



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Project-Team Cépage*

*Chercher et Essaimer dans les  
Plates-formes À Grande Échelle*

*Bordeaux - Sud-Ouest*

Theme : Distributed and High Performance Computing

*Activity*  
*R* *eport*

2009



## Table of contents

<b>1. Team</b>	<b>1</b>
<b>2. Overall Objectives</b>	<b>1</b>
2.1. General Objectives	1
2.2. Goal and Context	4
2.2.1. General Context	4
2.2.2. Limitations of Parallel Processing Solutions	4
2.2.3. Limitations of P2P Strategies	5
2.2.4. Targeted Platforms	5
2.3. Highlights	6
<b>3. Scientific Foundations</b>	<b>6</b>
3.1. Modeling Platform Dynamics	6
3.2. Models for Platform Topology and Parameter Estimation	6
3.3. Theoretical Validation	7
3.4. General Framework for Validation	7
3.4.1. Low level modeling of communications	7
3.4.2. Simulation	8
3.4.3. Practical validation and scaling	8
3.5. Efficient Queries and Compact Data Structures	9
3.6. New Services for Scheduling on Large-Scale Platforms	10
<b>4. Software</b>	<b>10</b>
4.1. Molecular Dynamics Simulations	10
4.2. Continuous Integration	12
4.3. Data warehouse	12
4.4. Compact Routing Simulator	13
<b>5. New Results</b>	<b>13</b>
5.1. Efficient Queries and Compact Data Structures	13
5.1.1. Planar Spanner of geometric graphs	13
5.1.2. Compression and Short Data Structures	14
5.1.2.1. Routing with Short Tables	14
5.1.2.2. Succinct Representation of Underlying Topologies	15
5.1.2.3. Local Data Structures and Other Queries	15
5.1.2.4. Distributed Greedy Coloring	16
5.1.2.5. Compression of Data Warehouse	16
5.1.2.6. Maximal Frequent Itemsets.	16
5.1.3. Distributed Algorithms	17
5.1.3.1. Overlay and Small World Networks	17
5.1.3.2. Image Retrieval in a Distributed Network	18
5.1.3.3. Mobile Agent Computing	18
5.1.3.4. Other Results	20
5.2. New Services for Scheduling and Processing on Large Scale Platforms	20
5.2.1. Requests and Task Scheduling on Large Scale Semi-Stable Distributed Platforms	20
5.2.1.1. Divisible Load Scheduling	20
5.2.1.2. Broadcasting and Independent Tasks Scheduling under bounded Multiport Model	21
5.2.2. New Services for Processing on Large Scale Distributed Platforms	22
5.2.2.1. Heterogeneous Dating Service and Distributed Storage Systems	22
5.2.2.2. Building Heterogeneous Clusters	23
5.2.2.3. Complex Queries for Non-Trivial Parallel Algorithms	23
<b>6. Contracts and Grants with Industry</b>	<b>23</b>
6.1. ALCATEL (2009 - 2010)	23

---

6.2.	Yahoo!	24
6.3.	4SH	24
<b>7.</b>	<b>Other Grants and Activities</b> .....	<b>24</b>
7.1.	National Actions	24
7.1.1.	ANR ARA "Masse de données" Alpage (2006–2009)	24
7.1.2.	ANR "programme blanc" Aladdin (2007-2011)	25
7.1.3.	ADT "Aladdin" (2008-2011)	25
7.1.4.	ANR "USS SimGrid" (2008-2011)	25
7.1.5.	ANR programme "defis": project IDEA (2009-2011)	25
7.2.	European Actions	25
7.2.1.	EPSRC travel grant with King's College London and the University of Liverpool	25
7.2.2.	European COST 293 Graal	26
7.2.3.	European Cost 295 DYNAMO	26
7.2.4.	European COST Action ComplexHPC (2009-2012)	26
<b>8.</b>	<b>Dissemination</b> .....	<b>26</b>
8.1.	Community Animation	26
8.1.1.	Editorial Work	26
8.1.1.1.	Steering Committees	26
8.1.1.2.	Program Committees	26
8.1.2.	Organizing Committees	27
8.1.3.	Working Groups	27
8.1.4.	PhD Thesis Refereeing Activities	27
8.1.5.	Invited Talks	28
8.2.	Visits to and from Cepage Members	28
8.2.1.	Visits to Cepage Members	28
8.2.2.	Visits of Cepage Members	28
8.3.	Teaching Activities	28
<b>9.</b>	<b>Bibliography</b> .....	<b>29</b>

*CEPAGE is an INRIA Project-Team joint with University of Bordeaux and CNRS (LaBRI, UMR 5800).*

## 1. Team

### Research Scientist

Olivier Beaumont [ Team Leader, Research Director, INRIA, HdR ]

Lionel Eyraud-Dubois [ Research Associate (CR), INRIA ]

Nicolas Hanusse [ Research Associate (CR), CNRS, HdR ]

David Ilcinkas [ Research Associate (CR), CNRS ]

Ralf Klasing [ Research Associate (CR), CNRS, HdR ]

### Faculty Member

Nicolas Bonichon [ Assistant Professor (MdC), Université Bordeaux I ]

Cyril Gavoille [ Professor (Pr), Université Bordeaux I, IUF, HdR ]

Arnaud Labourel [ Université Bordeaux I, ATER (since September 2009) ]

### External Collaborator

Philippe Duchon [ Professor (Pr), Université Bordeaux I, HdR ]

Adrian Kosowski [ Assistant Professor, University of Gdańsk (since September 2009) ]

### Technical Staff

Ludovic Courtes [ Engineer INRIA SED ]

Frédéric Majorczyk [ Engineer ADERA, Alcatel Grant (since September 2009) ]

### PhD Student

Yousou Dieng [ Université Bordeaux I, MENRT grant ]

Hubert Larchevêque [ CNRS (since September 2007) ]

Hejer Rejeb [ Université Bordeaux I, MENRT grant, (since March 2007) ]

Emilie Diot [ Université Bordeaux I, MENRT grant, (since September 2008) ]

Radu Tofan [ INRIA grant (since October 2007) ]

Ioannis Atsonios [ INRIA grant (since November 2009) ]

Florent Foucaud [ Université Bordeaux I, MENRT grant, (since September 2009) ]

Quentin Godfroy [ University Bordeaux I, ANR Aladdin grant (since October 2009) ]

### Post-Doctoral Fellow

Christopher Thraves-Caro [ Université Bordeaux I, ANR Alpage (left March 2009) ]

Evangelos Bampas [ INRIA grant (since October 2009) ]

Yusik Kim [ INRIA grant (since December 2009) ]

Adrian Kosowski [ Université Bordeaux I (left July 2009) ]

### Administrative Assistant

Alice Rivière [ INRIA ]

## 2. Overall Objectives

### 2.1. General Objectives

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 middle-ware 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 <sup>1</sup>, WCG <sup>2</sup> or XTremWeb) 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 <sup>3</sup>) and computationally intensive applications should be such that they can be arbitrarily split into small pieces of work (such as Folding@home <sup>4</sup> 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, 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.

Firstly, we aim both at building strong foundations for distributed algorithms (graph exploration, black-hole search,...) and distributed data structures (routing, efficient query, compact labeling...) to understand how to explore large scale networks in the context of failures and how to disseminate data so as to answer quickly to specific queries. Secondly, we aim at building simple (based on local estimations without centralized knowledge), realistic models to represent accurately resource performance and to build a realistic view of the topology of the network (based on network coordinates, geometric spanners,  $\delta$ -hyperbolic spaces). Then, we aim at proving that these models are tractable by providing low complexity distributed and randomized approximation algorithms for a set of basic scheduling problems (independent tasks scheduling, broadcasting, data dissemination,...) and associated overlay networks. At last, our goal is to prove the validity of our approach through softwares dedicated to several applications (molecular dynamics simulations, continuous integration)

---

<sup>1</sup><http://boinc.berkeley.edu/>

<sup>2</sup><http://www.worldcommunitygrid.org/>

<sup>3</sup><http://setiathome.berkeley.edu/>

<sup>4</sup><http://folding.stanford.edu/>

as well as more general tools related to the model we propose (AINEM for automatic topology discovery, SimGRID for simulations at large scale).

We will concentrate on the design of new services for computationally 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 4.

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.

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 5.1 and 5.2.
- 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 2.2).

- **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 5.1 and 5.2.
- 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 5.2 and 4.
- 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 5.2).
- To study the computational power of the mobile agent systems under various assumptions on few classical distributed computing problems (exploration, mapping problem, exploration of the network in spite of harmful hosts. The goal is to enlarge the knowledge on

the foundations of mobile agent computing. This will be done by developing new efficient algorithms for mobile agent systems and by proving impossibility results. This will also allow us to compare the different models (David Ilcinkas, Ralf Klasing, Evangelos Bampas) (Section 5.2).

- **Compact and distributed data structures:**
  - To understand how to dynamically adapt compact data structures to changes in network performance and topology (Nicolas Hanusse, Cyril Gavaille) (Section 5.1)
  - To design sophisticated labeling schemes in order to answer complex predicates using local labels only (Nicolas Hanusse, Cyril Gavaille) (Section 5.1)

We will detail in Section 4. how the various expertises in the team will be employed for the considered applications.

We therefore tackle several problems related to two priorities that INRIA identified in its strategic plan (2008-2012): "Modeling, Simulation and Optimization of Complex Dynamic Systems" and "Information, Computation and Communication Everywhere "

## 2.2. Goal and Context

### 2.2.1. 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.

### 2.2.2. 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.

### 2.2.3. *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 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.

### 2.2.4. *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.

## 2.3. Highlights

Year 2009 turns out to be a particular achievements for some members of CEPAGE:

- Cyril Gavaille has been honored as a new member of IUF (Institut Universitaire de France);
- Philippe Duchon has been recruited as a new professeur of Bordeaux I University;
- Ralf Klasing and Nicolas Hanusse presented their HDR defense (Habilitation à diriger les recherches).

# 3. Scientific Foundations

## 3.1. 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.

## 3.2. 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 (and/or 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 the same long-distance 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.

### 3.3. 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  $\int_{SimulationTime} OptThroughput(t)$  or at least  $\min_{SimulationTime} OptThroughput(t)$ .

For instance, Awerbuch and Leighton [57], [58] developed a very nice distributed algorithm for computing multi-flows. The algorithm proposed in [58] 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  $\min_{SimulationTime} OptThroughput(t)$ .

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  $(1 + \epsilon)d_i$ .

Most of Scheduling problems are NP-Complete and unapproximability results exist in on-line settings, especially when resources are heterogeneous. Therefore, we need to rely on simplified communication models (see next section) to prove theoretical results. In this context, resource augmentation techniques are very useful. It consists in identifying a weak parameter (a parameter whose value can be slightly increased without breaking any strong modeling constraint) and then to compare the solution produced by a polynomial time algorithm (with this relaxed constraint) with the optimal solution of the NP-Complete problem (without resource augmentation). This technique is both pertinent in a difficult setting and useful in practice.

### 3.4. General Framework for Validation

#### 3.4.1. 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 [121] are very much based on quasi-exhaustive determination of 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 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.

Under this communication model, it is possible to obtain pathological results. For instance, if we consider a master-slave setting (corresponding to the distribution of independent tasks on a Volunteer Computing platform such as BOINC), the number of slaves connected to the master may be unbounded. In fact, opening simultaneously a large number of TCP connections may lead to a bad sharing of communication resources. Therefore, we propose to add a bound on the number of connexions that can be handled simultaneously by a given node. Estimating this bound is an important issue to obtain realistic communication models.

### 3.4.2. 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. 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 ANR Aladdin Programme Blanc acts as a follow-up, with the members of GANG and Cepage projects). 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 in the framework of USS-Simgrid ANR Arpege project (with MESCAI, ALGORILLE and ASAP Teams). 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).

### 3.4.3. Practical validation and scaling

Finally, we propose several applications that will be described in detail in Section 4. 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 an academic or industrial partner. In all these collaborations, our goal is to prove that the services that we propose in Section 5.2.2 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 4.1). All the applications will first be tested on small scale platforms (using desktop workstations in the laboratory). Then, in order to test their scalability, we propose to implement them either on the GRID 5000 platform or the partner's platform.

### 3.5. 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 5.1.2 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 5.1.3). 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.

- *Mobile Agent Computing.*

Mobile Agent Computing has been proposed as a powerful paradigm to study distributed systems. Our purpose is to study the computational power of the mobile agent systems under various assumptions. Indeed, many models exist but little is known about their computational power. One major parameter describing a mobile agent model is the ability of the agents to interact.

The most natural mobile agent computing problem is the exploration or mapping problem in which one or several mobile agents have to explore or map their environment. The rendezvous problem consists for two agents to meet at some unspecified node of the network. Two other fundamental problems deal with security, which is often the main concern of actual mobile agent systems. The first one consists in exploring the network in spite of harmful hosts that destroy incoming agents. An additional goal in this context is to locate the harmful host(s) to prevent further agent losses. We already mentioned the second problem related to security, which consists for the agents in capturing an intruder.

The goal is to enlarge the knowledge on the foundations of mobile agent computing. This will be done by developing new efficient algorithms for mobile agent systems and by proving impossibility results. This will also allow to compare the different models.

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

### 3.6. New Services for Scheduling on Large-Scale Platforms

As mentioned in Section 2.1, 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 <sup>5</sup>, WCG <sup>6</sup> or XTremWeb 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, by providing, for example, tools for:

- Requests and Task scheduling on large scale platforms;
- New services for processing on large scale platforms.

## 4. Software

### 4.1. 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 5.2.1. This instance involves applications that require the manipulation of large files, which are initially distributed across the platform.

---

<sup>5</sup><http://boinc.berkeley.edu/>

<sup>6</sup><http://www.worldcommunitygrid.org/>

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 5.1.2.3 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 5.2.1.

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. [79] 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 [111].

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. 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 Gavaille 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, Philippe Duchon and Lionel Eyraud-Dubois will work on this part.

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. This prototype supports the dynamicity of the platform: contributing processors can come and leave. The management of configurations' location is now performed using a distributed hash table. This was done by integrating the library Bamboo in the prototype. We still have to solve numerical instability.

## 4.2. 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 <sup>7</sup> initiated by Xavier Hanin (Jayasoft <sup>8</sup>).

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
  - 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.

Ludovic Courtes (delegated to CEPAGE as INRIA SED Engineer) is currently working on a distributed version of the integration server Xooctory. We expect to distribute this new version at the end of 2010.

## 4.3. Data warehouse

We recently focused on two problems:

---

<sup>7</sup><http://xooctory.xoocode.org/>

<sup>8</sup><http://www.jayasoft.fr/index.php>



### 4.3.1. 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 [120] and parallel computation has been proposed but for a static and well-identified platform [80]. This application is typically an interesting example for which the distributed computation and storage can be useful in a heterogeneous and dynamic setting. We just started a collaboration with Sofian Maabout (Assistant Professor in Bordeaux) and 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 5.2.2 to compute and maintain huge datacubes. For the moment, we developed a centralized tool that sums up an whole datacube until dimension 20 and that outperforms usual data cube reduction scheme.

Some work is required to share our tool to a wide public. We plan to do it within the next two years.

## 4.4. Compact Routing Simulator

In the framework of the Alcatel-Lucent Bell collaboration, we are developing a simulator of routing algorithms. This developpement is performed by the engineer F. Majorczyk in Bordeaux site, and in collaboration with INRIA Sophia-Antipolis site.

The main objective is to give a complete experimental study of the Compact Routing Scheme given recently [52] by Abraham, Gavaille, Malkhi, Nisan, Thorup in 2008. This algorithm guarantees, for every weighted  $n$ -node network routing tables of size  $\sqrt{n}$  while the stretch factor is at most 3, i.e., the length of the routes induced by the scheme is never more than three time the optimal length (the distance). The bound on the stretch and in the memory are both optimal. Moreover, the scheme is “Name-Independent”, that is the routing decision at the source router is made on the base of the original name of the destination node. No information can be implicitly encoded in the node names, like coordinates in a grid network. This extra feature is important in practice since in many contexts, node names cannot be renamed according to some global state on the network, in particular whenever the network is growing and dynamic.

This study, if one succeeds, would be the first to report experiments on a name-independent routing scheme. Our simulator implement several graph generators, and the target algorithm currently works on 3000 nodes. We plain to extend the experiments up to 10.000 nodes, and in parallel to give a message efficient distributed algorithm and implementation of this algorithm.

## 5. New Results

### 5.1. Efficient Queries and Compact Data Structures

#### 5.1.1. Planar Spanner of geometric graphs

**Participants:** Nicolas Bonichon, Cyril Gavaille, Nicolas Hanusse, David Ilcinkas.

$\Theta_k$ -graphs are geometric graphs that appear in the context of graph navigation. The shortest-path metric of these graphs is known to approximate the Euclidean complete graph up to a factor depending on the cone number  $k$  and the dimension of the space.

TD-Delaunay graphs, a.k.a. triangular-distance Delaunay triangulations introduced by Chew, have been shown to be plane 2-spanners of the 2D Euclidean complete graph, i.e., the distance in the TD-Delaunay graph between any two points is no more than twice the distance in the plane.

Orthogonal surfaces are geometric objects defined from independent sets of points of the Euclidean space. Orthogonal surfaces are well studied in combinatorics (orders, integer programming) and in algebra. From orthogonal surfaces, geometric graphs, called geodesic embeddings can be built.

We have introduced a specific subgraph of the  $\Theta_6$ -graph defined in the 2D Euclidean space, namely the half- $\Theta_6$ -graph, composed of the even-cone edges of the  $\Theta$ -graph. Our main contribution is to show that these graphs are exactly the TD-Delaunay graphs, and are strongly connected to the geodesic embeddings of orthogonal surfaces of coplanar points in the 3D Euclidean space.

Using these new bridges between these three fields, we establish:

- Every  $\Theta$ -graph is the union of two spanning TD-Delaunay graphs. In particular,  $\Theta_6$ -graphs are 2-spanners of the Euclidean graph. It was not known that  $\Theta_6$ -graphs are  $t$ -spanners for some constant  $t$ , and  $\Theta_7$ -graphs were only known to be  $t$ -spanners for  $t \approx 7.562$ .
- Every plane triangulation is TD-Delaunay realizable, i.e., every combinatorial plane graph for which all its interior faces are triangles is the TD-Delaunay graph of some point set in the plane. Such realizability property does not hold for classical Delaunay triangulations.

In collaboration with Ljubomir Perković, we have also worked on the question of bounded degree planar spanner: what is the minimum  $\Delta$  such that there exists a planar spanner of degree at most  $\Delta$  for any point set? We have proposed an algorithm that computes a 6-spanner of degree at most 6. The best previous known bound on the maximum degree of planar spanner was 14 [113] with a stretch factor of 3.53.

## 5.1.2. Compression and Short Data Structures

### 5.1.2.1. Routing with Short Tables

**Participants:** Cyril Gavoille, Nicolas Hanusse, David Ilcinkas.

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. [97] for a survey of routing techniques).

Some techniques provide routes of length at most  $1 + \epsilon$  times the length of the shortest one (which is the definition of a stretch factor of  $1 + \epsilon$ ) while maintaining a poly-logarithmic number of entries per routing table [50], [75], [124]. 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 [51], [125]. Some experiments have shown that such methods, although universal, work very well in practice, in average, on realistic scale-free or existing topologies [119].

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 [71] 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 compactness  $k$  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.

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.

In 2007, we have improved compact routing scheme for planar networks, and more generally for networks excluding a fixed minor [53]. This later family of networks includes (but is not restricted to) networks embeddable on surfaces of bounded genus and networks of bounded treewidth. The stretch factor of our scheme is constant and the size of each routing table is only polylogarithmic (independently of the degree of the nodes), and the scheme does not require renaming (or a new addressing) of the nodes: it is name-independent. More importantly, the scheme can be constructed efficiently in polynomial time, and complexities do not hide large constants as we may encounter in Minor Graph Theory. This construction has been achieved by the design of new sparse covers for planar graphs, solving a problem open since STOC '93.

In [84], we have shown that routing of outerplanar networks can be done along the shortest paths with  $O(\log n)$ -bit labels, where  $n$  is the number of nodes in the network, extending a result of Fraigniaud *et al.* obtained for trees. The solution actually can be generalized to  $k$ -cellular networks, which is roughly a network that is the union of  $k$  outerplanar networks. It is worth to mention that such a scheme can be constructed in quadratic time.

In 2007, we also gave an invited lecture on compact routing schemes [98] at a workshop on Peer-to-Peer, Routing in Complex Graphs, and Network Coding in Thomson Labs in Paris.

In 2008, we have proposed a minimum stretch compact name-independent routing [52]. This scheme is based on the Compact Routing Simulator we are developing in the Alcatel-Lucent Bell project.

#### 5.1.2.2. 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 embeddable into a bounded page number [99], [101]. Typically, planar topologies with  $n$  nodes (thus embeddable on the plane with no edge crossings) can be 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 [74], [73], [72].

In 2008, we gave a compact encoding scheme of page number  $k$  graphs [100].

#### 5.1.2.3. 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. [56] for distributed NCA algorithms).

Recently, a theory of localized data structures has been developed (initialized by [123]; see [106] 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. These 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 [115]. Dynamic labeling schemes are also available for tree models [117], [118], 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 [49], the XML file structure being a tree. In this framework, distributed label-based data structures avoid the storing of a huge classical index database.

In 2007, we have proved that it is possible to assign with each node of  $n$ -node planar networks a label of  $2 \log n + O(\log \log n)$  bits so that adjacency between two nodes can be retrieved from these labels [104]. Classical representations of planar graphs in the distributed setting were based on the Three Schnyder Trees decomposition, leading to  $3 \log n + O(\log^* n)$  bit labels (FOCS '01). An intriguing question is to know whether  $c \log n$ -bit representation exists for planar graphs with  $c < 2$ .

For trees, we have can solve  $k$ -ancestry and distance- $k$  queries with shorter labels [103], [102]. Previous solutions achieve  $\log n + O(k^2 \log \log n)$ -bit labels [Alstrup-Bille-Rauhe 2005], whereas we have prove that  $\log n + O(k \log \log n)$ -bit labels suffice. For interval graphs, we have given an optimal distance labeling scheme [105], and we proposed a localized and compact data structure for comparability graphs [61].

In [96], [83], [82], we also analyzed the locality of the construction of sparse spanners. In [76], we proposed an efficient first-order model checking using short labels.

Finally, we have started a collaboration with Andrew Twigg (Thomson - Labs) and Bruno Courcelle (LaBRI) about connectivity in semi-dynamic planar networks (see preliminary results here [77] and here [78]). In this model, the must precompute some localized data-structure (given as a label associate with each node) and for a planar graph  $G$ , so that connectivity between any two nodes in  $G \setminus X$  where  $X$  is any subset of nodes or edges, can be determined from the labels of the two nodes and the labels of the nodes (or end-point of edges) of  $X$ . This field looks promising since it capture a kind of dynamicity of the network, and we hope to generalize this model and our results.

#### 5.1.2.4. Distributed Greedy Coloring

**Participants:** Cyril Gavaille, Ralf Klasing, Adrian Kosowski.

Distributed Greedy Coloring is an interesting and intuitive variation of the standard Coloring problem. Given an order among the colors, a coloring is said to be *greedy* if there does not exist a vertex for which its associated color can be replaced by a color of lower position in the fixed order without violating the property that neighboring vertices must receive different colors. In [16], we consider the problems of *Greedy Coloring* and *Largest First Coloring* (a variant of greedy coloring with strengthened constraints) in the Linial model of distributed computation, providing lower and upper bounds and a comparison to the  $(\Delta + 1)$ -Coloring and *Maximal Independent Set* problems, with  $\Delta$  being the maximum vertex degree in  $G$ .

#### 5.1.2.5. Compression of Data Warehouse

**Participants:** Nicolas Hanusse, Radu Tofan.

In [34], we propose a new view selection algorithm. Such algorithm takes as input a fact table and computes a set of views to store in order to speed up queries. The performance of view selection algorithm is usually measured by three criteria: (1) the amount of memory to store the selected views, (2) the query response time and (3) the time complexity of this algorithm. The two first measurements deal with the output of the algorithm. No existing solutions give good trade-offs between amount of memory and queries cost with a small time complexity. We propose in this paper an algorithm guaranteeing a constant approximation factor of queries response time with respect to the optimal solution. Moreover, the time complexity for a  $D$ -dimensional fact table is  $O(D * 2^D)$  corresponding to the fastest known algorithm. We provide an experimental comparison with two other well known algorithms showing that our approach also give good performance in terms of memory. Experiments are done in a centralized setting but our algorithm can easily be adapted in a parallel setting.

We also proposed a new algorithm that allow the administrator or user of a SGBD to choose which part of the data cube to optimize. This problem is called in the litterature *the views selection problem*. The goal consists in chosing the best part of the whole data cube to precompute. Our contribution is to consider that the main constraint is the time to answer to individual queries whereas the memory constraint is usually taken [109].

The next step consists in turning our approach into a parallel and distributed algorithm. We are currently experiencing a parallel algorithm with a theoretical guarantee of performance. More precisaly, given a constant  $f$ , the query time is at most  $f$  times the optimal query (defined whenever the result has already been computed).

#### 5.1.2.6. Maximal Frequent Itemsets.

It turns out that our solution can be adapted to the problem of finding quickly the maximal frequent itemsets within a transaction tables [37]. A transaction consists in a list of items. For a given frequency, we aim at computing the maximal itemsets that are frequent in list of transactions. To our knowledge, there is no parallel algorithm with a guarantee of performance that compute the maximal frequent itemsets. Our solution for the view selection algorithm should be experienced on real instances.

### 5.1.3. Distributed Algorithms

**Participants:** Olivier Beaumont, Philippe Duchon, Cyril Gavoille, Nicolas Hanusse, David Ilcinkas, Ralf Klasing, Radu Tofan.

#### 5.1.3.1. 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 [54] deals with the design of routing tables for stable platforms.

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

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 [81], [114]. 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 [122]. 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 [116], [94], [88].

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 some information can be easily (or locally) accessed like the distance between nodes. More formally, a network is  $f$ -navigable if a greedy routing can be used to get routing paths of  $O(f)$  hops. Recently, many authors aim at finding some networks that be turned into  $\log^{O(1)}$ -navigable network.

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 [89], 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 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.

In 2007, we have two contributions dealing with overlay networks: (1) in [93], there is a formal description of an algorithm turning any network into a  $n^{1/3}$ -navigable network. This article is particularly interesting since it is the first one that considers any input network in the small-worldization process; (2) in [87], [86], we prove that local knowledge is not enough to search quickly for a target node in scale-free networks. Recent studies showed that many real networks are scale-free: the distribution of nodes degree follows a power law on the form  $k^{-\beta}$  with  $\beta \in [2, 3]$ , that is the number of nodes of degree  $k$  is proportional to  $nk^{-\beta}$ . More precisely, we formally prove that in usual scale-free models, it takes  $\Omega(n^{1/2})$  steps to reach the target.

In 2008, we gave a small stretch polylogarithmic network navigability scheme using compact metrics [92].

#### 5.1.3.2. Image Retrieval in a Distributed Network

In [85], we propose a new concept for browsing and searching in large collections of content-based indexed images. Our approach is inspired by greedy routing algorithms used in distributed networks. We define a *navigation graph* whose vertices represent images. The edges of the navigation graph are computed according to a similarity measure between indexed images. The resulting graph can be seen as an ad-hoc network of images in which a greedy routing algorithm can be applied for retrieval purposes. Experiments are done in a centralized setting and could be easily adapted to a distributed setting.

#### 5.1.3.3. Mobile Agent Computing

In [91], we consider networks in which there exists a harmful node, called black hole, destroying any incoming mobile agent. The black hole search problem consists for a team of mobile agents to locate the black hole in the network. We prove that, for this problem, the pebble model is computationally as powerful as the whiteboard model; furthermore the complexity is exactly the same. More precisely, we prove that a team of *two* asynchronous agents, each endowed with a single identical pebble (that can be placed only on nodes, and with no more than one pebble per node) can locate the black hole in an arbitrary network of known topology; this can be done with  $\Theta(n \log n)$  moves, where  $n$  is the number of nodes, even when the links are not FIFO.

In the effort to understand the algorithmic limitations of computing by a swarm of robots, the research has focused on the minimal capabilities that allow a problem to be solved. The weakest of the commonly used models is ASYNCH where the autonomous mobile robots, endowed with visibility sensors (but otherwise unable to communicate), operate in Look-Compute-Move cycles performed asynchronously for each robot. The robots are often assumed (or required to be) oblivious: they keep no memory of observations and computations made in previous cycles. In the paper [90], we consider the setting when the robots are dispersed in an anonymous and unlabeled graph, and they must perform the very basic task of *exploration*: within finite time every node must be visited by at least one robot and the robots must enter a quiescent state. The complexity measure of a solution is the number of robots used to perform the task. We study the case when the graph is an arbitrary tree and establish some unexpected results. We first prove that there are  $n$ -node trees where  $\Omega(n)$  robots are necessary; this holds even if the maximum degree is 4. On the other hand, we show that if the maximum degree is 3, it is possible to explore with only  $O\left(\frac{\log n}{\log \log n}\right)$  robots. The proof of the result



is constructive. Finally, we prove that the size of the team is asymptotically *optimal*: we show that there are trees of degree 3 whose exploration requires  $\Omega(\frac{\log n}{\log \log n})$  robots.

In [29], we consider the problem of *periodic graph exploration* in which a mobile entity with constant memory, an *agent*, has to visit all  $n$  nodes of an arbitrary undirected graph  $G$  in a periodic manner. Graphs are supposed to be anonymous, that is, nodes are unlabeled. However, while visiting a node, the robot has to distinguish between edges incident to it. For each node  $v$  the endpoints of the edges incident to  $v$  are uniquely identified by different integer labels called *port numbers*. We are interested in minimisation of the length of the exploration period. This problem is unsolvable if the local port numbers are set arbitrarily [L. Budach: *Automata and labyrinths*, *Math. Nachrichten* 86(1): 195-282 (1978)]. However, surprisingly small periods can be achieved when assigning carefully the local port numbers. Dobrev et al. [S. Dobrev, J. Jansson, K. Sadakane, W.-K. Sung: *Finding Short Right-Hand-on-the-Wall Walks in Graphs*, *12th Colloquium on Structural Information and Communication Complexity SIROCCO, LNCS 3499, 127-139, 2005*] described an algorithm for assigning port numbers, and an oblivious agent (i.e. agent with no memory) using it, such that the agent explores all graphs of size  $n$  within period  $10n$ . Providing the agent with a constant number of memory bits, the optimal length of the period was proved in [95] to be no more than  $3.75n$  (using a different assignment of the port numbers). In this paper, we improve both these bounds. More precisely, we show a period of length at most  $4\frac{1}{3}n$  for oblivious agents, and a period of length at most  $3.5n$  for agents with constant memory. Moreover, we give the first non-trivial lower bound,  $2.8n$ , on the period length for the oblivious case.

In [28], we consider the problem of exploring an anonymous undirected graph using an oblivious robot. The studied exploration strategies are designed so that the next edge in the robot's walk is chosen using only local information, and so that some local equity (fairness) criterion is satisfied for the adjacent undirected edges. Such strategies can be seen as an attempt to derandomize random walks, and are natural undirected counterparts of the rotor-router model for symmetric directed graphs. The first of the studied strategies, known as Oldest-First (**OF**), always chooses the neighboring edge for which the most time has elapsed since its last traversal. Unlike in the case of symmetric directed graphs, we show that such a strategy in some cases leads to exponential cover time. We then consider another strategy called Least-Used-First (**LUF**) which always uses adjacent edges which have been traversed the smallest number of times. We show that any Least-Used-First exploration covers a graph  $G = (V, E)$  of diameter  $D$  within time  $O(D|E|)$ , and in the long run traverses all edges of  $G$  with the same frequency.

The *rotor-router model*, also called the *Propp machine*, was first considered as a deterministic alternative to the random walk. It is known that the route in an undirected graph  $G = (V, E)$ , where  $|V| = n$  and  $|E| = m$ , adopted by an agent controlled by the rotor-router mechanism forms eventually an Euler tour based on arcs obtained via replacing each edge in  $G$  by two arcs with opposite direction. The process of ushering the agent to an Euler tour is referred to as the *lock-in problem*. In recent work [V. Yanovski, I.A. Wagner, A.M. Bruckstein: *A Distributed Ant Algorithm for Efficiently Patrolling a Network*, *Algorithmica* 37: 165–186 (2003)], Yanovski et al. proved that independently of the initial configuration of the rotor-router mechanism in  $G$  the agent locks-in in time bounded by  $2mD$ , where  $D$  is the diameter of  $G$ . In [21], we examine the dependence of the lock-in time on the initial configuration of the rotor-router mechanism. The case study is performed in the form of a game between a player  $\mathcal{P}$  intending to lock-in the agent in an Euler tour as quickly as possible and its adversary  $\mathcal{A}$  with the counter objective. First, we observe that in certain (easy) cases the lock-in can be achieved in time  $O(m)$ . On the other hand we show that if adversary  $\mathcal{A}$  is solely responsible for the assignment of ports and pointers, the lock-in time  $\Omega(m \cdot D)$  can be enforced in any graph with  $m$  edges and diameter  $D$ . Furthermore, we show that if  $\mathcal{A}$  provides its own port numbering after the initial setup of pointers by  $\mathcal{P}$ , the complexity of the lock-in problem is bounded by  $O(m \cdot \min\{\log m, D\})$ . We also propose a class of graphs in which the lock-in requires time  $\Omega(m \cdot \log m)$ . In the remaining two cases we show that the lock-in requires time  $\Omega(m \cdot D)$  in graphs with the worst-case topology. In addition, however, we present non-trivial classes of graphs with a large diameter in which the lock-in time is  $O(m)$ .

In [22], we consider the model of exploration of an undirected graph  $G$  by a single *agent* which is called the *rotor-router mechanism* or the *Propp machine* (among other names). Let  $\pi_v$  indicate the edge adjacent to a node  $v$  which the agent took on its last exit from  $v$ . The next time when the agent enters node  $v$ , first a ‘‘rotor’’

at node  $v$  advances pointer  $\pi_v$  to the edge  $next(\pi_v)$  which is next after the edge  $\pi_v$  in a fixed cyclic order of the edges adjacent to  $v$ . Then the agent is directed onto edge  $\pi_v$  to move to the next node. It was shown before that after initial  $O(mD)$  steps, the agent periodically follows one established Eulerian cycle, that is, in each period of  $2m$  consecutive steps the agent traverses each edge exactly twice, once in each direction. The parameters  $m$  and  $D$  are the number of edges in  $G$  and the diameter of  $G$ . We investigate robustness of such exploration in presence of faults in the pointers  $\pi_v$  or dynamic changes in the graph. We show that after the exploration establishes an Eulerian cycle,

- (i) if at some step the values of  $k$  pointers  $\pi_v$  are arbitrarily changed, then a new Eulerian cycle is established within  $O(km)$  steps;
- (ii) if at some step  $k$  edges are added to the graph, then a new Eulerian cycle is established within  $O(km)$  steps;
- (iii) if at some step an edge is deleted from the graph, then a new Eulerian cycle is established within  $O(\gamma m)$  steps, where  $\gamma$  is the smallest number of edges in a cycle in graph  $G$  containing the deleted edge.

Our proofs are based on the relation between Eulerian cycles and spanning trees known as the “BEST” Theorem (after de Bruijn, van Aardenne-Ehrenfest, Smith and Tutte).

#### 5.1.3.4. Other Results

Within the wider context of the project, we have published two book chapters on data gathering and energy consumption in wireless networks, respectively [41], [42]. We have also considered the problems of modeling of wireless networks [20], energy efficiency in wireless networks [19], efficient realization of specific classes of permutation networks [17], and broadcasting in radio networks [112].

## 5.2. New Services for Scheduling and Processing on Large Scale Platforms

### 5.2.1. Requests and Task Scheduling on Large Scale Semi-Stable Distributed Platforms

#### 5.2.1.1. Divisible Load Scheduling

**Participants:** Olivier Beaumont, Nicolas Bonichon, Lionel Eyraud-Dubois.

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 [59], [110] and Divisible Load Scheduling (DLS) [70], [69], [55].

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 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.



In [66], we have considered the case task scheduling for parallel multi-frontal methods, what corresponds to map a set of tasks whose dependencies are depicted by a tree. In [65], we have proposed several distributed scheduling algorithms when several applications are to be simultaneously mapped onto an heterogeneous platform.

In [64], we discuss complexity issues for DLS on heterogeneous systems under the bounded multi-port model. To our best knowledge, this is the first attempt to consider DLS under a realistic communication model, where the master node can communicate simultaneously to several slaves, provided that bandwidth constraints are not exceeded. We concentrate on one round distribution schemes, where a given node starts its processing only once all data has been received. Our main contributions are (i) the proof that processors start working immediately after receiving their work (ii) the study of the optimal schedule in the case of 2 processors and (iii) the proof that scheduling divisible load under the bounded multi-port model is NP-complete. This last result strongly differs from divisible load literature and represents the first NP-completeness result when latencies are not taken into account.

Another important and still open issue for Divisible Load Scheduling deals with return communication. Under the classical model, it is assumed that the communication time of the results between the slaves and the master node can be neglected, what strongly limits the application field. In particular, the complexity of the problem with return messages is still opened. This question has been studied in cooperation with Abhay Ghatpande, from Waseda University in [107], [108], [126]. In particular, we have proposed two heuristics for scheduling return messages with different computational costs.

In this context, we have participated in the writing of two book chapters, [43] about different possible modelisations of communications and [40] about steady-state scheduling.

#### 5.2.1.2. *Broadcasting and Independent Tasks Scheduling under bounded Multiport Model*

**Participants:** Olivier Beaumont, Christopher Thraves-Caro, Lionel Eyraud-Dubois, Hejer Rejeb.

We have revisited several classical scheduling problems (Broadcasting, independent tasks scheduling) under more realistic communication models, whose parameters can be instantiated at runtime. We have proved that the use of resource augmentation techniques enables to derive quasi-optimal algorithms even if the underlying scheduling problems are strongly NP-Complete.

In [25], [24], we have considered the problem of allocating a large number of independent, equal-sized tasks to a heterogeneous large scale computing platform. We model the platform using a set of servers (masters) that initially hold (or generate) the tasks to be processed by a set of clients (slaves). All resources have different speeds of communication and computation and we model contentions using the bounded multi-port model. This model corresponds well to modern networking technologies, but for the sake of realism, another parameter needs to be introduced in order to bound the number of simultaneous connections that can be opened at a server node. We prove that unfortunately, this additional parameter makes the problem of maximizing the overall throughput NP-Complete. On the other hand, we also propose a polynomial time algorithm, based on a slight resource augmentation, to solve this problem. More specifically, we prove that, if  $d_j$  denotes the maximal number of connections that can be opened at server node  $S_j$ , then the throughput achieved using this algorithm and  $d_j + 1$  simultaneous connections is at least the same as the optimal one with  $d_j$  simultaneous connections. This algorithm also provides a good approximation for the dual problem of minimizing the maximal number of connections that need to be opened in order to achieve a given throughput, and it can be turned into a standard approximation algorithm (i.e., without resource augmentation).

We have also considered in [39], [26] an extension of the above problem (MTBD) trying to represent the more realistic situation when clients can arrive or leave the system at any time. This extension is called online MTBD. First, we have studied the complexity of the problem, and obtained a negative result saying that no totally online algorithm is able to guarantee a desired approximation factor, even if the algorithm uses resource augmentation. On the other hand, if new connections are allowed each time a client arrives or leaves the system, we propose an algorithm that provides the optimal throughput using resource augmentation and allowing only one new connection per server (each time a client arrives or leaves the system).

In [23], we have considered the problem of broadcasting a large message in a large scale distributed platform, with the same degree-constrained multi-port model. The message must be sent from a source node, with the help of the receiving peers which may forward the message to other peers. In this context, we are not interested in minimizing the makespan for a given message size but rather to maximize the throughput (i.e. the maximum broadcast rate, once steady state has been reached). In this case also, the degree constraint makes the problem of maximizing the overall throughput NP-Complete. On the other hand, we also propose a polynomial time algorithm based on a slight resource augmentation to solve this problem. More specifically, we prove that if  $d_j$  denotes the maximal number of connections that can be opened at node  $C_j$ , then the throughput achieved by this algorithm, using at most  $\max(4, d_j + 2)$  simultaneous connections at node  $C_j$ , is at least the same as the optimal one with  $d_j$  simultaneous connections.

In a closely related context, we have investigated in [27] the influence of bandwidth sharing mechanisms on the performance of scheduling algorithms on large scale distributed platforms. More specifically, we have considered three scheduling problems (file redistribution, independent tasks scheduling and broadcasting) on large scale heterogeneous platforms under the Bounded Multi-port Model. This model can be used when programming at TCP level and is also implemented in modern message passing libraries such as MPICH2. We prove, using the three above mentioned scheduling problems, that this model is tractable and that even very simple distributed algorithms can achieve optimal performance, provided that we can enforce bandwidth sharing policies. Our goal is to assert the necessity of such QoS mechanisms, that are now available in the kernels of modern operating systems, to achieve optimal performance. We prove that implementations of optimal algorithms that do not enforce prescribed bandwidth sharing can fail by a large amount if TCP contention mechanisms only are used. More precisely, for each considered scheduling problem, we have established upper bounds on the performance loss than can be induced by TCP bandwidth sharing mechanisms and we have proved that these upper bounds are tight by exhibiting instances achieving them.

## 5.2.2. New Services for Processing on Large Scale Distributed Platforms

### 5.2.2.1. Heterogeneous Dating Service and Distributed Storage Systems

**Participants:** Olivier Beaumont, Philippe Duchon, Hejer Rejeb.

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 randomly match demands and supplies of some resource of many nodes into couples. In a given round it produces a matching between demands and supplies which is of linear size (compared to the optimal one), even if available resources of individual nodes are very heterogeneous, 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 4..

We also have ongoing work on using this dating service for the maintenance of a randomized overlay network against arbitrary arrivals and departures of nodes, and are trying to remove the requirement for the algorithm to work in a succession of rounds.

In the context of our collaboration with Yahoo!, we have presented in [60] a new algorithm for disk reconfiguration in the context of *Vespa*, a scalable platform for storing, retrieving processing and searching large amounts large amounts of data developed by *Yahoo! Technologies Norway*. The corresponding scheduling problem is closely related to independent related tasks scheduling on heterogeneous platforms, when communication costs are taken into account, and when each task can only be processed on a prescribed set of

processors. We prove how to derive from a linear programming formulation in rational numbers an approximation algorithm whose approximation ratio is close to 1 in the condition of use of *Vespa*.

#### 5.2.2.2. Building Heterogeneous Clusters

**Participants:** Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Lionel Eyraud-Dubois, Hubert Larchevêque.

As already noted in Section 2.1 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 would like to 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 4..

In [63] we present a modeling of this problem called *bin-covering problem with distance constraint* and we propose a distributed approximation algorithm in the case where the elements are in a space of dimension 1. In [62], we describe a generic 2-phases algorithm, based on resource augmentation and whose approximation ratio is  $1/3$ . We also propose a distributed version of this algorithm when the metric space is  $\mathbb{Q}^D$  (for a small value of  $D$ ) and the  $L_\infty$  norm is used to define distances. This algorithm takes  $O((4^D) \log^2 n)$  rounds and  $O((4^D)n \log n)$  messages both in expectation and with high probability, where  $n$  is the total number of hosts.

#### 5.2.2.3. Complex Queries for Non-Trivial Parallel Algorithms

**Participants:** Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Lionel Eyraud-Dubois.

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 efficiency of such requests on the molecular dynamics application and on the continuous integration application described in Section 4.. 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.

A sound approach is to structure them in such a way that they reflect the structure of the application. Peers represent objects of the application so that neighbours in the peer to peer network are objects having similar characteristics from the application's point of view. Such structured peer to peer overlay networks provide a natural support for range and complex queries. We have proposed in [67] to use complex structures such as a Voronoï tessellation, where each peer is associated to a cell in the space. Moreover, since the associated cost to compute and maintain these structures is usually extremely high for dimensions larger than 2, we have proposed to weaken the Voronoï structure to deal with higher dimensional spaces [68].

We are currently adapting the techniques proposed in these papers to the molecular dynamics application in collaboration with Juan Elezgaray from IECB.

## 6. Contracts and Grants with Industry

### 6.1. ALCATEL (2009 - 2010)

**Participants:** Cyril Gavaille, Nicolas Hanusse, David Ilcinkas.

The title of this study is “Dynamic Compact Routing Scheme”. The aim of this projet is to develop new routing schemes achieving better performances than current BGP protocols. The problems faced by the inter-domain routing protocol of the Internet are numerous:

- The underlying network is dynamic: many observations of bad configurations show the instability of BGP;
- BGP does not scale well: the convergence time toward a legal configuration is too long, the size of routing tables is proportional to the number of nodes of network (the network size is multiplied by 1.25 each year);
- The impact of the policies is so important that the many packets can oscillated between two Autonomous Systems.

In this collaboration, we mainly focus on the scalability properties that a new routing protocol should guarantee. The main measures are the size of the local routing tables, and the time (or message complexity) to update or to generate such tables. The design of schemes achieving sub-linear space per routers, say in  $\sqrt{n}$  where  $n$  is the number of AS routers, is the main challenge. The target networks are AS-network like with more than 100,000 nodes. This projet, in colloboration with the MASCOTE INRIA-project in Nice Sophia-Antipolis, makes the use of simulation, developed at both sites.

## 6.2. Yahoo!

**Participants:** Olivier Beaumont, Lionel Eyraud-Dubois, Ralf Klasing, Hejer Rejeb.

Cyril Banino (Yahoo!, Trondheim, Norway) 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 started an informal collaboration with Yahoo! Research, that led to the publication of [60]. We now plan to establish a formal collaboration on document storage in large distributed databases, request scheduling and independent tasks distribution across large distributed platforms.

## 6.3. 4SH

**Participants:** Olivier Beaumont, Nicolas Bonichon, Lionel Eyraud-Dubois.

We started an informal collaboration with Xavier Hanin (4SH) who has developed Xooctory and who has initiated the project Ivy which is now a project of the Apache Software Foundation. This collaboration is supported by INRIA who delegated Ludovic Courtès (INRIA SED Engineer) to work in Cepage for one year from July 2009 on a distributed version of Xooctory.

We are testing and implementing in Xooctory different scheduling algorithms that distributes the build process. We now plan to analyse the graph of dependencies of tasks in the build process to proposed dedicated scheduling algorithms.

# 7. Other Grants and Activities

## 7.1. National Actions

### 7.1.1. ANR ARA "Masse de données" Alpage (2006–2009)

**Participants:** Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Lionel Eyraud-Dubois, Cyril Gavaille, Nicolas Hanusse, David Ilcinkas, Ralf Klasing.

Alpage, lead by Olivier Beaumont, 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

### 7.1.2. 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 projects CEPAGE and GANG.

### 7.1.3. ADT "*Aladdin*" (2008-2011)

**Participants:** Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Lionel Eyraud-Dubois, Cyril Gavoille, Nicolas Hanusse, David Ilcinkas, Ralf Klasing.

The objective of the ADT INRIA Aladdin<sup>9</sup> project is to continue and strengthen the efforts to develop the GRID'5000 platform. Olivier Beaumont is the local coordinator of Grid'5000.

### 7.1.4. ANR "*USS SimGrid*" (2008-2011)

**Participants:** Olivier Beaumont, Nicolas Bonichon, Lionel Eyraud-Dubois.

The objectives of USS SimGrid is to create a simulation framework that will answer (i) the need for simulation scalability arising in the HPC community; (ii) the need for simulation accuracy arising in distributed computing. The Cepage team will be involved in the development of tools to provide realistic model instantiations.

The project involves the following INRIA and CNRS teams: AlGorille, ASAP, Cepage, Graal, MESCAL, SysCom, CC IN2P3.

### 7.1.5. ANR *programme "defis": project IDEA* (2009-2011)

**Participant:** Ralf Klasing.

The goal of this ANR is the study of identifying codes in evolving graphs. Ralf Klasing is the overall leader of the project.

## 7.2. European Actions

### 7.2.1. EPSRC travel grant with King's College London and the University of Liverpool

**Participant:** Ralf Klasing.

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.

<sup>9</sup>Note that this project is absolutely not related to the previous one, even if both projects have the same name!

### 7.2.2. *European COST 293 Graal*

**Participant:** Ralf Klasing.

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>).

### 7.2.3. *European Cost 295 DYNAMO*

**Participants:** Cyril Gavaille, David Ilcinkas, Ralf Klasing.

The COST 295 is an action of the European COST program (European Cooperation in the Field of Scientific and Technical Research) inside of the Telecommunications, 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.lboro.ac.uk/>)

### 7.2.4. *European COST Action ComplexHPC (2009-2012)*

**Participants:** Olivier Beaumont, Nicolas Bonichon, Lionel Eyraud-Dubois.

The goal of ComplexHPC is to coordinate European groups working on the use of heterogeneous and hierarchical systems for HPC as well as the development of collaborative activities among the involved research groups, to tackle the problem at every level (from cores to large-scale environments) and to provide new integrated solutions for large-scale computing for future platforms (see <http://complexhpc.org/index.php>).

## 8. Dissemination

### 8.1. Community Animation

#### 8.1.1. Editorial Work

Ralf Klasing is a member of the Editorial Board of *Theoretical Computer Science, Networks, Parallel Processing Letters, Algorithmic Operations Research, Fundamenta Informaticae*, and *Computing and Informatics*.

##### 8.1.1.1. Steering Committees

- Cyril Gavaille is member of the Steering Committee (as treasurer) of the PODC '10 conference.

##### 8.1.1.2. Program Committees

- OPODIS '09 (Dec., Corsica, France) International Conference on Principles of Distributed Systems (C. Gavaille)
- JDIR '09 (Feb. 2-4, Belfort, UTBM, France) Journées Doctorales en Informatiques et Réseaux (C. Gavaille)
- Conférence SIROCCO 2009 (16th International Colloquium on Structural Information and Communication Complexity), Piran, Slovenia, mai 2009 (D. Ilcinkas)
- Conférence nationale AlgoTel 2008 (11èmes Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications), Carry-Le-Rouet, France, juin 2009 (D. Ilcinkas)
- PODC 2010, Twenty-Ninth Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing Zurich, Switzerland, July 25-28, 2010 (O. Beaumont)
- PASCO 2010, International Workshop on Parallel symbolic Computation, July 21st - July 23rd, 2010, Grenoble, France (O. Beaumont)

- IPDPS 2010 PhD Forum, IEEE International Parallel and Distributed Processing Symposium, Atlanta, EU, 2010 (O. Beaumont)
- SSS 2009, The 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems, Lyon, 2009 (O. Beaumont)
- ISCIS 2009 Northern Cyprus, septembre 2009 (O. Beaumont)
- ISPDPC 2009 8th International Symposium on Parallel and Distributed Computing, Lisbon, Portugal (O. Beaumont)
- HeteroPar 09 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks, August 2009, Delft, The Netherlands (O. Beaumont)
- RenPar 09, Rencontre du Parallélisme, Toulouse, France, 2009 (O. Beaumont)
- IPDPS 2009 PhD Forum, IEEE International Parallel and Distributed Processing Symposium, Rome, Italie, 2009 (O. Beaumont)
- IPDPS 09 IEEE International Parallel and Distributed Processing Symposium, Rome, Italie, 2009 (O. Beaumont)
- SIROCCO 2009 (16th International Colloquium on Structural Information and Communication Complexity), Piran, Slovenie, mai 2009 (R. Klasing)
- MFCS 2009 34th International Symposium on Mathematical Foundations of Computer Science August 24 - 28, 2009 Novy Smokovec, High Tatras, Slovakia (R. Klasing)
- ADHOC-NOW 2009 The 8th International Conference on Ad hoc Networks and Wireless September 16-19 2009 Murcia, Spain (R. Klasing)
- OPODIS 2009 (Dec., Corsica, France) International Conference on Principles of Distributed Systems (R. Klasing)
- ALGOTEL 2009 (June, Carry-le-Rouet, France), Algorithmes de communications dans les réseaux, (N. Hanusse)
- STACS 2010, 27th International Symposium on Theoretical Aspects of Computer Science, March 4-6, 2010 - Nancy, France (R. Klasing)
- ALGOSENSORS 2010, 6th International Workshop on Algorithmic Aspects of Wireless Sensor Networks, July 2010, Bordeaux, France (R. Klasing)
- IWOCA 2010, 21st International Workshop on Combinatorial Algorithms, July 26-28, 2010, London, United Kingdom (R. Klasing)

### **8.1.2. Organizing Commitees**

- Nicolas Bonichon, Philippe Duchon, Cyril Gavoille, Nicolas Hanusse and Ralf Klasing have been involved in the organizing committee of EuroComb 2009, held in September 2009 in Bordeaux.
- Nicolas Bonichon, Lionel Eyraud-Dubois, Cyril Gavoille, and Ralf Klasing are involved in the organization of ICALP 2010 to be held in July 2010 in Bordeaux. (Cyril Gavoille is the Conference Co-Chair, Ralf Klasing is the Workshops Chair.)

### **8.1.3. Working Groups**

- Nicolas Hanusse is responsible for the working group on "Distributed Algorithms" at the LaBRI.
- Ralf Klasing is in charge of the seminary "Distributed Algorithms" at the LaBRI.

### **8.1.4. PhD Thesis Refereeing Activities**

- Olivier Beaumont was external Ph.D. reviewer (rapporteur) and member of the Ph.D. committee of Matthieu Gallet (INRIA Project Team GRAAL, ENS Lyon, France)

- Olivier Beaumont was member of the Ph.D. committee of Gilles Tredan (INRIA Project Team ASAP, IRISA, Rennes, France)
- Olivier Beaumont was external Ph.D. reviewer (rapporteur) and member of the Ph.D. committee of Daouda Traore (INRIA Project Team MESCAL, Grenoble, France).
- Olivier Beaumont was member of the Habilitation thesis committee of François Pellegrini (INRIA Project Team BACCHUS, Bordeaux, France)
- Ralf Klasing was external Ph.D. co-reviewer (rapporteur) and member of the Ph.D. committee of Patricio Reyes (INRIA Project Team MASCOTTE, Université de Nice Sophia Antipolis, France, August 2009)
- Ralf Klasing was member of the Ph.D. committee of Cristiana Gomes (INRIA Project Team MASCOTTE, Université de Nice Sophia Antipolis, France, December 2009)
- Cyril Gavoille was Ph.D. reviewer (rapporteur) of Julien Robert, 12/2009, Ecole Normale Supérieure de Lyon, LIP
- Cyril Gavoille was Ph.D. reviewer (rapporteur) of Ittai Abraham, 11/2009, Jerusalem University, Israel
- Cyril Gavoille was Ph.D. co-reviewer (rapporteur) of Patricio Reyes, 08/2009, Université Nice Sophia-Antipolis (président de jury)

### 8.1.5. Invited Talks

- 04.06.2009: Derandomizing Random Walks in Undirected Graphs Using Locally Fair Exploration Strategies. Dagstuhl Workshop on Dynamic Networks. (R. Klasing)

## 8.2. Visits to and from Cepage Members

### 8.2.1. Visits to Cepage Members

- Nicolas Nisse, INRIA Sophia-Antipolis, March 9-13, 2009 and November 26-December 5, 2009 (D.I.)
- Adrian Kosowski, Gdansk University of Technology, Poland, 16-17/12/2009
- Arnold L. Rosenberg, Colorado State University, USA, 23-27/10/2009
- Ljubomir Perkovic, School of Computing, DePaul University, USA, 29/06-06/07/2009
- Tomasz Radzik, King's College London, UK, 15/06-22/06/2009
- Leszek Gasieniec, University of Liverpool, UK, 11/06-25/06/2009
- Mirosław Korzeniowski, Wrocław University of Technology, Pologne, 13/04-19/04/2009
- Marek Klonowski, Wrocław University of Technology, Pologne, 13/04-19/04/2009
- Leszek Gasieniec, University of Liverpool, UK, 03/04-17/04/2009
- Colin Cooper, King's College London, UK, 20/02-27/02/2009

### 8.2.2. Visits of Cepage Members

- Université Paris 6 (LIP 6), 3-5 March 2009 (Nicolas Hanusse)
- University of Ottawa and Carleton University (Canada), January 5-14, 2009 (David Ilcinkas)
- Université du Québec en Outaouais (Canada), January 14-23, 2009 (David Ilcinkas)
- Weizmann Institute (IL), (aug. 2009 - 1 week)

## 8.3. 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:

- Olivier Beaumont
  - Routing and P2P Networks (last year of engineering school ENSEIRB, 2009)
- Cyril Gavoille
  - Algorithm Analysis (2nd year MASTER "Models and Algorithms" - 2009)
  - Communication and Routing (last year of engineering school ENSEIRB 2009)
  - Real-World Algorithms (2nd year MASTER "Models and Algorithms" - 2009)
  - Graph Algorithms (3rd year Bachelor - 2009)
- Ralf Klasing
  - Communication Algorithms in Networks (2nd year MASTER "Algorithms and Formal Methods" - 2010)

## 9. Bibliography

### Major publications by the team in recent years

- [1] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, Y. ROBERT. *Bandwidth-centric allocation of independent tasks on heterogeneous platforms*, in "Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002", 2002, p. 67–72.
- [2] O. BEAUMONT, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Pipelining broadcasts on heterogeneous platforms*, in "IEEE Transactions on Parallel and Distributed Systems", vol. 16, n<sup>o</sup> 4, 2005, p. 300–313.
- [3] J.-C. BERMOND, J. GALTIER, R. KLASING, N. MORALES, S. PÉRENNES. *Hardness and approximation of Gathering in static radio +networks*, in "Parallel Processing Letters", vol. 16, n<sup>o</sup> 2, June 2006, p. 165–183, <http://dx.doi.org/10.1142/S0129626406002551>.
- [4] P. DUCHON, N. HANUSSE, E. LEBHAR. *Towards small world emergence*, in "Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures", ACM Press New York, NY, USA, 2006, p. 225–232.
- [5] P. DUCHON, N. HANUSSE, N. SAHEB, A. ZEMMARI. *Broadcast in the rendezvous model*, in "Information and Computation", vol. 204, n<sup>o</sup> 5, 2006, p. 697–712.
- [6] L. EYRAUD-DUBOIS, G. MOUNIÉ, D. TRYSTRAM. *Analysis of Scheduling Algorithms with Reservations*, in "IEEE International Parallel and Distributed Processing Symposium, IPDPS 2007, Long Beach, California, USA", IEEE, 2007, p. 1-8, <http://dx.doi.org/10.1109/IPDPS.2007.370304>.
- [7] P. FRAIGNIAUD, C. GAVOILLE, D. ILCINKAS, A. PELC. *Distributed Computing with Advice: Information Sensitivity of Graph Coloring*, in "ICALP", 2007, p. 231-242.
- [8] P. FRAIGNIAUD, C. GAVOILLE, A. KOSOWSKI, E. LEBHAR, Z. LOTKER. *Universal Augmentation Schemes for Network Navigability: Overcoming the  $\sqrt{n}$ -Barrier*, in "19th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)", ACM Press, June 2007, p. 1-7.

- [9] C. GAVOILLE, D. PELEG. *Compact and Localized Distributed Data Structures*, in "Distributed Computing", vol. 16, May 2003, p. 111-120, PODC 20-Year Special Issue.
- [10] J. HROMKOVIČ, R. KLASING, A. PELC, P. RUŽIČKA, W. UNGER. *Dissemination of Information in Communication Networks: Part I. Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*, Springer Monograph, Springer-Verlag, 2005, <http://www.springer.com/france/home?SGWID=7-102-22-33834650-0&changeHeader=true>.

## Year Publications

### Articles in International Peer-Reviewed Journal

- [11] F. BAZZARO, C. GAVOILLE. *Localized and Compact Data-Structure for Comparability Graphs*, in "Discrete Mathematics", vol. 309, n<sup>o</sup> 11, 2009, p. 3465-3484, <http://hal.archives-ouvertes.fr/hal-00368040/en/>.
- [12] C. COOPER, R. KLASING, T. RADZIK. *Locating and repairing faults in a network with mobile agents*, in "Theoretical Computer Science", 2010, <http://hal.archives-ouvertes.fr/hal-00446022/en/>, to appearGB.
- [13] B. COURCELLE, C. GAVOILLE, M. M. KANTÉ. *Compact Labelings For Efficient First-Order Model-Checking*, in "Journal of Combinatorial Optimization", 2009, <http://hal.archives-ouvertes.fr/hal-00342668/en/>, to appear.
- [14] P. FRAIGNIAUD, C. GAVOILLE, D. ILCINKAS, A. PELC. *Distributed computing with advice: information sensitivity of graph coloring*, in "Distributed Computing", vol. 21, n<sup>o</sup> 6, 2009, p. 395-403, <http://hal.archives-ouvertes.fr/hal-00395775/en/CA>.
- [15] P. FRAIGNIAUD, D. ILCINKAS, A. PELC. *Communication algorithms with advice*, in "Journal of Computer and System Sciences", 2009, <http://hal.archives-ouvertes.fr/hal-00412058/en/>, to appearCA.
- [16] C. GAVOILLE, R. KLASING, A. KOSOWSKI, L. KUSZNER, A. NAVARRA. *On the Complexity of Distributed Graph Coloring with Local Minimality Constraints*, in "Networks", vol. 54, n<sup>o</sup> 1, 2009, p. 12–19, <http://hal.archives-ouvertes.fr/hal-00342898/en/ITPL>.
- [17] J. HROMKOVIC, P. KANAREK, R. KLASING, K. LORYS, W. UNGER, H. WAGENER. *On the Size of Permutation Networks and Consequences for Efficient Simulation of Hypercube Algorithms on Bounded-Degree Networks*, in "SIAM Journal on Discrete Mathematics", vol. 23, n<sup>o</sup> 3, 2009, p. 1612–1645, <http://hal.archives-ouvertes.fr/hal-00402764/en/DECH>.
- [18] D. ILCINKAS, N. NISSE, D. SOGUET. *The Cost of Monotonicity in Distributed Graph Searching*, in "Distributed Computing", vol. 22, n<sup>o</sup> 2, 2009, p. 117-127, <http://hal.archives-ouvertes.fr/hal-00412063/en/>.
- [19] R. KLASING, A. KOSOWSKI, A. NAVARRA. *Cost Minimisation in Wireless Networks with a Bounded and Unbounded Number of Interfaces*, in "Networks", vol. 53, n<sup>o</sup> 3, 2009, p. 266–275, <http://hal.archives-ouvertes.fr/hal-00307235/en/ITPL>.
- [20] R. KLASING, Z. LOTKER, A. NAVARRA, S. PÉRENNES. *From Balls and Bins to Points and Vertices*, in "Algorithmic Operations Research", vol. 4, n<sup>o</sup> 2, 2009, p. 133–143, <http://hal.archives-ouvertes.fr/hal-00342915/en/ILIT>.

**International Peer-Reviewed Conference/Proceedings**

- [21] E. BAMPAS, L. GASIEŃIEC, N. HANUSSE, D. ILCINKAS, R. KLASING, A. KOSOWSKI. *Euler Tour Lock-in Problem in the Rotor-Router Model*, in "Proceedings of the 23rd International Symposium on Distributed Computing DISC 2009, Espagne", vol. 5805, Springer Berlin / Heidelberg, 2009, p. 421–433, <http://hal.archives-ouvertes.fr/hal-00402749/en/GRGBPL>.
- [22] E. BAMPAS, L. GASIEŃIEC, R. KLASING, A. KOSOWSKI, T. RADZIK. *Robustness of the rotor-router mechanism*, in "Proceedings of the 13th International Conference on Principles of Distributed Systems OPODIS 2009, France", vol. 5923, Springer Berlin / Heidelberg, 2009, p. 345–358, <http://hal.archives-ouvertes.fr/hal-00417639/en/GRGBPL>.
- [23] O. BEAUMONT, L. EYRAUD-DUBOIS, S. KUMAR. *Broadcasting on Large Scale Heterogeneous Platforms under the Bounded Multi-Port Model*, in "24th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2010), États-Unis d'Amérique Atlanta", 2010, <http://hal.inria.fr/inria-00444586/en/>.
- [24] O. BEAUMONT, L. EYRAUD-DUBOIS, H. REJEB, C. THRAVES. *Allocation of Clients to Multiple Servers on Large Scale Heterogeneous Platforms*, in "The Fifteenth International Conference on Parallel and Distributed Systems (ICPADS'09), Chine Shenzhen", 2009, <http://hal.inria.fr/inria-00413437/en/>.
- [25] O. BEAUMONT, L. EYRAUD-DUBOIS, H. REJEB, C. THRAVES. *Online Allocation of Splittable Clients to Multiple Servers on Large Scale Heterogeneous Platforms*, in "AlgoTel, France Carry-Le-Rouet", A. CHAINTREAU, C. MAGNIEN (editors), 2009, <http://hal.inria.fr/inria-00384569/en/>.
- [26] O. BEAUMONT, L. EYRAUD-DUBOIS, H. REJEB, C. THRAVES. *On-line Allocation of Clients to Multiple Servers on Large Scale Heterogeneous Systems*, in "PDP 2010 - The 18th Euromicro International Conference on Parallel, Distributed and Network-Based Computing, Italie Pisa", 2010, <http://hal.inria.fr/inria-00444584/en/>.
- [27] O. BEAUMONT, H. REJEB. *On the Importance of Bandwidth Control Mechanisms for Scheduling on Large Scale Heterogeneous Platforms*, in "24th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2010), États-Unis d'Amérique Atlanta", 2010, <http://hal.inria.fr/inria-00444585/en/>.
- [28] C. COOPER, D. ILCINKAS, R. KLASING, A. KOSOWSKI. *Derandomizing Random Walks in Undirected Graphs Using Locally Fair Exploration Strategies*, in "Proceedings of the 36th International Colloquium on Automata, Languages and Programming ICALP 2009, Grèce", vol. 5556, Springer Berlin / Heidelberg, 2009, p. 411–422, <http://hal.archives-ouvertes.fr/hal-00374071/en/GBPL>.
- [29] J. CZYZOWICZ, S. DOBREV, L. GASIEŃIEC, D. ILCINKAS, J. JANSSON, R. KLASING, I. LIGNOS, R. MARTIN, K. SADAKANE, W.-K. SUNG. *More Efficient Periodic Traversal in Anonymous Undirected Graphs*, in "Proceedings of the 16th Colloquium on Structural Information and Communication Complexity SIROCCO 2009, Slovénie", vol. 5869, Springer Berlin / Heidelberg, 2009, p. 174–188, <http://hal.archives-ouvertes.fr/hal-00371489/en/GBSKJPCASG>.
- [30] B. DERBEL, C. GAVOILLE, D. PELEG, L. VIENNOT. *Local Computation of Nearly Additive Spanners*, in "23rd International Symposium on Distributed Computing (DISC) DISC, Espagne Elche/Elx", vol. 5805, Springer, 2009, p. 176–190, <http://hal.archives-ouvertes.fr/hal-00408725/en/IL>.

- [31] Y. DIENG, C. GAVOILLE. *On the Tree-Width of Planar Graphs*, in "European Conference on Combinatorics, Graph Theory and Applications (EuroComb) EuroComb 2009, Bordeaux, France, France Bordeaux", vol. 34C, Elsevier, 2009, p. 593-596, <http://hal.archives-ouvertes.fr/hal-00408481/en/>, Electronic Notes in Discrete Mathematics.
- [32] E. DIOT, C. GAVOILLE. *On the Path Separability of Planar Graphs*, in "EuroComb 2009 EuroComb 2009, Bordeaux, France, France Bordeaux", vol. 34C, 2009, p. 549-552, <http://hal.archives-ouvertes.fr/hal-00408228/en/>.
- [33] C. GAVOILLE, A. KOSOWSKI, M. MARKIEWICZ. *What Can be Observed Locally? Round-based Models for Quantum Distributed Computing*, in "23rd International Symposium on Distributed Computing (DISC) DISC, Espagne Elche/Elx", vol. 5805, Springer, 2009, p. 243-257, <http://hal.archives-ouvertes.fr/hal-00408727/en/>.
- [34] N. HANUSSE, S. MAABOUT, R. TOFAN. *A view selection algorithm with performance guarantee*, in "International Conference on Extending Database Technology International Conference on Extending Database Technology, Russie St Petersburg", 2009, électronique, <http://hal.archives-ouvertes.fr/hal-00341475/en/>.

### National Peer-Reviewed Conference/Proceedings

- [35] C. GAVOILLE, Y. DIENG. *Routage dans les réseaux cellulaires*, in "10ièmes Journées Doctorales en Informatique et Réseaux 10ièmes Journées Doctorales en Informatique et Réseaux, Belfort, UTMB, France, février 2009, France Belfort", 2009, p. 7-12, <http://hal.archives-ouvertes.fr/hal-00366615/en/>.
- [36] C. GAVOILLE. *Spanner et routage compact : similarités et différences*, in "AlgoTel, France Carry-Le-Rouet", A. CHAINTREAU, C. MAGNIEN (editors), 2009, <http://hal.inria.fr/inria-00383276/en/>.
- [37] N. HANUSSE, S. MAABOUT, R. TOFAN. *Algorithme distribué pour l'extraction des fréquents maximaux*, in "Algotel, France Carry-Le-Rouet", A. CHAINTREAU, C. MAGNIEN (editors), 2009, <http://hal.inria.fr/inria-00385104/en/>.
- [38] N. HANUSSE, S. MAABOUT, R. TOFAN. *Algorithmes pour la sélection de vues à matérialiser avec garantie de performance*, in "Actes de la conférence EDA 2009 5ème journées francophones sur les entrepôts de données et l'analyse en ligne (EDA'09), France Montpellier", 2009, p. 1-15, <http://hal.archives-ouvertes.fr/hal-00375595/en/>.

### Workshops without Proceedings

- [39] O. BEAUMONT, L. EYRAUD-DUBOIS, H. REJEB, C. THRAVES. *Online and offline Independent Tasks Allocation Schemes on Large Scale Master-Worker Heterogeneous Platforms.*, in "1st Workshop on New Challenges in Distributed Systems, Chili Valparaiso", 2009, <http://hal.inria.fr/inria-00413434/en/>.

### Scientific Books (or Scientific Book chapters)

- [40] O. BEAUMONT, L. MARCHAL. *Steady-State Scheduling*, in "Introduction To Scheduling", Y. ROBERT, F. VIVIEN (editors), Taylor and Francis, 2009, <http://hal.inria.fr/inria-00344157/en/>.
- [41] V. BONIFACI, R. KLASING, P. KORTEWEG, L. STOUGIE, A. MARCHETTI-SPACCAMELA. *Data Gathering in Wireless Networks*, in "Graphs and Algorithms in Communication Networks", A. KOSTER, X. MUNOZ (editors), Springer-Verlag, 2009, p. 357-378, <http://hal.archives-ouvertes.fr/hal-00342958/en/>.

- [42] I. CARAGIANNIS, M. FLAMMINI, C. KAKLAMANIS, R. KLASING, A. NAVARRA. *Energy Consumption Minimization in Ad Hoc Wireless and Multi-Interface Networks*, in "Graphs and Algorithms in Communication Networks", A. KOSTER, X. MUNOZ (editors), Springer-Verlag, 2009, <http://hal.archives-ouvertes.fr/hal-00342955/en/>, to appearIT.
- [43] L. EYRAUD-DUBOIS, A. LEGRAND. *The impact of platform models on scheduling techniques*, in "Introduction to Scheduling", Y. ROBERT, F. VIVIEN (editors), Taylor and Francis, 2009, <http://hal.inria.fr/inria-00344035/en/>.

### Research Reports

- [44] O. BEAUMONT, L. EYRAUD-DUBOIS, H. REJEB, C. THRAVES. *Extended Version: Online Allocation of Splittable Clients to Multiple Servers on Large Scale Heterogeneous Platforms*, INRIA, 2009, <http://hal.inria.fr/inria-00384475/en/>, Rapport de recherche.
- [45] H. REJEB, O. BEAUMONT. *On the Importance of Bandwidth Control Mechanisms for Scheduling on Large Scale Heterogeneous Platforms*, INRIA, 2009, <http://hal.inria.fr/inria-00433124/en/>, RR-7105, Rapport de recherche.

### Other Publications

- [46] J. CZYZOWICZ, D. ILCINKAS, A. LABOUREL, A. PELC. *Asynchronous deterministic rendezvous in bounded terrains*, 2009, <http://hal.archives-ouvertes.fr/hal-00442196/en/>, SubmittedCA.
- [47] J. CZYZOWICZ, D. ILCINKAS, A. LABOUREL, A. PELC. *Optimal Exploration of Terrains with Obstacles*, 2009, <http://hal.archives-ouvertes.fr/hal-00442209/en/>, SubmittedCA.
- [48] C. GAVOILLE, A. KOSOWSKI, M. MARKIEWICZ. *What Can be Observed Locally? Round-based Models for Quantum Distributed Computing*, 2009, <http://hal.archives-ouvertes.fr/hal-00393801/en/>, SubmittedPL.

### References in notes

- [49] S. ABITEBOUL, S. ALSTRUP, H. KAPLAN, T. MILO, T. RAUHE. *Compact Labeling Schemes for Ancestor Queries*, in "SIAM Journal on Computing", 2005.
- [50] I. ABRAHAM, C. GAVOILLE, A. V. GOLDBERG, D. MALKHI. *Routing in Networks with Low Doubling Dimension*, n<sup>o</sup> MSR-TR-2005-175, Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052 - <http://www.research.microsoft.com>, December 2005, Technical report.
- [51] I. ABRAHAM, C. GAVOILLE, D. MALKHI, N. NISAN, M. THORUP. *Compact Name-Independent Routing with Minimum Stretch*, in "16th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)", ACM Press, July 2004, p. 20-24.
- [52] I. ABRAHAM, C. GAVOILLE, D. MALKHI, N. NISAN, M. THORUP. *Compact Name-Independent Routing with Minimum Stretch*, in "ACM Transactions on Algorithms", vol. 3, n<sup>o</sup> 4, 2008, Article 37, <http://dx.doi.org/10.1145/1367064.1367077>.
- [53] I. ABRAHAM, C. GAVOILLE, D. MALKHI, U. WIEDER. *Strong-Diameter Decompositions of Minor Free Graphs*, in "19th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)", ACM Press, June 2007, p. 16-24.

- [54] I. ABRAHAM, D. MALKHI, O. DOBZINSKI. *LAND: stretch (1 + epsilon) locality-aware networks for DHTs.*, in "Symposium of Discrete Algorithms (SODA)", 2004, p. 550-559, <http://doi.acm.org/10.1145/982873>.
- [55] M. ADLER, Y. GONG, A. L. ROSENBERG. *Optimal sharing of bags of tasks in heterogeneous clusters*, in "15th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA'03)", ACM Press, 2003, p. 1–10.
- [56] S. ALSTRUP, C. GAVOILLE, H. KAPLAN, T. RAUHE. *Nearest Common Ancestors: A Survey and a New Algorithm for a Distributed Environment*, in "Theory of Computing Systems", vol. 37, 2004, p. 441-456, <http://dx.doi.org/10.1007/s00224-004-1155-5>.
- [57] B. AWERBUCH, F. T. LEIGHTON. *A Simple Local-Control Approximation Algorithm for Multicommodity Flow*, in "IEEE Symposium on Foundations of Computer Science", 1993, p. 459-468.
- [58] B. AWERBUCH, T. 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", 1994, p. 487–496.
- [59] C. BANINO, O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, Y. ROBERT. *Scheduling strategies for master-slave tasking on heterogeneous processor platforms*, in "IEEE Trans. Parallel Distributed Systems", vol. 15, n<sup>o</sup> 4, 2004, p. 319-330.
- [60] C. BANINO-ROKKONES, O. BEAUMONT, H. REJEB. *Scheduling Techniques for Effective System Reconfiguration in Distributed Storage Systems*, in "IEEE Intl Conference on Parallel and Distributed Systems Conferences, Australie Melbourne", 2008, <http://hal.inria.fr/inria-00336242/en/NO>.
- [61] F. BAZZARO, C. GAVOILLE. *Localized and Compact Data-Structure for Comparability Graphs*, in "Discrete Mathematics", 2008, <http://dx.doi.org/10.1016/j.disc.2007.12.091>.
- [62] O. BEAUMONT, N. BONICHON, P. DUCHON, L. EYRAUD-DUBOIS, H. LARCHEVÊQUE. *A Distributed Algorithm for Resource Clustering in Large Scale Platforms*, in "OPODIS 2008, Égypte Louxor", 2008, <http://hal.inria.fr/inria-00369417/fr/>.
- [63] O. BEAUMONT, N. BONICHON, P. DUCHON, H. LARCHEVÊQUE. *Distributed Approximation Algorithm for Resource Clustering*, in "SIROCCO 2008, Suisse", 2008, p. 61–73, <http://hal.archives-ouvertes.fr/hal-00323068/en/>.
- [64] O. BEAUMONT, N. BONICHON, L. EYRAUD-DUBOIS. *Scheduling divisible workloads on heterogeneous platforms under bounded multi-port model*, in "Heterogeneity in Computing Workshop, in IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008., États-Unis d'Amérique Miami, FL", 2008, <http://hal.inria.fr/inria-00336189/en/>.
- [65] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Centralized versus Distributed Schedulers for Bag-of-Tasks Applications*, in "IEEE Transactions on Parallel and Distributed Systems", 2008, <http://hal.inria.fr/inria-00336197/en/US>.
- [66] O. BEAUMONT, A. GUERMOUCHE. *Task Scheduling for Parallel Multifrontal Methods*, in "Euro-Par 2007 Parallel Processing", Lecture Notes in Computer Science, 2007, <http://www.labri.fr/publications/paradis/2007/BG07>.



- [67] O. BEAUMONT, A. KERMARREC, L. MARCHAL, E. RIVIÈRE. *VoroNet: A scalable object network based on Voronoi tessellations*, in "International Parallel and Distributed Processing Symposium IPDPS", 2007.
- [68] O. BEAUMONT, A. KERMARREC, E. RIVIÈRE. *Peer to peer multidimensional overlays: Approximating complex structures*, in "OPODIS", 2007.
- [69] O. BEAUMONT, L. MARCHAL, Y. 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.
- [70] V. BHARADWAJ, D. GHOSE, V. MANI, T. ROBERTAZZI. *Scheduling Divisible Loads in Parallel and Distributed Systems*, IEEE Computer Society Press, 1996.
- [71] H. L. BODLAENDER, J. V. LEEUWEN, R. B. TAN, D. M. THILIKOS. *On Interval Routing Schemes and Treewidth*, in "Information and Computation", vol. 139, November 1997, p. 92-109.
- [72] N. BONICHON, C. GAVOILLE, N. HANUSSE. *An Information-Theoretic Upper Bound of Planar Graphs Using Triangulation*, in "20th Annual Symposium on Theoretical Aspects of Computer Science (STACS)", vol. 2607 of Lecture Notes in Computer Science, Springer, February 2003, p. 499-510.
- [73] N. BONICHON, C. GAVOILLE, N. HANUSSE. *Canonical Decomposition of Outerplanar Maps and Application to Enumeration, Coding and Generation*, in "29th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)", vol. 2880 of Lecture Notes in Computer Science, Springer-Verlag, June 2003, p. 81-92.
- [74] N. BONICHON, C. GAVOILLE, N. HANUSSE, D. POULALHON, G. SCHAEFFER. *Planar Graphs, via Well-Orderly Maps and Trees*, in "30th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)", vol. 3353 of Lecture Notes in Computer Science, Springer, June 2004, <http://dx.doi.org/10.1007/b104584>, 270-284.
- [75] T.-H. H. CHAN, A. GUPTA, B. M. MAGGS, S. ZHOU. *On Hierarchical Routing in Doubling Metrics*, in "16th Symposium on Discrete Algorithms (SODA)", ACM-SIAM, January 2005, p. 762-771.
- [76] B. COURCELLE, C. GAVOILLE, M. M. KANTÉ. *Efficient First-Order Model-Checking using Short Labels*, in "2nd International Frontiers of Algorithmics Workshop (FAW)", vol. 5059 of Lecture Notes in Computer Science, Springer, 2008, p. 159-170.
- [77] B. COURCELLE, C. GAVOILLE, M. KANTÉ, D. A. TWIGG. *Forbidden-Set Labeling on Graphs*, in "2nd Workshop on Locality Preserving Distributed Computing Methods (LOCALITY)", August 2007, Co-located with PODC 2007.
- [78] B. COURCELLE, C. GAVOILLE, M. M. KANTÉ, D. A. TWIGG. *Connectivity check in 3-connected planar graphs with obstacles*, in "International Conference on Topological & Geometric Graph Theory", vol. 31, Electronic Notes in Discrete Mathematics, 2008, p. 151-155, <http://dx.doi.org/10.1016/j.endm.2008.06.030>.
- [79] E. DARVE, A. POHORILLE. *Calculating free energies using average force*, in "Journal of Chemical Physics", vol. 115, 2001, p. 9169-9183.

- [80] F. K. H. A. DEHNE, T. EAVIS, SUSANNE E. HAMBRUSCH, A. RAU-CHAPLIN. *Parallelizing the Data Cube*, in "Distributed and Parallel Databases", vol. 11, n<sup>o</sup> 2, 2002, p. 181-201.
- [81] A. J. DEMERS, D. H. GREENE, C. HAUSER, W. IRISH, J. LARSON, S. SHENKER, H. E. STURGIS, D. C. SWINEHART, D. B. TERRY. *Epidemic Algorithms for Replicated Database Maintenance.*, in "Operating Systems Review", vol. 22, n<sup>o</sup> 1, 1988, p. 8-32.
- [82] B. DERBEL, C. GAVOILLE. *Fast Deterministic Distributed Algorithms for Sparse Spanners*, in "Theoretical Computer Science", vol. 399, n<sup>o</sup> 1-2, 2008, p. 83-100, <http://dx.doi.org/10.1016/j.tcs.2008.02.019>.
- [83] B. DERBEL, C. GAVOILLE, D. PELEG, L. VIENNOT. *Construction Locale de Sous-Graphes Couvrants Peu Denses*, in "10 èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)", 2008, p. 105-108.
- [84] Y. DIENG, C. GAVOILLE. *Routage dans les graphes cellulaires*, in "9 èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)", May 2007, p. 91-94.
- [85] A. DON, N. HANUSSE. *Content-based image retrieval using greedy routing*, in "Electronic Imaging - Multimedia Content Access Track, États-Unis d'Amérique", vol. 6820, Theo Gevers, Ramesh C. Jain and Simone Santini, 2008, p. 6820I1-6820I11, <http://hal.archives-ouvertes.fr/hal-00307538/en/>.
- [86] P. DUCHON, N. EGGEMANN, N. HANUSSE. *Non-Navigability of random scale-free graphs*, in "OPODIS - International Conference Of Principles of Distributed Systems", 2007.
- [87] P. DUCHON, N. EGGEMANN, N. HANUSSE. *Non-searchability of random scale-free graphs*, in "PODC - Principles Of Distributed Computing", 2007, p. 380-381.
- [88] P. DUCHON, N. HANUSSE, E. LEBHAR, N. SCHABANEL. *Could any graph be turned into a small world ?*, in "International Symposium on Distributed Computing (DISC)", P. FRAIGNIAUD (editor), Lecture Notes in Computer Science, vol. 3724, Springer Verlag, 2005, p. 511-513, <http://www.labri.fr/publications/combalgo/2005/DHLS05a>.
- [89] P. DUCHON, N. HANUSSE, E. LEBHAR, N. SCHABANEL. *Towards Small World Emergence*, in "SPAA2006 - 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures, PO box 11405, NY - 10286-6626", U. VISHKIN (editor), ACM Press, ACM SIGACT - ACM SIGARCH, July 2006, p. 225-232, <http://www.labri.fr/publications/combalgo/2006/DHLS06>.
- [90] P. FLOCCHINI, D. ILCINKAS, A. PELC, N. SANTORO. *Remembering without Memory: Tree Exploration by Asynchronous Oblivious Robots*, in "SIROCCO 2008, Suisse Villars-sur-Ollon", A. A. SHVARTSMAN, P. FELBER (editors), vol. 5058, Springer Berlin / Heidelberg, 2008, p. 33-47, <http://hal.archives-ouvertes.fr/hal-00341465/en/CA>.
- [91] P. FLOCCHINI, D. ILCINKAS, N. SANTORO. *Ping Pong in Dangerous Graphs: Optimal Black Hole Search with Pure Tokens*, in "DISC 2008, France", G. TAUBENFELD (editor), vol. 5218, Springer Berlin / Heidelberg, 2008, p. 227-241, <http://hal.archives-ouvertes.fr/hal-00341523/en/CA>.
- [92] P. FRAIGNIAUD, C. GAVOILLE. *Polylogarithmic Network Navigability Using Compact Metrics with Small Stretch*, in "20th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)", ACM Press, 2008, p. 62-69.



- [93] P. FRAIGNIAUD, C. GAVOILLE, A. KOSOWSKI, E. LEBHAR, Z. LOTKER. *Universal augmentation schemes for network navigability: Overcoming the  $\sqrt{n}$ -barrier*, in "19th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)", 2007, p. 1-7.
- [94] P. FRAIGNIAUD, C. GAVOILLE, C. PAUL. *Eclecticism Shrinks Even Small Worlds*, in "23rd Annual ACM Symposium on Principles of Distributed Computing (PODC)", ACM Press, July 2004, p. 169-178.
- [95] L. GASIENIEC, R. KLASING, R. MARTIN, A. NAVARRA, X. ZHANG. *Fast Periodic Graph Exploration with Constant Memory*, in "Journal of Computer and System Sciences", vol. 74, 2008, p. 808–822, <http://hal.archives-ouvertes.fr/hal-00306995/en/GBIT>.
- [96] C. GAVOILLE, B. DERBEL, D. PELEG, L. VIENNOT. *On the Locality of Distributed Sparse Spanner Construction*, in "27th Annual ACM Symposium on Principles of Distributed Computing (PODC), Canada", ACM Press, 2008, p. 273-282, <http://hal.archives-ouvertes.fr/hal-00343001/en/IL>.
- [97] C. GAVOILLE. *Routing in Distributed Networks: Overview and Open Problems*, in "ACM SIGACT News - Distributed Computing Column", vol. 32, n<sup>o</sup> 1, March 2001, p. 36-52.
- [98] C. GAVOILLE. *An Overview on Compact Routing*, in "Workshop on Peer-to-Peer, Routing in Complex Graphs, and Network Coding", March 2007.
- [99] C. GAVOILLE, N. HANUSSE. *On Compact Encoding of Pagenumber  $k$  Graphs*, in "Discrete Mathematics & Theoretical Computer Science", 2005, To appear.
- [100] C. GAVOILLE, N. HANUSSE. *On Compact Encoding of Pagenumber  $k$  Graphs*, in "Discrete Mathematics & Theoretical Computer Science", 2008, To appear.
- [101] C. GAVOILLE, N. HANUSSE. *Compact Routing Tables for Graphs of Bounded Genus*, in "26th International Colloquium on Automata, Languages and Programming (ICALP)", J. WIEDERMANN, P. VAN EMDE BOAS, M. NIELSEN (editors), vol. 1644 of Lecture Notes in Computer Science, Springer, July 1999, p. 351-360.
- [102] C. GAVOILLE, A. LABOUREL. *Brief Annoucement: On Local Representation of Distances in Trees*, in "26th Annual ACM Symposium on Principles of Distributed Computing (PODC)", ACM Press, August 2007, p. 246-247, <http://portal.acm.org/citation.cfm?id=1281169>.
- [103] C. GAVOILLE, A. LABOUREL. *Distributed Relationship Schemes for Trees*, in "18th Annual International Symposium on Algorithms and Computation (ISAAC)", vol. 4835 of Lecture Notes in Computer Science, Springer, December 2007, p. 728-738.
- [104] C. GAVOILLE, A. LABOUREL. *Shorter Implicit Representation for Planar Graphs and Bounded Treewidth Graphs*, in "15th Annual European Symposium on Algorithms (ESA)", L. ARGE, E. WELZL (editors), vol. 4698 of Lecture Notes in Computer Science, Springer, October 2007, p. 582-593.
- [105] C. GAVOILLE, C. PAUL. *Optimal Distance Labeling for Interval Graphs and Related Graphs Families*, in "SIAM Journal on Discrete Mathematics", vol. 22, n<sup>o</sup> 3, 2008, p. 1239-1258, <http://dx.doi.org/10.1137/050635006>.

- [106] C. GAVOILLE, D. PELEG. *Compact and Localized Distributed Data Structures*, in "Journal of Distributed Computing", vol. 16, May 2003, p. 111-120, PODC 20-Year Special Issue.
- [107] A. GHATPANDE, H. NAKAZATO, O. BEAUMONT, H. WATANABE. *Analysis of Divisible Load Scheduling with Result Collection on Heterogeneous Systems*, in "IEICE Transactions on Communications", 2008, <http://hal.inria.fr/inria-00336236/en/JP>.
- [108] A. GHATPANDE, H. NAKAZATO, O. BEAUMONT, H. WATANABE. *SPORT: An Algorithm for Divisible Load Scheduling with Result Collection on Heterogeneous Systems*, in "IEICE Transactions on Communications", 2008, <http://hal.inria.fr/inria-00336212/en/JP>.
- [109] N. HANUSSE, D. KAVVADIAS, E. KRANAKIS, D. KRIZANC. *Memoryless search algorithms in a network with faulty advice*, in "Theoretical Computer Science (TCS)", vol. 402, 2008, p. 190-198, <http://hal.archives-ouvertes.fr/hal-00341468/en/GRUS>.
- [110] B. HONG, V. 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.
- [111] J. HÉNIN, C. CHIPOT. *Overcoming free energy barriers using unconstrained molecular dynamics simulations*, in "Journal of Chemical Physics", vol. 121, 2004, p. 2904–2914.
- [112] D. ILCINKAS, D. R. KOWALSKI, A. PELC. *Fast Radio Broadcasting with Advice*, in "SIROCCO 2008, Suisse Villars-sur-Ollon", A. A. SHVARTSMAN, P. FELBER (editors), vol. 5058, Springer Berlin / Heidelberg, 2008, p. 291-305, <http://hal.archives-ouvertes.fr/hal-00341449/en/GBCA>.
- [113] I. A. KANJ, L. PERKOVIĆ. *On Geometric Spanners of Euclidean and Unit Disk Graphs*, in "25th Annual Symposium on Theoretical Aspects of Computer Science (STACS)", vol. hal-00231084, HAL, February 2008, p. 409-420.
- [114] R. M. KARP, C. SCHINDELHAUER, S. SHENKER, B. VÖCKING. *Randomized Rumor Spreading.*, in "FOCS", 2000, p. 565-574.
- [115] M. KATZ, N. A. KATZ, A. KORMAN, D. PELEG. *Labeling schemes for flow and connectivity*, in "SIAM Journal on Computing", vol. 34, n<sup>o</sup> 1, 2004, p. 23-40.
- [116] J. KLEINBERG. *The Small-World Phenomenon: An Algorithmic Perspective*, in "Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)", 2000, p. 163–170.
- [117] A. KORMAN. *General Compact Labeling Schemes for Dynamic Trees*, in "19th International Symposium on Distributed Computing (DISC)", vol. 3724 of Lecture Notes in Computer Science, Springer, September 2005, p. 457-471.
- [118] A. KORMAN, D. PELEG. *Labeling Schemes for Weighted Dynamic Tree*, in "30th International Colloquium on Automata, Languages and Programming (ICALP)", vol. 2719 of Lecture Notes in Computer Science, Springer, July 2003, p. 369-383.

- 
- [119] D. KRIOUKOV, K. FALL, X. YANG. *Compact Routing on Internet-Like Graphs*, in "IEEE INFOCOM", 2004, To appear.
- [120] M. LAPORTE, N. NOVELLI, R. CICHETTI, L. LAKHAL. *Computing Full and Iceberg Datacubes Using Partitions*, in "ISMIS '02: Proceedings of the 13th International Symposium on Foundations of Intelligent Systems, London, UK", Springer-Verlag, 2002, p. 244–254.
- [121] A. LEGRAND, F. MAZOIT, M. QUINSON. *An Application-level network mapper*, n<sup>o</sup> RR-2003-09, LIP, ENS Lyon, France, feb 2003, Research Report.
- [122] S. MILGRAM. *The small world problem*, in "Psychology Today", vol. 61, n<sup>o</sup> 1, 1967.
- [123] D. PELEG. *Informative Labeling Schemes for Graphs*, in "25th International Symposium on Mathematical Foundations of Computer Science (MFCS)", vol. 1893 of Lecture Notes in Computer Science, Springer, August 2000, p. 579-588.
- [124] A. SLIVKINS. *Distance Estimation and Object Location via Rings of Neighbors*, in "24th Annual ACM Symposium on Principles of Distributed Computing (PODC)", ACM Press, July 2005, p. 41-50, Appears earlier as Cornell CIS technical report TR2005-1977.
- [125] M. THORUP, U. ZWICK. *Compact Routing Schemes*, in "13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)", ACM Press, July 2001, p. 1-10.
- [126] H. WATANABE, H. NAKAZATO, A. GHATPANDE, O. BEAUMONT. *Divisible Load Scheduling with Result Collection on Heterogeneous Systems*, in "IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008., États-Unis d'Amérique Miami, FL", 2008, <http://hal.inria.fr/inria-00336186/en/JP>.