for their adaptation to more general topologies.

Finally, localized data structures have applications to platforms implementing large database XML file types. Roughly speaking pieces of a large XML file are distributed along some platform, and some queries (typically some SELECT ... FROM extractions) involve many tree ancestor queries [AAK+05], the XML file structure being a tree. In this framework, distributed label-based data structures avoid the storing of a huge classical index database.

5.2 Overlay and small world networks

An overlay network is a virtual network whose nodes correspond either to processors or to resources of the network. Virtual links may depend on the application; for instance, different overlay networks can be designed for routing and broadcasting.

These overlay networks should support insertion and deletion of users/resources, and thus they inherently have a high dynamism.

We should distinguish *structured* and *unstructured* overlay networks:

- In the first case, one aims at designing a network in which queries can be answered efficiently: greedy routing should work well (without backtracking), the spreading of a piece of information should take a very short time and few messages. The natural topology of these networks are graph of small diameter and bounded degree (De Bruijn graph for instance). However, dynamic maintenance of a precise structure is difficult and any perturbation of the topology gives no guarantee for the desired tasks.

- In the case of unstructured networks, there is no strict topology control. For the information retrieval task, the only attempt to bound the total number of messages consists of optimizing a flooding by taking into account statistics stored at each peer: number of requests that found an item traversing a given link, ...

In both approaches, the physical topology is not involved. To our knowledge, there exists only one attempt in this direction. The work of Abraham and Malhki [AMD04] deals with the design of routing tables for stable platforms.

We are interested in designing overlay topologies that take into account the physical topology.

[GP03] Cyril Gavoille and David Peleg. Compact and localized distributed data structures. *Journal of Distributed Computing*, 16:111–120, May 2003. PODC 20-Year Special Issue.

[KKKP04] Michal Katz, Nir A. Katz, Amos Korman, and David Peleg. Labeling schemes for flow and connectivity. *SIAM Journal on Computing*, 34(1):23–40, 2004.

[Kor05] Amos Korman. General compact labeling schemes for dynamic trees. In 19^{th} *International Symposium on Distributed Computing (DISC)*, volume 3724 of Lecture Notes in Computer Science, pages 457–471. Springer, September 2005.

[KP03] Amos Korman and David Peleg. Labeling schemes for weighted dynamic tree. In 30^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2719 of Lecture Notes in Computer Science, pages 369–383. Springer, July 2003.

[AAK+05] Serge Abiteboul, Stephen Alstrup, Haim Kaplan, Tova Milo, and Theis Rauhe. Compact labeling schemes for ancestor queries. *SIAM Journal on Computing*, 2005.

[AMD04] Ittai Abraham, Dahlia Malkhi, and Oren Dobzinski. Land: stretch (1 + epsilon) locality-aware networks for dhts. In *Symposium of Discrete Algorithms (SODA)*, pages 550–559, 2004.

Another work is promising. If we relax the condition of designing an overlay network with a precise topology but with some topological properties, we might construct very efficient overlay networks. Two directions can be considered: *random graphs* and *small-world* networks.

Random graphs are promising for broadcast and have been proposed for the update of replicated databases in order to minimize the total number of messages and the time complexity [DGH+88,KSSV00]. The underlying topology is the complete graph but the communication graph (pairs of nodes that effectively interact) is much more sparse. At each pulse of its local clock, each node tries to send or receive any new piece of information. The advantage of this approach is fault-tolerance. However, this epidemic spreading leads to a waste of messages since any node can receive many times the same update. We are interested in fixing this drawback and we think that it should be possible.

For several queries, recent solutions use small-world networks. This approach is inspired from experiments in social sciences [Mil67]. It suggests that adding a few (non uniform) random and uncoordinated virtual long links to every node leads to shrink drastically the diameter of the network. Moreover, paths with a small number of hops can be found [Kle00,FGP04,DHLS05].

Solutions based on network augmentation (i.e. by adding virtual links to a base network) have proved to be very promising for large scale networks. This technique is referred to as turning a network into a small-world network, also called the *small-worldization* process. Indeed, it allows to transform many arbitrary networks into networks in which search operations can be performed in a greedy fashion and very quickly (typically in time poly-logarithmic in the size of the network). This property implies that any information can be easily accessed.

Our goal is to study more precisely the algorithmic performance of these new small-world networks (w.r.t. time, memory, pertinence, fault-tolerance, auto-stabilization, ...) and to propose new networks of this kind, i.e. to construct the augmentation of the base network as well as to conceive the corresponding navigation algorithm. Like classical algorithms for routing and navigation (that are essentially based on greedy algorithms), the proposed solutions have to take into account that no entity has a global knowledge of the network. A first result in this direction is promising. In [DHLS06], we proposed an economic distributed algorithm to turn a bounded growth network into a small-world. Moreover, the practical challenge will be to adapt such constructions to dynamic networks, at least under the models that are identified as relevant.

Can the *small-worldization* process be supported in dynamic platforms? Up to now, the literature on small-world networks only deals with the routing task. We are convinced that

[DGH+88] Alan J. Demers, Daniel H. Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard E. Sturgis, Daniel C. Swinehart, and Douglas B. Terry. Epidemic algorithms for replicated database maintenance. *Operating Systems Review*, 22(1):8–32, 1988.

[KSSV00] Richard M. Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vöcking. Randomized rumor spreading. In *FOCS*, pages 565–574, 2000.

[Mil67] S. Milgram. The small world problem. *Psychology Today*, 61(1), 1967.

[Kle00] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*, pages 163–170, 2000.

[FGP04] P. Fraigniaud, C. Gavoille, and C. Paul. Eclecticism shrinks even small worlds. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 169–178, 2004.

[DHLS05] P. Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Could any graph be turned into a small world ? In Pierre Fraigniaud, editor, *International Symposium on Distributed Computing (DISC)*, volume 3724 of *Lecture Notes in Computer Science*, pages 511–513. Springer Verlag, 2005.

[DHLS06] Philippe Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Towards small world emergence. In Uzi Vishkin, editor, *SPAA2006 - 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 225–232, PO box 11405, NY - 10286-6626, July 2006. ACM SIGACT - ACM SIGARCH, ACM Pess.

small-world topologies are also relevant for other tasks: quick broadcast, search in presence of faulty nodes, ... In general, we think that maintaining a small-world topology can be much more realistic than maintaining a rigidly structured overlay network and much more efficient for several tasks in unstructured overlay networks.

6 New services for scheduling and processing on large scale platforms.

As mentioned in Section 2, solutions provided by the parallel algorithm community are dedicated to stable platforms whose resource performances can be gathered at a single node that is responsible for computing the optimal solution. On the other hand, P2P systems are fully distributed but the set of available queries in these systems is much too poor for computationally intensive applications. Therefore, actual solutions for large scale distributed platforms such as BOINC [7], WCG [8] or XTremWeb [9] mostly rely on a client-server model, where no direct communication between peers is allowed. The objective of our project is to extend the application field that can be executed on large scale distributed platforms.

6.1 Requests and Task scheduling on large scale semi-stable distributed platforms

Even if the application field for large scale platforms is currently too poor, targeted platforms are clearly not suited to tightly coupled codes and we need to concentrate on simple scheduling problems in the context of large scale distributed unstable platforms. Indeed, most of the scheduling problems are already NP-Complete with bad approximation ratios in the case of static homogeneous platforms when communication costs are not taken into account.

Recently, many algorithms have been derived, under several communication models, for master slave tasking [BBC+04,HP04] and Divisible Load Scheduling (DLS) [BGMR96,BMR05,AGR03].

In this case, we aim at executing a large bag of independent, same-size tasks. First we assume that there is a single master, that initially holds all the (data needed for all) tasks. The problem is to determine an architecture for the execution. Which processors should the master enroll in the computation? How many tasks should be sent to each participating processor? In turn, each processor involved in the execution must decide which fraction of the tasks must be computed locally, and which fraction should be sent to which neighbor (these neighbors must be determined too).

Parallelizing the computation by spreading the execution across many processors may well be limited by the induced communication volume. Rather than aiming at makespan minimization, a more relevant objective is the optimization of the throughput in steady-state mode. There are three main reasons for focusing on the steady-state operation. First is *simplicity*, as the

[7] http://boinc.berkeley.edu/
[8] http://www.worldcommunitygrid.org/
[9] http://www.lri.fr/ fedak/XtremWeb/

[BBC+04] C. Banino, O. Beaumont, L. Carter, J. Ferrante, A. Legrand, and Y. Robert. Scheduling strategies for master-slave tasking on heterogeneous processor platforms. *IEEE Trans. Parallel Distributed Systems*, 15(4):319–330, 2004.

[HP04] B. Hong and V.K. Prasanna. Distributed adaptive task allocation in heterogeneous computing environments to maximize throughput. In *International Parallel and Distributed Processing Symposium IPDPS'2004*. IEEE Computer Society Press, 2004.

[BGMR96] V. Bharadwaj, D. Ghose, V. Mani, and T.G. Robertazzi. *Scheduling Divisible Loads in Parallel and Distributed Systems*. IEEE Computer Society Press, 1996.

[BMR05] Olivier Beaumont, Loris Marchal, and Yves Robert. Scheduling divisible loads with return messages on heterogeneous master-worker platforms. In *International Conference on High Performance Computing HiPC'2005*, LNCS. Springer Verlag, 2005.

[AGR03] M. Adler, Y. Gong, and A. L. Rosenberg. Optimal sharing of bags of tasks in heterogeneous clusters. In *15th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA'03)*, pages 1–10. ACM Press, 2003.

steady-state scheduling is in fact a relaxation of the makespan minimization problem in which the initialization and clean-up phases are ignored. One only needs to determine, for each participating resource, which fraction of time is spent computing for which application, and which fraction of time is spent communicating with which neighbor; the actual schedule then arises naturally from these quantities.

Even if some problems remain open in the case of static platforms (especially the difficult case of return messages in the case of DLS), the problem of throughput maximization in the presence of heterogeneous processing and communication resources is now well understood. On the other hand, all proposed algorithms are based on the centralized computation of the optimal schedule and therefore, are not well suited to large scale dynamic platforms. Unstable platforms would require a totally different approach, where task replication would have to play a major role. How to choose the replication factor, and how to efficiently keep track of successfully executed copies? Another important criterion to consider are the *average* response time (or delay in the system), and *maximal* response time. In fact, designing multi-criteria algorithms capable of achieving a wide range of throughput/response time trade-offs would be very valuable.

6.2 New services for processing on large scale distributed platforms

6.2.1 Heterogeneous dating service

In many distributed applications on large distributed systems, nodes may offer some local resources and request some remote resources. For instance, in a distributed storage environment, nodes may offer some space to store remote files and request some space to duplicate remotely some of their files. In the context of broadcasting, offer may be seen as the outgoing bandwidth and request as the incoming bandwidth. In the context of load balancing, overloaded nodes may request to get rid of some tasks whereas underloaded nodes may offer to process them. In this context, we propose a distributed algorithm, called *dating service* which is meant to organize communication in a fully heterogeneous network, so that communication capabilities of nodes are not exceeded. The abstract purpose of our scheme is to randomly join demands and supplies of some resource of many nodes into couples. In a round it produces a matching between demands and supplies which is of linear size (compared to optimal one) and is chosen uniformly at random from all matchings of this size.

We believe that this basic operation can be of great interest in many practical applications and could be used as a building block for writing efficient software on large distributed unstable platforms. We plan to demonstrate its practical efficiency for content distribution, management of large databases and distributed storage applications described in Section 7.

6.2.2 Building Heterogeneous Clusters

As already noted in Section 2 with the example of WCG call for proposal, the application field of Grid computing is limited by several constraints. In particular, the target application should be easy to divide into small independent pieces of work, so that each individual piece can be executed on a single node. This strongly limits the application field since in many cases, data may be too large to fit into the memory of a single node.

In this context, we propose a distributed algorithm to dynamically build clusters of nodes able to process large tasks. These sets of nodes should satisfy constraints on the overall available memory, on its processing power together with constraints on the maximal latency between nodes and the minimal bandwidth between two participating nodes.

We believe that such a distributed service would enable to consider a much larger application field. We plan to demonstrate first its practical efficiency for the application of molecular

dynamics (based on NAMD) described in more detail in Section 7.

6.2.3 Complex queries for non-trivial parallel algorithms

In many applications on large scale distributed platforms, the application data files are distributed among the platform and the volatility in the availability of resources forbids to rely on a centralized system to locate data.

In this context, complex queries, such as finding a node holding a given set of files, or holding a file whose index is close to a given value, or a set of (close) nodes covering a given set of files, should be treated in a distributed manner. Queries built for P2P systems are much too poor to handle such requests.

We plan to demonstrate the usefulness and the efficiency of such requests on the molecular dynamics application and on the continuous integration application described in Section 7. Again, we strongly believe that these operations can be considered as useful building blocks for most large scale distributed applications that cannot be executed in a client-server model, and that providing a library with such mechanisms would be of great interest.

7 Software

7.1 Molecular Dynamics Simulations

Another interesting scheduling problem is the case of applications sharing (large) files stored in replicated distributed databases. We deal here with a particular instance of the scheduling problem mentioned in Section 6.1. This instance involves applications that require the manipulation of large files, which are initially distributed across the platform.

It may well be the case that some files are replicated. In the target application, all tasks depend upon the whole set of files. The target platform is composed of many distant nodes, with different computing capabilities, and which are linked through an overlay network (to be built). To each node is associated a (local) data repository. Initially, the files are stored in one or several of these repositories. We assume that a file may be duplicated, and thus simultaneously stored on several data repositories, thereby potentially speeding up the next request to access them. There may be restrictions on the possibility of duplicating the files (typically, each repository is not large enough to hold a copy of all the files). The techniques developed in Section 5.1.3 will be used to dynamically maintain efficient data structures for handling files.

Our aim is to design a prototype for both maintaining data structures and distributing files and tasks over the network.

This framework occurs for instance in the case of Monte-Carlo applications where the parameters of new simulations depend on the average behavior of the simulations previously performed. The general principle is the following: several simulations (independent tasks) are launched simultaneously with different initial parameters, and then the average behavior of these simulations is computed. Then other simulations are performed with new parameters computed from the average behavior. These parameters are tuned to ensure a much faster convergence of the method. Running such an application on a semi-stable platform is a particular instance of the scheduling problem mentioned in Section 6.1.

We will focus on a particular algorithm picked from Molecular Dynamics: calculation of Potential of Mean Force (PMF) using the technique of Adaptive Bias Force (ABF). This work is done via a collaboration with Juan Elezgaray, IECB, Bordeaux. Here is a quick presentation of this context. Estimating the time needed for a molecule to go through a cellular membrane is an important issue in biology and medicine. Typically, the diffusion time is far too long to be computed with atomistic molecular simulations (the average time to be simulated is of order of 1s and the integration step cannot be chosen larger than 10^{-15}, due to the nature of physical interactions). Classical parallel approaches, based on domain decomposition methods, lead to very poor results due to the number of barriers. Another method to estimate this time is by calculating the PMF of the system, which is in this context the average force the molecule is subject to at a given position within or around the membrane. Recently, Darve et al. [DP01] presented a new method, called ABF, to compute the PMF. The idea is to run a small number of simulations to estimate the PMF, and then add to the system a force that cancels the estimated PMF. With this new force, new simulations are performed starting from different configurations (distributed over the computing platform) of the system computed during the previous simulations and so on. Iterating this process, the algorithm converges quite quickly to a good estimation of the PMF with a uniform sampling along the axis of diffusion. This application has been implemented and integrated to the famous molecular dynamics software NAMD [HC04].

[DP01] E. Darve and A. Pohorille. Calculating free energies using average force. *Journal of Chemical Physics*, 115:9169–9183, 2001.

[HC04] J. Hénin and C. Chipot. Overcoming free energy barriers using unconstrained molecular dynamics simulations. *Journal of Chemical Physics*, 121:2904–2914, 2004.

Our aim is to propose a distributed implementation of ABF method using NAMD. It is worth noting that NAMD is designed to run on high-end parallel platforms or clusters, but not to run efficiently on instable and distributed platforms. A prototype of a steering tool for NAMD has been developed in the project, that may be used to validate our approach and that has been tested on GRID'5000 up to 200 processors.

The different problems to be solved in order to design this application are the following:

- Since we need to start a simulation from a valid configuration (which can represent several Mbytes) with a particular position of the molecule in the membrane, and these configurations are spread among participating nodes, we need to be able to find and to download such configuration. Therefore, the first task is to find an overlay such that those requests can be handled efficiently. This requires expertise in overlay networks, compact data structures and graph theory. Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Nicolas Hanusse, Cyril Gavoille and Ralf Klasing will work on this part.

- In our context, each participating node may offer some space for storing some configurations, some bandwidth and some computing power to run simulations. The question arising here is how to distribute the simulations to nodes such that computing power of all nodes are fully used. Since nodes may join and leave the network at any time, redistributions of configurations and tasks between nodes will also be necessary (but all tasks only contribute to update the PMF, so that some tasks may fail without changing the overall result). The techniques designed for content distribution will be used to spread and redistribute the set of configurations over the set of participating nodes. This requires expertise in task scheduling and distributed storage. Olivier Beaumont, Nicolas Bonichon and Philippe Duchon will work on this part.

7.2 Continuous Integration

Continuous Integration is a development method in which developers commit their work in a version control system (such as CVS or Subversion) very frequently (typically several times per day) and the project is automatically rebuilt. One of the advantages of this technique is that merge problems are detected and corrected early.

The build process not only generates the binaries, it also runs automated tests, generates documentation, checks the code coverage of tests and analyzes code style...

The whole process can take several hours for large projects. Therefore, the efficiency of this development method relies on the speed of the feedback. There is a real need to speed up the build process, and thus to distribute it. This is one of the goal continuous integration server xooctory [10] initiated by Xavier Hanin (Jayasoft [11]).

In order to obtain an efficient distribution of the build, the build process can be decomposed into nearly independent sub processes, executed on different nodes. Nevertheless, to be completed, a sub process must be run on a node that holds the appropriate version of the tools (compiler, code auditing software, ...), the appropriate version of the libraries, and the appropriate version of source code. Of course, if the target node does not have all these items, it can download them from another node, but these communications may be more expensive than the execution of the sub processes.

This raises several challenging problems:

- Build a distributed data structure that can efficiently provide

[10] http://xooctory.xoocode.org/
[11] http://www.jayasoft.fr/index.php

  - one of the nodes that stores a certain set S of files.
  - one of the nodes that stores a maximum subset S' of a set S of files.
  - one of the nodes that can obtain quickly a certain set S of files (i.e. a node that can download efficiently the files of S that it does not already holds).

- Design distribution strategies of the build that take advantage of the processing and communication capabilities of the nodes.

We are collaborating with Xavier Hanin and Jayasoft in order to solve distribution problems in the context of distributed continuous integration. Our goal is to incorporate some of the services developed in Cepage to obtain a large scale distributed version of the continuous integration server xooctory.

7.3 Data Cubes

Data cube queries represent an important class of On-Line Analytical Processing (OLAP) queries in decision support systems. They consist in a pre-computation of the different group-bys of a database (aggregation for every combination of GROUP BY attributes) that is a very consuming task. For instance, databases of some megabytes may lead to the construction of a datacube requiring terabytes of memory [LNCL02] and parallel computation has been proposed but for a static and well-identified platform [DEHRC02]. This application is typically an interesting example for which the distributed computation and storage can be useful in an heterogeneous and dynamic setting. We just started a collaboration with Noel Novelli (Assistant Professor of Marseille University) who is a specialist of datacube computation. Our goal is to rely on the set of services defined in Section 6.2 to compute and maintain huge datacubes.

7.4 Requests in Large Databases

We are working with Cyril Banino (Yahoo Research, Trondheim) on data management for large scale distributed databases. In the context of the Yahoo platform, data is stored among several thousands of nodes, so that centralized solutions are no longer valid, and the system must rely on self-organization to balance the load. In this context, the platform is relatively stable (although nodes frequently experience failures and nodes are frequently added), but the set of stored data is highly dynamic, since data are frequently added and their popularity changes very quickly over time.

We work on data-management issues and the adaptation of CRUSH [WBMM] and Sorrento [TGZ+04] protocols used to localize data. An important issue is the design of mechanisms to distribute data over the set of participating nodes. The objective is both to balance the load in terms of storage among the different storage devices and to balance the load in terms of processed requests among the different processing units. Given the dynamism of the requests

[LNCL02] Marc Laporte, Noel Novelli, Rosine Cicchetti, and Lotfi Lakhal. Computing full and iceberg datacubes using partitions. In *ISMIS '02: Proceedings of the 13th International Symposium on Foundations of Intelligent Systems*, pages 244–254, London, UK, 2002. Springer-Verlag.

[DEHRC02] Frank K. H. A. Dehne, Todd Eavis, Susanne E. Hambrusch, and Andrew Rau-Chaplin. Parallelizing the data cube. *Distributed and Parallel Databases*, 11(2):181–201, 2002.

[WBMM] S.A. Weil, S.A. Brandt, E.L. Miller, and C. Maltzahn. CRUSH: Controlled, scalable, decentralized placement of replicated data. *Proceedings of the 2006 ACM,IEEE Conference on Supercomputing (SC06)*.

[TGZ+04] H. Tang, A. Gulbeden, J. Zhou, L. Chu, and T. Yang. Sorrento: a self-organizing storage cluster for parallel data-intensive applications. *International Conf for High Performance Computing Networking and Storage, Pittsburgh, PA, USA*, 2004.

and the files to be stored, the scale of the system and the risk of failure due to the large number of storage and processing units, we believe that the techniques developed in the context of P2P systems may also be used in the context of large distributed databases. To balance both loads (storage and requests), we plan to rely on the services described in Section 6.2.

8 Positioning

8.1 Within INRIA

Many INRIA teams are currently working on large scale distributed networks. In this section, we list the projects closest to our proposal and we underline the main differences between Cepage and these projects. The large number of projects working in this area is certainly due to the convergence between the parallel processing and distributed systems and distributed algorithms communities.

On the one hand, parallel platforms evolve towards the aggregation of many distributed resources (either computational or desktop grids) and therefore become more heterogeneous and less stable, what induces dramatic changes in algorithm design and scheduling strategies.

- Graal (INRIA R.A.), Mescal (INRIA R.A) and Algorille (LORIA) recently began to work on this evolution. They strongly differ from our project since they mostly concentrate on task scheduling, whereas we mostly deal with communication scheduling and queries. One notable exception is the application described in Section 7.1. Nevertheless, in the context of this application, the design of the overlay network and compact data structures are the most important issues. Moreover, we collaborate on load balancing issues with these projects in the ANR ARA Alpage project (lead by O. Beaumont). These projects mostly concentrate on small instabilities due do changes in resource performance. In this context, overlay networks are not useful since it is possible to discover the actual topology of the platform.

- Grand Large (INRIA Futurs Orsay) is also working on independent tasks distribution on large scale distributed dynamic platforms. They designed XTremWeb, a software platform for distributing such applications. Grand Large mainly concentrates on platform design and experimentation rather than the design of scheduling strategies or optimal algorithms. They also designed several strategies devoted to security issues (sandboxing, identification,...) that will be useful for the actual deployment of the applications we propose.

On the other hand, many projects originally devoted to distributed algorithms and distributed systems concentrate on large scale dynamic platforms, due to the huge success of P2P file sharing applications. These projects can be split into two main categories, depending on their main focus.

- TREC (INRIA Rocquencourt), MAESTRO (INRIA Sophia) and GANG (Inria Rocquencourt) work on the understanding of structural properties of large scale dynamic networks and the design of realistic models. This aspect is not covered in the Cepage proposal [12], but structural properties (such as low doubling dimension and small world characteristics) and actual traces of P2P networks will be used in the design of both overlay networks, compact data structures and efficient algorithms. The members of GANG also deal with the design of algorithms and applications on P2P networks. On this particular point, there are several differences between GANG and Cepage

  - We consider collaborative environments and target the design of optimal algorithms (or at least approximation algorithms) for a few well-identified applications described in Section 7 whereas GANG deals with incitation mechanisms in non-collaborative environments.

[12] In fact, this part of the Cepage project has been removed in oder to avoid intersections with these projects.

  - Most of the applications we consider in Cepage associate computations and communications. Our goal is to design overlay networks and task distribution algorithms specific to computation intensive problems.
  - We concentrate on the design and proofs for randomized algorithms, based on the specific expertise of several members of the project (Philippe Duchon and Nicolas Hanusse). More generally, we come from several computer science communities (parallel computing, distributed algorithms, routing, wireless networks, probability), which strongly influences our methodology.
  - At last, the application fields are completely disjoint (communication primitives and distributed storage for GANG, computationally intensive applications, requests in databases and continuous integration for Cepage).

- Regal (INRIA Rocquencourt) and ASAP (IRISA) also deal with the design of algorithms on P2P platforms. REGAL deals with large scale replication problems for applications on dynamic distributed platforms. Members of REGAL mainly focus on the guarantee of the consistency between data replicas and on the adaptative configuration of the execution support at a low layer. Concerning the activity of Cepage, we do not concentrate on consistency but on algorithmic aspects of replication by taking into consideration trade-offs between amount of memory, load balancing and time complexity. We also share several goals with the ASAP project. The focus of ASAP is more on the design of algorithms and their practical experimentation rather than their theoretical analysis. Moreover, we focus on performance analysis of a limited number of applications (described in Section 7), graph theoretic work on overlay design and small world networks, analysis of randomized algorithms on large scale platforms and the design of compact data structures, that are not considered in ASAP. We started collaborations with ASAP on the design and analysis of geographic overlays in the ANR ARA Alpage project.

Finally, the main originality of our project is to gather expertise in scheduling of tasks and collective communication, graph theory, design of overlay networks, compact data structures, routing and randomized algorithms. We have tried to show in Section 7 how these expertises interact and are necessary to solve the target applications considered in this project proposal.

8.2 On the international scene

- **Cornell University**

  Jon Kleinberg is the pioneer of small world routing. Emin Gun Sirer has developed many practical projects on P2P systems and Internet geo-localization such as Meridian which is a P2P system for solving the nearest node location problem. This group has also a strong background on distributed systems and P2P with Ken Birman and Robbert van Rennesse. This group covers most of the domains we want to work on.

- **Microsoft Research Silicon Valley**

  Dahlia Malkhi and Alex Slivkins are joining the algorithms and theory team. Dahlia Malkhi is a leader in theoretical aspects of routing and P2P systems. Alex Slivkins is a leader of routing aspects in low doubling dimension metric networks. This research team covers a large part of the theoretical aspects we want to extend towards practice.

- **Theoretical aspects of routing**

David Peleg (Weizman Institute), Uri Zwick (Tel Aviv University), Mikkel Thorup (Copenhagen University) and Michael A. Bender (Stony Brook University) are leaders of theoretical aspects of routing and low stretch routing schemes. Active groups on routing with short tables include Andrew V. Goldberg and Dahlia Malkhi (Microsoft Research Silicon Valley), Bruce Maggs and Anupam Gupta (Carnegie Mellon University).

- **P2P algorithms**

  Active groups on P2P algorithms include MIT (with Frans Kaashoek and Robert Morris), Microsoft Research Cambridge (with Antony Rowstron and Miguel Castro), Berkeley (with Scott Shenker, John Kubiatowicz, and Ian Stoica), Weizman Institute (with Moni Naor), and the Max Plank Institute (with Peter Druschel).

- **Tree width, clique width and other graph width**

  Tree width was introduced by P. Seymour (Princeton). B. Courcelle (Bordeaux) studied relationship with graph grammars and logics and introduced cliquewidth. These graph parameters have been widely studied and applications to networks have been investigated by Cyril Gavoille (Bordeaux), Bruce Reed (Mc Gill, Montreal), Hans Bodlaender (Utrecht University).

- **Local data structures and other queries**

  Active groups in this area are Stephen Alstrup, Theis Rauhe (Copenhagen University), David Peleg, Amos Korman (Weizman Institute), Michal Katz, Nir A. Katz (Bar Ilan University), Uri Zwick, Tova Milo, Haim Kaplan (Tel Aviv University).

- **Overlay and small world networks**

  Active groups in this area are Ittai Abraham (Hebrew University of Jerusalem), Dahlia Malkhi (Microsoft Research Silicon Valley), Alan J. Demers, Jon Kleinberg (Cornell), Richard M. Karp, Scott Shenker (Berkeley), Christian Schindelhauer (Freiburg), Pierre Fraigniaud, Emmanuelle Lebhar (Liafa), Nicolas Schabanel (Lyon).

- **Network Coding**

  Active groups in this area are Rudolf Ahlswede (University of Bielefeld), Shuo-Yen Robert Li, Raymond W. Yeung (Chinese University of Hong Kong), Ralf Koetter (Technische Universität München) and Muriel Médard (MIT), Ying Zhu, Baochun Li, Jiang Guo (University of Toronto).

- **Decentralized multi-commodity flow algorithms**

  Active groups in this area are Baruch Awerbuch (Johns Hopkins University) and Frank Thomson Leighton (MIT).

- **Decentralized randomized network coding algorithms**

  Active groups in this area are Philip A. Chou (Microsoft Research Redmond), Tracey Ho (California Institute of Technology), Muriel Médard (MIT), Michelle Effros (CalTech), David R. Karger (MIT).

- **Content distribution on fully dynamic platforms**

  Active groups in this area are Peter Druschel (Max Planck Institute for Software Systems), Miguel Castro, Antony Rowstron (Microsoft Research Cambridge), Vivek S. Pai (Princeton), Muriel Médard (MIT).

- **Request and task scheduling on semi-stable platforms**

  Active groups in this area are Larry Carter and Jeanne Ferrante (U.C. San Diego), Viktor K. Prasanna (University of Southern California), Thomas G. Robertazzi (Stony Brook), Micah Adler, Arnold L. Rosenberg (University of Massachusetts).

- **Modeling of Dynamic Networks and Probablistic analysis**

  Active groups in this area are Christian Scheideler (Technische Universität München), Friedhelm Meyer auf der Heide (Paderborn), Robert Elsässer (Paderborn), Miroslaw Kutylowski, Miroslaw Korzeniowski, Marcin Bienkowski (Wroclaw), Christian Schindelhauer (Freiburg), Colin Cooper (London), Martin Dyer (Leeds), Berthold Vöcking (RWTH Aachen).

- **Content distribution using network coding**

  Active groups in this area are Christos Gkantsidis (Microsoft Research Cambridge), Pablo Rodriguez (Telefonica Research Lab Barcelona).

9 Collaborations and Grants

9.1 Current and Recent International Collaboration and Grants

- **EPSRC travel grant with King's College London and the University of Liverpool**

  Travel grant, 2006-2008, on "Models and Algorithms for Scale-Free Structures", in collaboration with the Department of Computer Science, King's College London, and the Department of Computer Science, the University of Liverpool. Funded by the EPSRC. Main investigators on the UK side: Colin Cooper (King's College London) and Michele Zito (University of Liverpool). Ralf Klasing is the principal investigator on the French side.

- **Royal Society Grant with King's College London**

  Bilateral Cooperation, 2004-2006, on "Web Graphs and Web Algorithms", in collaboration with the Department of Computer Science, King's College London. Funded by the Royal Society, U.K. Main investigators on the UK side: Colin Cooper and Tomasz Radzik. Ralf Klasing is the principal investigator on the French side.

- **European COST 293 Graal**

  European COST Action: "COST 293, Graal", 2004-2008. The main objective of this COST action is to elaborate global and solid advances in the design of communication networks by letting experts and researchers with strong mathematical background meet peers specialized in communication networks, and share their mutual experience by forming a multidisciplinary scientific cooperation community. This action has more than 25 academic and 4 industrial partners from 18 European countries. (http://www.cost293.org).

- **European Cost 295 DYNAMO**

  The COST 295 is an action of the European COST program (European Cooperation in the Field of Scientific and Technical Research) inside of the Telecomunications, Information Science and Technology domain (TIST). The acronym of the COST 295 Action, is DYNAMO and stands for "Dynamic Communication Networks". The COST295 Action is motivated by the need to supply a convincing theoretical framework for the analysis and control of all modern large networks induced by the interactions between decentralized and evolving computing entities, characterized by their inherently dynamic nature. (http://cost295.net/cost295/jsp/site/Portal.jsp)

9.2 List of academic collaborators abroad

- Juraj Hromkovič (ETH Zürich, Switzerland):
  Juraj Hromkovič is the Chair of Information Technology and Education at ETH Zürich, Switzerland). We collaborate with him and his group on probabilistic and approximation methods. This collaboration if manifested by mutual visits between the research groups. Several joint papers have been published.

- Leszek Gasieniec (University of Liverpool):
  Leszek Gasieniec is the Head of the Complexity Theory and Algorithmics Group in the Department of Computer Science at the University of Liverpool. We collaborate with him and his group on graph search and graph exploration. This collaboration is manifested by mutual research visits between the research groups. In 2006, Leszek Gasieniec visited the LaBRI for one month as a guest professor. Joint papers are in preparation.

- Joseph G. Peters (Simon Fraser University, Canada):
  Joseph G. Peters is the Head of the Network Modeling Group at Simon Fraser Unversity, Burnaby, Canda. We collaborate with him in the context of modeling and algorithms for network communication. This collaboration is manifested by mutual research visits between the research groups. Several joint papers have been published. Further joint papers are in preparation.

- Walter Unger (RWTH Aachen, Germany):
  Walter Unger is a permanent member of the research group on Algorithms and Complexity at RWTH Aachen. We collaborate with him and his group on algorithmic methods for network communication. This collaboration if manifested by mutual visits between the research groups. Several joint papers have been published. Further joint papers are in preparation.

- Michele Flammini (University of L'Aquila, Italy):
  Michele Flammini is the Head of the Algorithms and Computational Complexity at the University of L'Aquila, Italy. We collaborate with him and his group on algorithmic methods and modeling of network communication. This collaboration is manifested by mutual visits between the research groups. Several joint papers have been published. Further joint papers are in preparation.

- Andrzej Pelc (Université du Québec en Outaouais, Canada):
  Andrzej Pelc is the Research Chair in Distributed Computing at the Université du Québec en Outaouais. We collaborate with him and his group on gathering and rendezvous problems in distributed networks. This collaboration is manifested by mutual research visits between the research groups. In 2006, Andrzej Pelc visited the LaBRI for one month as a guest professor. Several joint papers have been published. Further joint papers are in preparation.

- Colin Cooper, Tomasz Radzik (King's College London):
  Colin Cooper and Tomasz Radzik are permanent members of the Algorithm Design Group in the Department of Computer Science at King's College London. We have been collaborating with them within the framework of the Royal Society Grant on "Web Graphs and Web Algorithms" and the EPSRC grant on "Models and Algorithms for Scale-Free Structures". The collaboration is manifested by mutual visits between the research groups. Several joint papers have been published. Further joint papers are in preparation. Also, a joint grant application is in preparation.

- Evangelos Kranakis (Carleton University, Canada):
  Evangelos Kranakis is a professor at Carleton University. He wrote 3 books about Ad Hoc Networking, Cryptography and Computation Models. He contributed to 3 papers with Nicolas Hanusse and Danny Krizanc on mobile agents algorithmic.

- Larry Carter and Jeanne Ferrante (U.C. San Diego):
  Larry Carter and Jeanne Ferrante are professor at the University of California, San Diego. They both visited Labri for one week in 2006, and Olivier Beaumont stayed at U.C.S.D. for a total of one month since 2003. Several papers have been jointly written (IPDPS'02, IPDPS'06, IEEE TPDS) and a joint paper (IEEE TPDS) is currently under revision.

- Henri Casanova (University of Hawaii at Manoa):
  Henri Casanova is a professor at the University of Manoa. He has been working during

a stay in 2005 with Olivier Beaumont on divisible load theory (joint work published in Parallel Computing).

- David Peleg (Weizmann Institute, Israel):
  David Peleg is a specialist in distributed computing. He has wrote a book in distributed computing and has more than hundred articles in journals, and 18 papers coauthored with Gavoille. He has been Professor invited of Bordeaux University few years ago, and came several times (short visit) in LaBRI for a common 3-year bilateral project on "graph labeling" with Gavoille. His last visit was december 2006 for a Ph.D defense.

- Dahlia Malkhi, Andrew V. Goldberg, and Udi Wieder (Microsoft Research, US):
  All are coauthors of Gavoille. Recently Malkhi was PC-chair of PODC '06, the world top conference in distributed computing, and with Gavoille they were organizators of an international the workshop "LOCALITY" joint with DISC '05 conference. Malkhi is a specialist in distributed computing, and together with her student Ittai Abraham, have more than 10 papers coauthored with Gavoille. Abraham visited LaBRI in 2004, and Gavoille visited Abraham at Jerusalem University in 2005.

- Mikkel Thorup (ATT Bell Labs, US):
  Mikkel Thorup is an expert in algorithms and data structures, and has tens of STOC and FOCS papers. Recently he wrote several papers on compact routing with Uri Zwick (Tel Aviv University, IL), and one with Gavoille.

9.3 List of industrial collaborators abroad

- Cyril Banino (Yahoo, Trondheim, Norway):
  Cyril Banino did his Master degree at the University of Bordeaux in 2002 under the supervision of Olivier Beaumont and his PhD in Trondheim (N.T.N.U.). During his PhD, he worked with Olivier Beaumont on decentralized algorithms for independent tasks scheduling. This collaboration is manifested by several research visits (for a total of 5 weeks since 2003) and several joint papers (IEEE TPDS, Europar'06, IPDPS'03). He has been recently appointed at Yahoo (Trondheim), and we plan to establish a formal collaboration on document storage in large distributed databases, request scheduling and independent tasks distribution across large distributed platforms.

- Dahlia Malki (Microsoft Research Silicon Valley, California):
  Dahlia Malkhi is member of the "Distributed Systems" group and of the "Algorithms and Theory" group at Microsoft Research - Silicon Valley (MSR-SCV). In order to strenghten the already well-established collaboration with Dahlia we plan the two following actions: 1) Gavoille plan to visit MSR-SCV as consultant in a near future; and 2) to write a proposal between LaBRI and Microsoft for student exchange and funding, and in order to organize visits between members of our two teams. The themes that have been mutually selected are "Broadcasting with contents" and "Tree-likeness of the Internet network".

9.4 On the National Scene

- ANR "programme blanc" Aladdin (2007-2011) Participants: Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Cyril Gavoille, Nicolas Hanusse, David Ilcinkas, Ralf Klasing. The scientific objectives of ALADDIN are to solve what are identified as the most challenging problems in the theory of interaction networks. The ALADDIN project is thus an opportunity to create a full continuum from fundamental research to applications in coordination with both INRIA project-teams CEPAGE and GANG.

- ANDT "Aladdin" (submitted) Participants: Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Lionel Eyraud-Dubois, Cyril Gavoille, Nicolas Hanusse, David Ilcinkas, Ralf Klasing. The objective of the (submitted) ANDT INRIA Aladdin 12 project is to continue and strengthen the efforts to develop the GRID'5000 platform. Olivier Beaumont is the local coordinator of Grid'5000 and is also responsible (together with Frederic Vivien, INRIA project GRAAL) of the national working group on "Efficient exploitation of highly heterogeneous and hierarchical large-scale systems".

- ANR ARA "Masse de données" Alpage (Leader: Olivier Beaumont, 2006–2009):
  Alpage focuses on the design of algorithms on large scale platforms. In particular, we will tackle the following problems

  - Large scale distributed platforms modeling
  - Overlay network design
  - Scheduling for regular parallel applications
  - Scheduling for applications sharing large files.

  The project involves the following INRIA and CNRS teams : Cepage, Graal, Mescal, Algorille, ASAP, LRI and LIX

- ACI "Masse de Données" GeoComp (2004-2007):
  Cyril Gavoille and Nicolas Bonichon participates in this ACI lead by Gilles Schaeffer (LIX). GEOCOMP tackles the problem of coding geometric data structures. Members of this project propose effective solutions to do the compression almost optimally without the need of a decompression process for basic requests on the structure.

10 Teaching Activities and Scientific Responsibilities

10.1 Teaching activities

The members of CEPAGE are heavily involved in teaching activities at undergraduate level (Licence 1, 2 and 3, Master 1 and 2, Engineering Schools ENSEIRB). The teaching is carried out by members of the University as part of their teaching duties, and for CNRS (at master 2 level) as extra work. It represents more than 500 hours per year.

At master 2 level, here is a list of courses taught the last two years:

- Nicolas Hanusse
  – Graph algorithms for data visualization (2nd year MASTER "Models and Algorithms" - 2005 and 2006)
  – Distributed computing (2nd year MASTER "Models and Algorithms" - 2006)

- Cyril Gavoille
  – Introduction to Distributed Computing (2nd year MASTER "Models and Algorithms" - 2005, 2006)
  – Algorithms and Communications in Networks (2nd year MASTER "Models and Algorithms" - 2005, 2006)
  – Communication and Routing (last year of engineering school ENSEIRB 2005, 2006)

- Olivier Beaumont
  – Routing and P2P Networks (last year of engineering school ENSEIRB, 2005)

- Philippe Duchon
  – Randomized Algorithms (2nd year MASTER "Models and Algorithms" - 2006)

10.2 Program Committees (since 2003)

10.2.1 Program Chair

- HeteroPar 07 (Olivier Beaumont, chair), International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks, Austin, 2007

- EuroPar'07 (Olivier Beaumont, Local Chair, Scheduling and Load Balancing), Rennes, France, 2007

- RenPar 06 (Olivier Beaumont, co-chair) Rencontre du Parallélisme, Perpignan, 2006

- LOCALITY '05 (Cyril Gavoile, co-chair, workshop co-located with DISC '05, sep. 26, Cracow, Poland) Locality Preserving Distributed Computing Methods

- AlgoTel '03 (co-chair, May 12-14, Banyuls-sur-mer, France) Rencontres Francophones sur les aspects Algorithmiques des Télécommunications

10.2.2 Program Committees

- Olivier Beaumont
  - HeteroPar 08 (Olivier Beaumont, chair), International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks, Tsukuba, Japan
  - PMAA 08 International Workshop on Parallel Matrix Algorithms and Applications, Neuchâtel, Switzerland

- RenPar 08 Rencontre du Parallélisme, Fribourg, Switzerland
- IPDPS 07 IEEE International Parallel and Distributed Processing Symposium, Long Beach, USA, 2007
- PMGC'07 Workshop on Programming Models for Grid Computing, Rio de Janeiro, Brazil, 2007
- IPDPS 06 IEEE International Parallel and Distributed Processing Symposium, Rhodes Island, Greece
- ICPADS 06 International Conference on Parallel and Distributed Systems (Minneapolis, USA, 2006)
- HeteroPar 06 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Barcelona, Spain)
- PMAA 06 International Workshop on Parallel Matrix Algorithms and Applications, Rennes, France
- IPDPS 05 IEEE International Parallel and Distributed Processing Symposium, Denver Colorado, USA
- HeteroPar 05 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Boston, Massachusetts, USA)
- RenPar 05 Rencontre du Parallélisme, Le Croisic, France
- HeteroPar 04 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Cork, Ireland)
- HeteroPar 03 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Czestochowa, Poland)
- PMAA 02 International Workshop on Parallel Matrix Algorithms and Applications, Neuchâtel, Switzerland

- Cyril Gavoille
  - PODC '08 ((Aug. 18-21, Toronto, Canada) Annual ACM Symposium on Principles of Distributed Computing
  - DISC '08 (Sep. 22-24, Arcachon, France) International Symposium on Distributed Computing
  - AlgoTel '08 (May 14-16, Saint-Malo, France)
  - JDIR '08 (Jan. 16-18, Villeneuve d'Ascq, France)
  - DISC '07 (Sep/Oct, Lemesos, Cyprus) International Symposium on Distributed Computing
  - SPAA '07 (June 9-11, San Diego, Californie, USA) Symposium on Parallelism in Algorithms and Architectures
  - AlgoTel '07 (May 29 - Jun 1, Ile d'Oléron, France) Rencontres Francophones sur les aspects Algorithmiques des Télécommunications
  - PDCN '07 (Feb. 13-15, Innsbruck, Austria) Parallel and Distributed Computing and Networks
  - PODC '06 (July 23-26, Denver, Colorado, USA) Annual ACM Symposium on Principles of Distributed Computing
  - PDCN '06 (Feb. 14-16, Innsbruck, Austria) Parallel and Distributed Computing and Networks

- PODC '05 (Jul. 17-20, Las Vegas, Nevada, USA) Annual ACM Symposium on Principles of Distributed Computing
- PDCN '05 (Feb. 15-17, Innsbruck, Austria) Parallel and Distributed Computing and Networks
- HiPC '05 (Dec. 18-21, Goa, India) International Conference On High Performance Computing
- IWDC '05 (Dec. 27-30, Kharagpur, India) International Workshop on Distributed Computing
- STACS '04 (Mar. 25-27, Montpellier, France) Symposium on Theoretical Aspects of Computer Science
- SIROCCO '04 (Jun. 21-23, Smolenice, Slovakia) Colloquium on Structural Information and Communication Complexity
- SIROCCO '03 (Jun. 18-20, Umeå, Sweden) Colloquium on Structural Information and Communication Complexity

- Nicolas Hanusse
  - ALGOTEL 06 Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Tregastel, France
  - ALGOTEL 05 Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Presqu'île de Giens, France
  - Scientific chair and main organisator of Research School "Interaction and Data Visualization", september 2004, Bordeaux

- Ralf Klasing
  - SIROCCO 06 13th Colloquium on Structural Information and Communication Complexity (2006, Chester, United Kingdom).

- Philippe Duchon
  - ALGOTEL 08 Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Saint Malo, France

11 Short biographies of Cepage permanent members

- **Olivier Beaumont** received his PhD degree from the University of Rennes in 1999. From 1999 to 2001, he was assistant professor at Ecole Normale Supérieure de Lyon. Then, he was appointed at ENSEIRB in Bordeaux. In 2004, he defended his "habilitation à diriger les recherches". His research interests focus on the design of parallel and distributed algorithms, overlay networks on large scale heterogeneous platforms and combinatorial optimization.

- **Nicolas Bonichon** received his PhD degree from the Université Bordeaux 1 in 2002. He has been holding a position as assistant professor in Université Bordeaux 1 since 2004. His research interests include distributed algorithms, compact data structure, graph drawing and enumerative combinatorics.

- **Lionel Eyraud-Dubois** received his PhD degree from the University Joseph Fourier of Grenoble in 2006. He joined the INRIA as a permanent researcher in 2007. His research interests include scheduling, combinatorics and distributed computing.

- **Philippe Duchon** received the PhD degree from Université Bordeaux 1 in 1998. He has been holding a position as assistant professor at ENSEIRB since 1999. His research interests range from enumerative combinatorics to distributed computing, with a focus on random models and randomized algorithms.

- **Cyril Gavoille** received the PhD degree from Ecole Normale Supérieure of Lyon in 1996. From 1996 to 2002, he was assistant professor in Bordeaux 1 University. In 2000, he defended his "habilitation à diriger les recherches" and became professor in 2002. His research interests focus on distributed compact data structures and graph algorithmic.

- **Nicolas Hanusse** received the PhD degree from Université Bordeaux 1 in 1997. As a post-doc, he spent one year at Carleton University, Canada, in 2000. In 2001, he had a position of assistant professor at LRI, Paris-Sud University. He is currently a CNRS permanent researcher in the LaBRI laboratory of Bordeaux. He is mainly interested in distributed computing and graph algorithmic.

- **David Ilcinkas** received his PhD degree from the University of Paris 11 in July 2006. Then he spent one year at the Université du Québec en Outaouais as a postdoctoral fellow, in collaboration with the University of Ottawa and Carleton University, in Canada. Since November 2007, he is a CNRS researcher at LaBRI, Bordeaux. His research is mainly focused on computing by mobile entities and distributed computing in general.

- **Ralf Klasing** received the PhD degree from the University of Paderborn in 1995. From 1995 to 1997, he was an Assistant Professor at the University of Kiel. From 1997 to 1998, he was a Research Fellow at the University of Warwick. From 1998 to 2000, he was an Assistant Professor at RWTH Aachen. From 2000 to 2002, he was a Lecturer at King's College London. In 2002, he joined the CNRS as a permanent researcher. From 2002 to 2005, he was affiliated to the laboratory I3S in Sophia Antipolis. Currently, he is affiliated to the laboratory LaBRI in Bordeaux. His research interests include communication algorithms in networks, algorithmic methods for combinatorially hard problems, web graphs and web algorithms, optimization problems in ad-hoc wireless networks, and graph exploration.

12 Publications of project members (since 2004) in relationship with the project

References

12.1 Books and Habilitation Thesis

[1] Olivier Beaumont. *Nouvelles méthodes pour l'ordonnancement sur plates-formes hétérogènes*. PhD thesis, Habilitation à diriger des recherches de l'Université de Bordeaux 1, December 2004.

[2] Cyril Gavoille. *Structures de données compactes et distribuées*. PhD thesis, December 2000. Thèse d'habilitation à diriger les recherches, Université de Bordeaux.

[3] J. Hromkovič, R. Klasing, A. Pelc, P. Ružička, and W. Unger. *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*. Springer Monograph. Springer-Verlag, 2005.

12.2 Articles in refereed journals and book chapters

[4] Stephen Alstrup, Cyril Gavoille, Haim Kaplan, and Theis Rauhe. Nearest common ancestors: A survey and a new algorithm for a distributed environment. *Theory of Computing Systems*, 37:441–456, 2004.

[5] Cyril Banino, Olivier Beaumont, Larry Carter, Jeanne Ferrante, Arnaud Legrand, and Yves Robert. Scheduling strategies for master-slave tasking on heterogeneous processor platforms. *IEEE Trans. Parallel Distributed Systems*, 15(4):319–330, 2004.

[6] Beaumont, Olivier and Carter, Larry and Ferrante, Jeanne and Legrand, Arnaud and Marchal, Loris and Robert, Yves. Centralized Versus Distributed Schedulers Multiple Bag-of-Tasks Applications. *IEEE Trans. Parallel Distributed Systems*, 2007. To appear.

[7] O. Beaumont, L. Marchal, and Y. Robert. Complexity results for collective communications on heterogeneous platforms. *Int. Journal of High Performance Computing Applications*, 20(1):5–17, 2006.

[8] Olivier Beaumont, Henri Casanova, Arnaud Legrand, Yves Robert, and Yang Yang. Scheduling divisible loads on star and tree networks: results and open problems. *IEEE Trans. Parallel Distributed Systems*, 16(3):207–218, 2005.

[9] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Pipelining broadcasts on heterogeneous platforms. *IEEE Trans. Parallel Distributed Systems*, 16(4):300–313, 2005.

[10] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Steady-state scheduling on heterogeneous clusters. *Int. J. of Foundations of Computer Science*, 16(2), 2005.

[11] Jean-Claude Bermond, Jérôme Galtier, Ralf Klasing, Nelson Morales, and Stéphane Pérennes. Hardness and approximation of gathering in static radio networks. *Parallel Processing Letters*, 16(2):165–183, June 2006.

[12] H.-J. Böckenhauer, D. Bongartz, J. Hromkovič, R. Klasing, G. Proietti, S. Seibert, and W. Unger. On the hardness of constructing minimal 2-connected spanning subgraphs in complete graphs with sharpened triangle inequality. *Theoretical Computer Science*, 326(1–3):137–153, 2004.

[13] N. Bonichon. A bijection between realizers of maximal plane graphs and pairs of non-crossing dyck paths. *Discrete Mathematics*, 298:104–114, 2005. FPSAC'02 Special Issue.

[14] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected planar graphs. *Algorithmica*, 47(4):399–420, 2007. To appear.

[15] N. Bonichon, B. Le Saëc, and M. Mosbah. Orthogonal drawings based on the stratification of planar graphs. *Discrete Mathematics*, 276(1-3):43–57, 2004. Special issue: 6th International Conference on Graph Theory - Edited by J.-L. Fouquet, I. Rusu.

[16] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, Dominique Poulalhon, and Gilles Schaeffer. Planar graphs, via well-orderly maps and trees. *Graphs and Combinatorics*, 22(2):185–202, 2006.

[17] C. Cooper, R. Klasing, and M. Zito. Lower bounds and algorithms for dominating sets in web graphs. *Internet Mathematics*, 2007. To appear.

[18] Pierre Fraigniaud and Cyril Gavoille. Header-size lower bounds for end-to-end communication in memoryless networks. *Computer Networks*, 2004.

[19] Cyril Gavoille and Martin Nehéz. Interval routing in reliability networks. *Theoretical Computer Science*, 333(3):415–432, 2005.

[20] Cyril Gavoille, David Peleg, Stéphane Pérennès, and Ran Raz. Distance labeling in graphs. *Journal of Algorithms*, 53(1):85–112, 2004.

[21] Nicolas HANUSSE, Evangelos Kranakis, and Danny Krizanc. Searching with mobile agents in networks with liars. *Discrete Applied Mathematics*, 137(1):69–85, 2004.

[22] R. Klasing and C. Laforest. Hardness results and approximation algorithms of k-tuple domination in graphs. *Information Processing Letters*, 89(2):75–83, 2004.

[23] Ralf Klasing, Christian Laforest, Joseph G. Peters, and Nicolas Thibault. Constructing incremental sequences in graphs. *Algorithmic Operations Research*, 1(2):1–7, 2006.

[24] Ralf Klasing, Euripides Markou, Tomasz Radzik, and Fabiano Sarracco. Hardness and approximation results for black hole search in arbitrary graphs. *Theoretical Computer Science*, 2007. to appear.

[25] Colin Cooper, Ralf Klasing, and Tomasz Radzik. A randomized algorithm for the joining protocol in dynamic distributed networks. *Theoretical Computer Science*, 2007. to appear.

[26] Yon Dourisboure, Feodor F. Dragan, Cyril Gavoille, and Chenyu Yan. Spanners for bounded tree-length graphs. *Theoretical Computer Science*, 383(1):34–44, September 2007.

[27] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Could any graph be turned into a small-world? *Theoretical Computer Science*, 355(1):96–103, 2006.

[28] P. Duchon, N. Hanusse, N. Saheb, and A. Zemmari. Broadcast in the rendezvous model. *Information and Computation*, 204(5):697–712, 2006.

[29] Michele Flammini, Ralf Klasing, Alfredo Navarra, and Stéphane Pérennes. Improved approximation results for the minimum energy broadcasting problem in wireless ad hoc networks. *Algorithmica*, 2007. Online First.

[30] Michele Flammini, Ralf Klasing, Alfredo Navarra, and Stéphane Pérennes. Tightening the upper bound for the minimum energy broadcasting. *Wireless Networks*, 2007. Online First.

[31] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. *Distributed Computing*, 18(4):279–291, March 2006.

[32] Leszek Gą asieniec, Ralf Klasing, Russell Martin, Alfredo Navarra, and Xiaohui Zhang. Fast periodic graph exploration with constant memory. *Journal of Computer and System Sciences*, 2007. to appear.

[33] Sylvain Gravier, Ralf Klasing, and Julien Moncel. Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs. *Algorithmic Operations Research*, 2007. to appear.

[34] Subir Bandyopadhyay (in cooperation with Ralf Klasing). *Dissemination of Information in Optical Networks: From Technology to Algorithms*. Springer Monograph. Springer-Verlag, 2007. to appear.

[35] Ralf Klasing, Euripides Markou, and Andrzej Pelc. Gathering asynchronous oblivious mobile robots in a ring. *Theoretical Computer Science*, 2007. to appear.

[36] Ralf Klasing, Euripides Markou, Tomasz Radzik, and Fabiano Sarracco. Approximation bounds for black hole search problems. *Networks*, 2007. to appear.

[37] Ralf Klasing, Euripides Markou, Tomasz Radzik, and Fabiano Sarracco. Hardness and approximation results for black hole search in arbitrary graphs. *Theoretical Computer Science*, 384(2–3):201–221, October 2007.

[38] Ralf Klasing, Nelson Morales, and Stéphane Pérennes. On the complexity of bandwidth allocation in radio networks. *Theoretical Computer Science*, 2007. to appear.

[39] Reuven Cohen, Pierre Fraigniaud, David Ilcinkas, Amos Korman, and David Peleg. Label-guided graph exploration by a finite automaton. *ACM Transactions on Algorithms*, to appear.

[40] Reuven Cohen, Pierre Fraigniaud, David Ilcinkas, Amos Korman, and David Peleg. Labeling schemes for tree representation. *Algorithmica*, to appear.

[41] Pierre-François Dutot, Lionel Eyraud, Grégory Mounié, and Denis Trystram. Scheduling on large scale distributed platforms: from models to implementations. *Intl. Journal of Foundations of Computer Science*, 16(2):217–237, 2005.

[42] Lionel Eyraud. A pragmatic analysis of scheduling environments on new computing platforms. *Intl. Journal of High Performance Computing and Applications*, 20:507–516, 2006.

[43] Pierre Fraigniaud, David Ilcinkas, Guy Peer, Andrzej Pelc, and David Peleg. Graph exploration by a finite automaton. *Theoretical Computer Science*, 345(2-3):331–344, November 2005.

[44] Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. Impact of memory size on graph exploration capability. *Discrete Applied Mathematics*, to appear.

[45] David Ilcinkas and Andrzej Pelc. Impact of asynchrony on the behavior of rational selfish agents. *Fundamenta Informaticae*, 82(1-2):113–125, 2008.

12.3 Publications in Conferences and Workshops

[46] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, and Udi Wieder. Strong-diameter decompositions of minor free graphs. In 19^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 16–24. ACM Press, June 2007.

[47] O. Beaumont, N. Bonichon, and L. Eyraud-Dubois. Scheduling divisible workload on heterogeneous platforms under bounded multi-port model. In *International Heterogeneity in Computing Workshop*, 2008.

[48] Olivier Beaumont, Abhay Ghatpande, Hidenori Nakazato and Hiroshi Watanabe. SDivisible Load Scheduling with Result Collection on Heterogeneous Systems. In *International Heterogeneity in Computing Workshop*, 2008.

[49] Olivier Beaumont, Philippe Duchon and Mirek Korzeniowski Heterogenous dating service with application to rumor spreading. In *International Parallel and Distributed Processing Symposium IPDPS*, 2008.

[50] O. Beaumont and Abdou Guermouche. Task scheduling for parallel multifrontal methods. In *Euro-Par 2007 Parallel Processing*. Lecture Notes in Computer Science, 2007.

[51] O. Beaumont, A.M. Kermarrec, L. Marchal, and E. Rivière. VoroNet: A scalable object network based on Voronoi tessellations. In *International Parallel and Distributed Processing Symposium IPDPS*, 2007.

[52] O. Beaumont, A.M. Kermarrec, and E. Rivière. Peer to peer multidimensional overlays: Approximating complex structures. In *OPODIS*, 2007.

[53] Bruno Courcelle, Cyril Gavoille, Mustapha Kanté, and David Andrew Twigg. Forbidden-set labeling on graphs. In 2^{nd} *Workshop on Locality Preserving Distributed Computing Methods (LOCALITY)*, August 2007. Co-located with PODC 2007.

[54] Youssou Dieng and Cyril Gavoille. Routage dans les graphes cellulaires. In $9^{èmes}$ *Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, pages 91–94, May 2007.

[55] Philippe Duchon, Nicole Eggemann, and Nicolas Hanusse. Non-navigability of random scale-free graphs. In *OPODIS - International Conference Of Principles of Distributed Systems*, 2007.

[56] Philippe Duchon, Nicole Eggemann, and Nicolas Hanusse. Non-searchability of random scale-free graphs. In *PODC - Principles Of Distributed Computing*, pages 380–381, 2007.

[57] Pierre Fraigniaud, Cyril Gavoille, Adrian Kosowski, Emmanuelle Lebhar, and Zvi Lotker. Universal augmentation schemes for network navigability: Overcoming the sqrt(n)-barrier. In *19th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 1–7, 2007.

[58] Leszek Gąsieniec, Ralf Klasing, Russell Martin, Alfredo Navarra, and Xiaohui Zhang. Fast periodic graph exploration with constant memory. In *Proceedings of the 14th Colloquium on Structural Information and Communication Complexity (SIROCCO 2007)*, volume 4474 of *Lecture Notes in Computer Science*, pages 26–40. Springer Verlag, June 2007.

[59] Cyril Gavoille. Localized data structures (keynote talk). In 2^{nd} *Workshop on Locality Preserving Distributed Computing Methods (LOCALITY)*, August 2007. Co-located with PODC 2007.

[60] Cyril Gavoille. An overview on compact routing. In *Workshop on Peer-to-Peer, Routing in Complex Graphs, and Network Coding*, March 2007.

[61] Cyril Gavoille, Ralf Klasing, Adrian Kosowski, and Alfredo Navarra. On the complexity of distributed greedy coloring. In *Proceedings of the 21st International Symposium on Distributed Computing (DISC 2007)*, volume 4731 of *Lecture Notes in Computer Science*, pages 482–484. Springer Verlag, September 2007.

[62] Cyril Gavoille and Arnaud Labourel. Brief annoucement: On local representation of distances in trees. In 26^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 246–247. ACM Press, August 2007.

[63] Cyril Gavoille and Arnaud Labourel. Distributed relationship schemes for trees. In 18^{th} *Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 4835 of Lecture Notes in Computer Science, pages 728–738. Springer, December 2007.

[64] Cyril Gavoille and Arnaud Labourel. Shorter implicit representation for planar graphs and bounded treewidth graphs. In Lars Arge and Emo Welzl, editors, 15^{th} *Annual European Symposium on Algorithms (ESA)*, volume 4698 of Lecture Notes in Computer Science, pages 582–593. Springer, October 2007.

[65] Ralf Klasing, Adrian Kosowski, and Alfredo Navarra. Cost minimisation in multi-interface networks. In *Proceedings of the 1st Annual International Conference on Network Control and Optimization (NET-COOP 2007)*, volume 4465 of *Lecture Notes in Computer Science*, pages 276–285. Springer Verlag, June 2007.

[66] Ittai Abraham and Cyril Gavoille. Object location using path separators. In 25^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 188–197. ACM Press, July 2006.

[67] Ittai Abraham, Cyril Gavoille, Andrew V. Goldberg, and Dahlia Malkhi. Routing in networks with low doubling dimension. In 26^{th} *International Conference on Distributed Computing Systems (ICDCS)*. IEEE Computer Society Press, July 2006.

[68] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Routing with improved communication-space trade-off. In 18^{th} *International Symposium on Distributed Computing (DISC)*, volume 3274 of Lecture Notes in Computer Science, pages 305–319. Springer, October 2004.

[69] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Compact routing for graphs excluding a fixed minor. In 19^{th} *International Symposium on Distributed Computing (DISC)*, volume 3724 of Lecture Notes in Computer Science, pages 442–456. Springer, September 2005.

[70] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. On space-stretch trade-offs: Lower bounds. In 18^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 217–224. ACM Press, July 2006.

[71] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. On space-stretch trade-offs: Upper bounds. In 18^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 207–216. ACM Press, July 2006.

[72] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In 16^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 20–24. ACM Press, July 2004.

[73] C. Banino, O Beaumont, and L. Natvig. Master-slave tasking on asymmetric networks. In *Proceedings of Euro-Par 2006*, volume 4128 of *Lecture Notes in Computer Science*, pages 437–447, Dresden, Germany, August 2006. Springer.

[74] Fabrice Bazzaro and Cyril Gavoille. Localized and compact data-structure for comparability graphs. In 16^{th} *Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 3827 of Lecture Notes in Computer Science, pages 1122–1131. Springer, December 2005.

[75] O. Beaumont and V. Boudet, editors. *Perpignan, Octobre 2006*. Actes de Renpar 2006, 2006.

[76] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, L. Marchal, and Y. Robert. Centralized versus distributed schedulers for multiple bag-of-task applications. In *International Parallel and Distributed Processing Symposium IPDPS'2006*. IEEE Computer Society Press, 2006.

[77] O. Beaumont, E.M. Daoudi, N. Maillard, P. Manneback, and J.-L. Roch. Tradeoff to minimize extra-computations and stopping criterion tests for parallel iterative schemes. In *PMAA'04 Parallel Matrix Algorithms and Applications*. CIRM, Marseille, 2004.

[78] O. Beaumont, A.-M. Kermarrec, L. Marchal, and E Riviere. Voronet: A scalable object network based on voronoi tessellations. In *International Parallel and Distributed Processing Symposium IPDPS'2007*. IEEE Computer Society Press, 2007.

[79] O. Beaumont, L. Marchal, V. Rehn, and Y. Robert. Fifo scheduling of divisible loads with return messages under the one-port model. In *Heterogeneous Computing Workshop HCW'2006*. IEEE Computer Society Press, 2006.

[80] Olivier Beaumont, Vincent Boudet, Pierre-François Dutot, Yves Robert, and Denis Trystram. *Informatique répartie : architecture, parallélisme et système*, chapter "Fondements théoriques pour la conception d'algorithmes efficaces de gestion de ressources". Hermès Publications, 2004.

[81] Olivier Beaumont, Larry Carter, Jeanne Ferrante, Arnaud Legrand, Loris Marchal, and Yves Robert. Centralized versus distributed schedulers for multiple bag-of-task applications. In *International Parallel and Distributed Processing Symposium IPDPS'2006*. IEEE Computer Society Press, accepted for presentation, 2006.

[82] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Assessing the impact and limits of steady-state scheduling for mixed task and data parallelism on heterogeneous platforms. In *HeteroPar'2004: International Conference on Heterogeneous Computing, jointly published with ISPDC'2004: International Symposium on Parallel and Distributed Computing*. IEEE Computer Society Press, 2004.

[83] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Complexity results and heuristics for pipelined multicast operations on heterogeneous platforms. In *2004 International Conference on Parallel Processing (ICPP'2004)*, pages 267–274. IEEE Computer Society Press, 2004.

[84] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Pipelining broadcasts on heterogeneous platforms. In *IPDPS'2004*. IEEE Computer Society Press, 2004.

[85] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Steady-state scheduling on heterogeneous clusters: why and how? In *6th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2004*. IEEE Computer Society Press, 2004.

[86] Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert. Independent and divisible tasks scheduling on heterogeneous star-schaped platforms with limited memory. In *PDP'2005, 13th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 179–186. IEEE Computer Society Press, 2005.

[87] Olivier Beaumont, Loris Marchal, Veronika Rehn, and Yves Robert. Fifo scheduling of divisible loads with return messages under the one-port model. In *Heterogeneous Computing Workshop HCW'2006*. IEEE Computer Society Press, accepted for presentation, 2006.

[88] Olivier Beaumont, Loris Marchal, and Yves Robert. Broadcast trees for heterogeneous platforms. In *International Parallel and Distributed Processing Symposium IPDPS'2005*. IEEE Computer Society Press, 2005.

[89] Olivier Beaumont, Loris Marchal, and Yves Robert. Scheduling divisible loads with return messages on heterogeneous master-worker platforms. In *International Conference on High Performance Computing HiPC'2005*, LNCS. Springer Verlag, 2005.

[90] J.-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Pérennes. Hardness and approximation of gathering in static radio networks. In *FAWN06*, Pisa, Italy, March 2006.

[91] Jean-Claude Bermond, Jérôme Galtier, Ralf Klasing, Nelson Morales, and Stéphane Pérennes. Gathering in specific radio networks. In *8èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel06), Trégastel*, May 2006.

[92] Jean-Claude Bermond, Jérôme Galtier, Ralf Klasing, Nelson Morales, and Stéphane Pérennes. Hardness and approximation of gathering in static radio networks. In *FAWN06, Pisa, Italy*, March 2006.

[93] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected planar graphs - (extended abstract). In *Graph Drawing: 12th International Symposium, GD 2004*, volume 3383 of *LNCS*, pages 60–70, 2004.

[94] N. Bonichon, CYRIL GAVOILLE, Nicolas HANUSSE, D. POULALHON, and Gilles SCHAEFFER. Planar graphs, via well-orderly maps and trees. In 30^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 3353 of Lecture Note. Springer, 2004. 270-284.

[95] Nicolas Bonichon, Cyril Gavoille, and Arnaud Labourel. Adjacency labeling for bounded degree trees and applications. In 6^{th} *Czech-Slovak International Symposium on Combinatorics, Graph Theory, Algorithms and Applications*, July 2006. Dedicated to Jarik Nešetřil on the occasion of his 60th birthday.

[96] Nicolas Bonichon, Cyril Gavoille, and Arnaud Labourel. Short labels by traversal and jumping. In 13^{th} *International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, volume 4056 of Lecture Notes in Computer Science, pages 143–156. Springer, July 2006.

[97] C. Cooper, R. Klasing, and M. Zito. Dominating sets in web graphs. In *Proceedings of the Third Workshop on Algorithms and Models for the Web-Graph (WAW 2004)*, volume 3243 of *Lecture Notes in Computer Science*, pages 31–43. Springer-Verlag, 2004.

[98] Colin Cooper, Ralf Klasing, and Tomasz Radzik. Searching for black-hole faults in a network using multiple agents. In *Proceedings of the 10th International Conference on Principles of Distributed Systems (OPODIS 2006)*, volume 4305 of *Lecture Notes in Computer Science*, pages 320–332. Springer Verlag, December 2006.

[99] Bilel Derbel and Cyril Gavoille. Fast deterministic distributed algorithms for sparse spanners. In 13^{th} *International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, volume 4056 of Lecture Notes in Computer Science, pages 100–114. Springer, July 2006.

[100] A. Don and Nicolas Hanusse. A deterministic multidimensional scaling algorithm for data visualization. In *IEEE IV2006 - International Conference on Information Visualization*, pages 511–520. IEEE, July 2006.

[101] P. Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Could any graph be turned into a small-world ? In *Actes d'AlgoTel'2005 (conférence francophone sur les algorithmes de communications)*, pages –, Giens, Mai 2005.

[102] P. Duchon, Nicolas HANUSSE, Emmanuelle LEBHAR, and Nicolas SCHABANEL. Could any graph be turned into a small world ? In Pierre Fraigniaud, editor, *International Symposium on Distributed Computing (DISC)*, volume 3724 of *Lecture Notes in Computer Science*, pages 511–513. Springer Verlag, 2005.

[103] P. Duchon, Nicolas HANUSSE, Nasser SAHEB-DJAHROMI, and Akka ZEMMARI. Broadcast in the rendezvous model. In V. Diekert and M. Habib, editors, *Proceedings of STACS 2004*, volume 2996 of *Lecture Notes in Computer Science*, pages 559–570. Springer, 2004.

[104] P. Duchon, Nicolas HANUSSE, and Sébastien TIXEUIL. Optimal randomized self-stabilizing mutual exclusion on synchronous rings. In Rachid Guerraoui, editor, *Proceedings of DISC'2004*, number 3274 in Lecture Notes in Computer Science, pages 216–229, Amsterdam, October 2004. Springer.

[105] P. Duchon, Nicolas HANUSSE, and Sébastien TIXEUIL. Protocoles auto-stabilisants synchrones d'exclusion mutuelle pour les anneaux anonymes et uniformes. In *Actes d'AlgoTel 2004*, pages 135–140, Batz sur Mer, Mai 2004. Université de Rennes.

[106] Philippe Duchon, Nicolas Hanusse, Emmanuelle Lebhar, and Nicolas Schabanel. Towards small world emergence. In Uzi Vishkin, editor, *SPAA2006 - 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 225–232, PO box 11405, NY - 10286-6626, July 2006. ACM SIGACT - ACM SIGARCH, ACM Pess.

[107] M. Flammini, R. Klasing, A. Navarra, and S. Pérennes. Improved approximation results for the minimum energy broadcasting problem. In *2nd ACM/SIGMOBILE Annual International Joint Workshop on Foundation of Mobile Computing (DIALM-POMC 2004)*, pages 85–91. ACM Press, 2004.

[108] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. In 23^{rd} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 169–178. ACM Press, July 2004.

[109] Cyril Gavoille. Distributed data structures: A survey (invited talk). In 12^{th} *International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, volume 3499 of Lecture Notes in Computer Science, page 2. Springer, May 2005.

[110] Cyril Gavoille. Distributed data structures: A survey on informative labeling schemes (invited talk). In R. Královic and P. Urzyczyn, editors, 31^{st} *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 4162 Lecture Notes in Computer Science, page 38. Springer, August 2006.

[111] Cyril Gavoille. An overview on compact routing. In 2^{nd} *Research Workshop on Flexible Network Design*, October 2006.

[112] Cyril Gavoille and Olivier Ly. Distance labeling in hyperbolic graphs. In 16^{th} *Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 3827 of Lecture Notes in Computer Science, pages 1071–1079. Springer, December 2005.

[113] R. Klasing, Z. Lotker, A. Navarra, and S. Pérennes. From balls and bins to points and vertices. In *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, volume 3827 of *Lecture Notes in Computer Science*, pages 757–766. Springer Verlag, December 2005.

[114] R. Klasing, E. Markou, T. Radzik, and F. Sarracco. Approximation bounds for black hole search problems. In *Proceedings of the 9th International Conference on Principles of Distributed Systems (OPODIS 2005)*, Lecture Notes in Computer Science. Springer Verlag, December 2005.

[115] R. Klasing, E. Markou, T. Radzik, and F. Sarracco. Hardness and approximation results for black hole search in arbitrary graphs. In *Proceedings of the 12th Colloquium on Structural Information and Communication Complexity (SIROCCO 2005)*, volume 3499 of *Lecture Notes in Computer Science*, pages 200–215. Springer Verlag, May 2005.

[116] R. Klasing, A. Navarra, A. Papadopoulos, and S. Pérennes. Adaptive broadcast consumption (abc), a new heuristic and new bounds for the minimum energy broadcast routing problem. In *Proc. 3rd FIP-TC6 Networking Conference (Networking 2004)*, volume 3042 of *Lecture Notes in Computer Science*, pages 866–877. Springer-Verlag, 2004.

[117] Ralf Klasing, Euripides Markou, and Andrzej Pelc. Gathering asynchronous oblivious mobile robots in a ring. In *Proceedings of the 17th Annual International Symposium on Algorithms and Computation (ISAAC 2006)*, volume 4288 of *Lecture Notes in Computer Science*, pages 744–753. Springer Verlag, December 2006.

[118] Reuven Cohen, Pierre Fraigniaud, David Ilcinkas, Amos Korman, and David Peleg. Label-guided graph exploration by a finite automaton. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3580 of *LNCS*, pages 335–346, 2005.

[119] Reuven Cohen, Pierre Fraigniaud, David Ilcinkas, Amos Korman, and David Peleg. Labeling schemes for tree representation. In *Proceedings of the 7th International Workshop on Distributed Computing(IWDC)*, volume 3741 of *LNCS*, pages 13–24, 2005.

[120] Pierre-François Dutot, Lionel Eyraud, Grégory Mounié, and Denis Trystram. Models for scheduling on large scale platforms: Which policy for which application? In *Proc. 18th Intl. Parallel and Distributed Processing Symposium (IPDPS)*, 2004.

[121] Pierre-François Dutot, Lionel Eyraud, Grégory Mounié, and Denis Trystram. Bi-criteria algorithm for scheduling jobs on cluster platforms. In *Symposium on Parallel Algorithm and Architectures*, pages 125–132, Barcelona, 2004.

[122] Lionel Eyraud-Dubois, Arnaud Legrand, Martin Quinson, and Frédéric Vivien. A first step towards automatically building network representations. In *Proceedings of Euro-Par 2007*, volume 4641 of *LNCS*, pages 160–169, 2007.

[123] Lionel Eyraud-Dubois, Gregory Mounie, and Denis Trystram. Analysis of scheduling algorithms with reservations. In *Proc. 21th Intl. Parallel and Distributed Processing Symposium (IPDPS)*, pages 1–8, Long Beach, California, USA, 2007. IEEE.

[124] Lionel Eyraud-Dubois and Martin Quinson. Assessing the quality of automatically built network representations. In *CCGRID '07: Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, 2007.

[125] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. Computing without communicating: Ring exploration by asynchronous oblivious robots. In *Proceedings of the 11th International Conference on Principles of Distributed Systems (OPODIS)*, volume 4878 of *LNCS*, pages 415–428, 2007.

[126] Pierre Fraigniaud, Cyril Gavoille, David Ilcinkas, and Andrzej Pelc. Distributed computing with advice: Information sensitivity of graph coloring. In *Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4596 of *LNCS*, pages 231–242, 2007.

[127] Pierre Fraigniaud and David Ilcinkas. Digraphs exploration with little memory. In *Proceedings of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2996 of *LNCS*, pages 246–257, 2004.

[128] Pierre Fraigniaud, David Ilcinkas, Guy Peer, Andrzej Pelc, and David Peleg. Graph exploration by a finite automaton. In *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 3153 of *LNCS*, pages 451–462, 2004.

[129] Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. Oracle size: a new measure of difficulty for communication tasks. In *Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 179–187, 2006.

[130] Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. Tree exploration with an oracle. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 4162 of *LNCS*, pages 24–37, 2006.

[131] Pierre Fraigniaud, David Ilcinkas, Sergio Rajsbaum, and Sébastien Tixeuil. Space lower bounds for graph exploration via reduced automata. In *Proceedings of the 12th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, volume 3499 of *LNCS*, pages 140–154, 2005.

[132] David Ilcinkas. Setting port numbers for fast graph exploration. In *Proceedings of the 13th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, volume 4056 of *LNCS*, pages 59–69, 2006.