INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team Cépage

Chercher et Essaimer dans les Plates-formes À Grande Échelle

Futurs

THEME COM

Activity Report

2007

Table of contents

1. Team

Head of the team
Olivier Beaumont [Assistant Professor (MdC) ENSEIRB, HdR]

Administrative assistant
Laetitia Grimaldi [Secretary (SAR) Inria]

Research scientists
Nicolas Bonichon [Assistant Professor (MdC), Université Bordeaux I]
Lyonel Eyraud-Dubois [Research Associate (CR), Inria (since September 2007)]
Cyril Gavoille [Professor (Pr), Université Bordeaux I, HdR]
Nicolas Hanusse [Research Associate (CR), CNRS]
David Ilcinkas [Research Associate (CR), CNRS (since October 2007)]
Ralf Klasing [Research Associate (CR), CNRS]

External collaborator
Philippe Duchon [Assistant Professor (MdC) ENSEIRB]

PHD Students
Youssou Dieng [MENRT grant, Université Bordeaux I]
Arnaud Labourel [MENRT grant, Université Bordeaux I]
Hubert Larchevêque [BDI Région/CNRS (since September 2007)]
Hejer Rejeb [MENRT grant, Université Bordeaux I (since February 2007)]
Radu Tofan [INRIA grant (since October 2007)]

Post-doctoral fellow
Miroslaw Korzeniowski [Post-Doc Inria (until September 2007)]
Alfredo Navarra [Post-Doc Université Bordeaux I (until April 2007)]

Visitor
Robert Elsässer [INRIA guest professor/University of Paderborn, Germany, from August 31 till October 12]

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 middleware and application deployment times, applications must be long-running and involve large amounts of data; also, only loosely-coupled applications may currently be executed on unstable platforms.

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

The goal of our project is to establish a link between these two directions, by gathering researchers from the distributed algorithms and data structures, parallel and randomized algorithms communities. More precisely, the objective of our project is to extend the application field that can be executed on large scale distributed platforms. Indeed, whereas protocols designed for P2P file exchange are actually distributed, computationally intensive applications executed on large scale platforms (BOINC [1], WCG [2] or XTremWeb) 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 [3]) and computationally intensive applications should be such that they can be arbitrarily split into small pieces of work (such as Folding@home [4] 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.

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

[1] http://boinc.berkeley.edu/
[2] http://www.worldcommunitygrid.org/
[3] http://setiathome.berkeley.edu/
[4] http://folding.stanford.edu/

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).
- **Compact and distributed data structures:**
  - to understand how to dynamically adapt compact data structures to changes in network performance and topology (Nicolas Hanusse, Cyril Gavoille) (Section 5.1)
  - to design sophisticated labeling schemes in order to answer complex predicates using local labels only (Nicolas Hanusse, Cyril Gavoille) (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 the first of the major challenges that INRIA identified in its strategic plan (2003-2007) "Designing and mastering the future network infrastructures and communication services platforms".

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 between peers occur between peers connected in the overlay network, communications between two close nodes (in the physical network) may well involve many wide area messages, and therefore such a constraint is in contradiction with performance optimization. Fortunately, in the case of file sharing applications, only queries are transmitted using the overlay network, and the communication of large files is made at IP level. On the other hand, in the case of more complex communication schemes, such as broadcast or multicast, the communication of large files is done using the overlay network, due to the lack of support, at IP level, for those complex operations. In this case, in order to achieve good results, it is crucial that virtual and physical topologies be as close as possible.

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

1. The International Symposium on Distributed Computing (DISC) is one of the leading international conferences in the area of foundations of distributed computing (together with PODC). It has a long tradition (it goes into its 22nd edition), it awards (together with PODC) the prestigious Edsger W. Dijkstra Prize in Distributed Computing to an outstanding paper on the principles of distributed computing. DISC has about 100 participants every year from more than 20 different countries. CEPAGE has been chosen by the steering committee of DISC to host the 22nd edition of DISC in 2008. The conference will be organized by CEPAGE in Arcachon from 22-24 September 2008.
2. Cyril Gavoille obtained the best paper award at the major conference SPAA 2007 for the paper "Universal Augmentation Schemes for Network Navigability: Overcoming the $\sqrt{n}$-Barrier", co-authored with Pierre Fraigniaud, Adrian Kosowski, Emmanuelle Lebhar and Zvi Lotker. Augmented graphs were introduced for the purpose of analyzing the "six degrees of separation between individuals" observed experimentally by the sociologiest Standley Milgram in the 60's. In augmented graphs, greedy routing is the oblivious routing process in which every intermediate node chooses among all its neighbors (including its long range contact) the one that is closest to the target according to the distance measured in the underlying graph, and forwards to it. Roughly, augmented graphs aim at modeling the structure of social networks, while greedy routing aims at modeling the searching procedure applied in Milgram's experiment. Our objective is to design efficient *universal* augmentation schemes, i.e., augmentation schemes that give to any graph G a collection of probability distributions φ such that greedy routing in (G, φ) is fast.
3. The members of Cepage are involved in the following program committees in 2007 and 2008 (either as PC Chair or PC Member) SPAA07, DISC07, AltoTel07, HeteroPar 07, EuroPar07, IPDPS 07, PMGC07, PMAA 08, RenPar 08, HeteroPar 08, PODC08, DISC08, AlgoTel08, JDIR08, and in the editorial board of the following journals *Networks*, *Parallel Processing Letters*, and *Algorithmic Operations Research*.

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 (or/and receive from) more than one node. Most of the steady state scheduling results have been obtained using this model. On the other hand, some authors propose to model incoming and outgoing communication from a node using fictitious incoming and outgoing links, whose bandwidths are fixed. The main advantage of this model, although it might be slightly less accurate, is that it does not require strong synchronization and that many scheduling problems can be expressed as multi-commodity flow problems, for which decentralized efficient algorithms are known. Another important issue is to model the bandwidth actually allocated to each communication when several communications compete for a WAN link.
- At a higher level, proving good approximation ratios on general graphs may be too difficult, and it has been observed that actual platforms often exhibit a simple structure. For instance, many real life networks satisfy small-world properties, and it has been proved, for instance, that greedy routing protocols on small world networks achieve good performance. It is therefore of interest to prove that logical (given by the interactions between hosts) and physical platforms (given by the network links) exhibit some structure in order to derive efficient algorithms.

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 [53], [54] developed a very nice distributed algorithm for computing multi-flows. The algorithm proposed in [54] 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 $(1+\epsilon)d_i$.

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 [82] 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[5] 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.

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 we have proposed a follow-up, with the members of GANG and Cepage projects to ANR Blanche program). On the other hand, we also need to rely on an efficient simulator in order to test our algorithms. We have not yet chosen the discrete event simulator we will use for simulations. One attractive possibility would be to adapt SimGRID, developed in the Mescal project, to large scale dynamic environments. Indeed, a parallel version of SimGrid, based on activations is currently under development. This version will be able to deal with platforms containing more than 10^5 resources. SimGrid has been developed by Henri Casanova (U.C. San Diego) and Arnaud Legrand during his PhD (under the co-supervision of O. Beaumont).

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.

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

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 5.1.1 show how to pre-process the network so that local computations are performed efficiently. Due to the dynamism of large distributed platforms, some requirements on the coding of local data structures and the udpating mechanism are needed. This last process is done using the maintenance of light virtual networks, so-called *overlay networks* (see Section 5.1.2). In our approach, we focus on:

- *Compression.*

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

  - In order to guarantee that local computations are done efficiently, that is avoiding external memory requests, it may be of interest that the coding of the underlying topology can be stored within *fast memory* space.
  - The dynamism of the network implies many basic message communications to update the knowledge of each node. The smaller the message size is, the better the performance.

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

- *Routing tables.*

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

- *Local computations.*

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

- *Overlay and small world networks.*

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

  In order to meet these requirements, *overlay networks* are normally implemented. These are light virtual networks, i.e., they are sparse and a local change of the physical network will only lead to a small change of the corresponding virtual network. As a result, small address books are sufficient at each node.

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

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

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 [6], WCG [7] 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.

- *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.

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 5.1.1.3 will be used to dynamically maintain efficient data structures for handling files.

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

This framework occurs for instance in the case of Monte-Carlo applications where the parameters of new simulations depend on the average behavior of the simulations previously performed. The general principle is the following: several simulations (independent tasks) are launched simultaneously with different initial parameters, and then the average behavior of these simulations is computed. Then other simulations are performed with new parameters computed from the average behavior. These parameters are tuned to ensure a much faster convergence of the method. Running such an application on a semi-stable platform is a particular instance of the scheduling problem mentioned in Section 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. [63] presented a new method, called ABF, to compute the

[6] http://boinc.berkeley.edu/
[7] http://www.worldcommunitygrid.org/

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 [74].

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

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. 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 [8] initiated by Xavier Hanin (Jayasoft [9]).

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.

[8] http://xooctory.xoocode.org/

[9] http://www.jayasoft.fr/index.php

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.

4.3. Data Cubes

Data cube queries represent an important class of On-Line Analytical Processing (OLAP) queries in decision support systems. They consist in a pre-computation of the different group-bys of a database (aggregation for every combination of GROUP BY attributes) that is a very consuming task. For instance, databases of some megabytes may lead to the construction of a datacube requiring terabytes of memory [81] and parallel computation has been proposed but for a static and well-identified platform [64]. This application is typically an interesting example for which the distributed computation and storage can be useful in an heterogeneous and dynamic setting. We just started a collaboration with Noel Novelli (Assistant Professor of Marseille University) who is a specialist of datacube computation. Our goal is to rely on the set of services defined in Section 5.2.2 to compute and maintain huge datacubes.

4.4. Requests in Large Databases

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

We work on data-management issues and the adaptation of CRUSH [88] and Sorrento [86] protocols used to localize data. An important issue is the design of mechanisms to distribute data over the set of participating nodes. The objective is both to balance the load in terms of storage among the different storage devices and to balance the load in terms of processed requests among the different processing units. Given the dynamism of the requests and the files to be stored, the scale of the system and the risk of failure due to the large number of storage and processing units, we believe that the techniques developed in the context of P2P systems may also be used in the context of large distributed databases. To balance both loads (storage and requests), we plan to rely on the services described in Section 5.2.2.

5. New Results

5.1. Efficient queries and compact data structures

5.1.1. Compression and short data structures

5.1.1.1. 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. [69] for a survey of routing techniques).

Some techniques provide routes of length at most $1+\epsilon$ times the length of the shortest one while maintaining a poly-logarithmic number of entries per routing table [48], [62], [85]. 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 [49], [87]. Some experiments have shown that such methods, although universal, work very well in practice, in average, on realistic scale-free or existing topologies [80].

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 [58] have constructed (in polynomial time) network topologies for which nodes can be labeled once such that whatever the link weights vary in time, shortest path routing tables with compacity k can be designed, i.e., for each routing table the set of destinations using the same first outgoing edge can be grouped in at most k ranges of consecutive labels.

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 [22]. This later family of networks includes (but is not rectrict 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 hid large constant as we may encounter in Minor Graph Theory. This construction has been achieved by the design of new sparse cover for planar graphs, solving a problem open since STOC '93.

In [28], we have shown that routing if 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-celullar 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 [33] at a workshop on Peer-to-Peer, Routing in Complex Graphs, and Network Coding in Thomson Labs in Paris.

5.1.1.2. *Succinct representation of underlying topologies*

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

Classical data structures for representing a graph (matrix or list) can be significantly improved when the targeted graph falls in some specific classes or obeys to some properties: the graph has bounded genus (embeddable on surface of fixed genus), bounded tree-width (or c-decomposable), or embeddabble into a bounded page number [70], [71]. Typically, planar topologies with n nodes (thus embeddable on the plane with no edge crossings) can by efficiently coded in linear time with at most $5n+o(n)$ bits supporting adjacency queries in constant time. This improves the classical adjacency list within a non negligible $\log n$ factor on the size (the size is about $6n \log n$ bits for edge list), and also on the query time [61], [60], [59].

5.1.1.3. *Local data structures and other queries*

The basic routing scheme and the overlay networks must also allow us to route other queries than routing driven by applications. Typically, divide-and-conquer parallel algorithms require to compute many nearest common ancestor (NCA) queries in some tree decomposition. In a large scale platform, if the current tree structure is fully or partially distributed, then the physical location of the NCA in the platform must be optimized. More precisely, the NCA computation must be performed from distributed pieces of information, and then addressed via the routing overlay network (cf. [52] for distributed NCA algorithms).

Recently, a theory of localized data structures has been developed (initialized by [84]; see [72] for a survey). One associates with each node a label such that some given function (or predicate) of the node can be extracted from two or more labels. Theses labels are usually joined to the addresses or inserted into a global database index.

In relation with the project, queries involving the flow computation between any sink-target pair of a capacitated network is of great interest [76]. Dynamic labeling schemes are also available for tree models [78], [79], 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 [47], 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 prove that it is possible to assigned 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 there labels [38]. Classical representations of planar graphs in the distributed setting where 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 [37], [36]. Previous solutions achieve $\log n + O(k2\log\log n)$-bit labels [Alstrup-Bille-Rauhe 2005], whereas we have prove that $\log n + O(k\log\log n)$-bit labels suffice.

We also mention a keynote talk [34] at the LOCALITY workshop, an event joint to PODC '07 in Portland, about "Localized Data Structures".

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 [27]). 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.

5.1.1.4. Distributed Greedy Coloring

Distributed Greedy Coloring is an interesting and intuitive variation of the standard Coloring problem. It still consists in coloring in a distributed setting each node of a given graph in such a way that two adjacent nodes do not get the same color, but it adds a further constraint. Given an order among the colors, a coloring is said to be *greedy* if there does not exist a node for which its associated color can be replaced by a color of lower position in this order without violating the coloring property. In [35], we provide lower and upper bounds for this problem in Linial's model and we relate them to other well-known problems, namely *Coloring*, *Maximal Independent Set (MIS)*, and *Largest First Coloring*. Whereas the best known upper bound for Coloring, MIS, and Greedy Coloring are the same, we prove a lower bound which is strong in the sense that it now makes a difference between Greedy Coloring and MIS.

Within the wider context of the project, we have also considered the problems of periodic graph exploration with little memory [32], [42], [16], black hole search [20], [19] and gathering of asynchronous oblivious mobile robots in a ring [45], [18].

5.1.2. Overlay and small world networks

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

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 [50] 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 [65], [75]. 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 [83]. 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 [77], [68], [66].

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 [67], 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 [31], 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 [30], [29], 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 [13], [41], we describe a randomized algorithm for assigning neighbors to vertices joining a dynamic distributed network. The aim of the algorithm is to maintain connectivity, low diameter and constant vertex degree. On joining each vertex donates a constant number of tokens to the network. These tokens contain the address of the donor vertex. The tokens make independent random walks in the network. A token can be used by any vertex it is visiting to establish a connection to the donor vertex. This allows joining vertices to be allocated a random set of neighbors although the overall vertex membership of the network is unknown. The network we obtain in this way is robust under adversarial deletion of vertices and edges and actively reconnects itself.

Within the wider context of the project, we have published a book on information dissemination in optical networks [11]. We have also considered the problems of modeling of wireless networks [44], energy efficiency in wireless networks [14], [15], [39], [43], and bandwidth allocation in radio networks [21]. We have also investigated the problems of designing survivable networks [40], and of constructing identifying codes and locating-dominating codes in graphs [17].

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*

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 [55], [73] and Divisible Load Scheduling (DLS) [57], [56], [51].

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 [23], 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. In [24], 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 [12], we have proposed several distributed scheduling algorithms when several applications are to be simultaneously mapped onto an heterogeneous platform.

5.2.2. New services for processing on large scale distributed platforms

5.2.2.1. Heterogeneous dating service

Participants: Olivier Beaumont, Philippe Duchon, Miroslaw Korzeniowski.

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.

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 [46], we work on a similar famous NP-complete problem called *bin-covering problem*: Given a set of weighted elements, build the greatest number of bins such that the total weight of the elements in a bin is bigger than a certain constant. We also introduce a generalization of the problem: *bin-covering problem with distance constraint*: the elements are in some metric space, and the diameter of the bins is bounded by a certain constant. We consider algorithms distributed over the elements to be clustered. We propose a distributed $1/2$-approximation algorithm for the bin-covering problem. In the case of the bin-covering problem with distance constraint, we give a distributed $1/3$-approximation algorithm in the case where the elements are in a space of dimension 1. In both algorithms, each element sends at most $O(\log^2 n)$ messages (with high probability) to build the solution. We have further ongoing work for the cases where the elements are in a higher-dimensional space or even some abstract metric space.

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 [25] 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 [26].

6. Contracts and Grants with Industry

6.1. 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 plan to establish a formal collaboration on document storage in large distributed databases, request scheduling and independent tasks distribution across large distributed platforms.

6.2. Microsoft Research

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

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

7. Other Grants and Activities

7.1. National actions

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

Participant: Olivier Beaumont.

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. ACI "Masse de données" Navgraph (2003–2006)

Participant: Nicolas Hanusse.

Navgraph, lead by Nicolas Hanusse, is a project on data visualization based upon graph modeling. We mainly focus on applications on visual data mining for the navigation in huge graphs dedicated to video databases, genomic and topic maps.
The project involves the following laboratories: LaBRI, LRI, LIRMM, CLIPS/IMAG, LINA, LSC, IGM

7.1.3. ACI "Masse de Données" Pair à Pair (2003–2006)

Participants: Olivier Beaumont, Philippe Duchon, Cyril Gavoille, Ralf Klasing.

The goal of this ACI, lead by Laurent Viennot (Gyroweb-GANG), is the design of P2P protocols. A follow-up of "Pair-A-Pair" is under preparation, involving all members of GANG and Cepage, and should be submitted this year to "ANR Blanche" program.

7.1.4. ACI "Masse de Données" GeoComp (2004-2007)

Participants: Cyril Gavoille, Nicolas Bonichon.

GEOCOMP, lead by Gilles Schaeffer (LIX), tackles the problem of coding geometric data structures. Members of this project propose effective solutions to do the compression almost optimally without the need of a decompression process for basic requests on the structure.

7.1.5. 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 pre-projects CEPAGE and GANG.

7.1.6. ANDT "Aladdin" (submitted)

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

The objective of the (submitted) ANDT INRIA Aladdin [10] project is to continue and strengthen the efforts to develop the GRID'5000 platform. Olivier Beaumont is the local coordinator of Grid'5000 and is responsible (together with Frederic Vivien, INRIA project GRAAL) of the national working group on "Efficient exploitation of highly heterogeneous and hierarchical large-scale systems".

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.

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 Gavoille, 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/)

[10] note that this project is absolutely not related to the previous one, even if both projects have the same name!

7.3. Visites, et invitations de chercheurs

- Adrian Kosowski, Gdansk University of Technology, Pologne 03/02-15/02/2007 (STSM GRAAL/DYNAMO)
  - collaboration on: "Cost minimisation in multi-interface networks" and "Distributed Greedy Coloring"
  - 1 talk (GT Algorithmique Distribuee 12/02/2007): "Distributed implementation of greedy graph coloring algorithms."
- Robert Elsaesser, University of Paderborn, Germany 19/03-23/03/2007 (ALPAGE)
  - collaboration on: "Information dissemination on unstable platforms."
  - 1 talk (GT Algorithmique Distribuee 19/03/2007): "THE POWER OF MEMORY IN RANDOMIZED BROADCASTING."
- David Ilcinkas, Université du Québec, Canada 23/04-27/04/2007 (ALPAGE)
  - collaboration on: "graph exploration with little memory"
- Colin Cooper, King's College London (UK) 04/05-11/05/2007 (Royal Society Grant)
  - collaboration on: "algorithms for the web graph, peer-to-peer networks, graph exploration, black hole search"
- 22 juin 2007, 7 jours. Marcin Bienkowski, Wroclaw University, Pologne (STSM DYNAMO)
  - collaboration on: "Data management in networks."
  - 1 talk (GT Algorithmique Distribuee 26/06/2007) "Data management in networks."
- 13 juillet 2007, 14 jours. Colin Cooper, King's College London (UK) Tomasz Radzik, King's College London (UK) (Royal Society Grant)
  - collaboration on: "algorithms for the web graph, peer-to-peer networks, graph exploration, black hole search"
- Robert Elsaesser, University of Paderborn, Germany 31/08-12/10/2007 (Prof invité INRIA)
  - collaboration on: "Dating service - a tool to cope with heterogeneity in distributed systems."
  - 1 talk (GT Algorithmique Distribuee 01/10/2007) "Distributed implementation of greedy graph coloring algorithms."
- Alfredo Navarra 01/05/2006-30/04/2007 Postdoc University Bordeaux 1
  - collaboration on: "Distributed Coloring, Graph exploration with small memory, Cost minimisation in wireless networks."
  - 1 talk in 2007 (GT Algorithmique Distribuee 26/03/2007): "Fast Periodic Graph Exploration with Constant Memory."
- Miroslaw Korzeniowski 01/10/2006-30/09/2007 Postdoc INRIA
  - 1 talk in 2007 (GT Algorithmique Distribuee 18/06/2007): "Dating service - a tool to cope with heterogeneity in distributed systems."
- Planned visits:
  - Leszek Gasieniec, University of Liverpool (07/12/2007-14/12/2007)
  - Jurek Czyzowicz, Université du Québec à Hull (07/12/2007-14/12/2007)
  - Miroslaw Korzeniowski, Wroclaw University of Technology (27/11/2007-08/12/2007)

8. Dissemination

8.1. Community animation

8.1.1. Editorial Work

Ralf Klasing is a member of the Editorial Board of *Networks*, *Parallel Processing Letters*, and *Algorithmic Operations Research*.

8.1.1.1. Program Chair

- HeteroPar 07 (Olivier Beaumont, chair), International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks, Austin, 2007
- EuroPar'07 (Olivier Beaumont, Local Chair, Scheduling and Load Balancing), Rennes, France, 2007

8.1.1.2. Program Committees

- Olivier Beaumont
  - IPDPS 07 IEEE International Parallel and Distributed Processing Symposium, Long Beach, USA, 2007
  - PMGC'07 Workshop on Programming Models for Grid Computing, Rio de Janeiro, Brazil, 2007
  - PMAA 08 International Workshop on Parallel Matrix Algorithms and Applications, Rennes, France
  - RenPar 08 Rencontre du Parallélisme, Le Croisic, France
  - HeteroPar 08 International Workshop on Algorithms, models, and tools for parallel computing on heterogeneous networks (Cork, Ireland)
- Cyril Gavoille
  - DISC '07 (Sep/Oct, Lemesos, Cyprus) International Symposium on Distributed Computing
  - SPAA '07 (June 9-11, San Diego, Californie, USA) Symposium on Parallelism in Algorithms and Architectures
  - AlgoTel '07 (May 29 - Jun 1, Ile d'Oléron, France) Rencontres Francophones sur les aspects Algorithmiques des Télécommunications
  - PDCN '07 (Feb. 13-15, Innsbruck, Austria) Parallel and Distributed Computing and Networks

8.1.2. Organizing Commitee

Nicolas Hanusse was president of the organizing committee for Algotel'07. Nicolas Bonichon, Cyril Gavoille and Ralf Klasing were members of this organizing committee.

8.2. Teaching activities

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

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

- Nicolas Hanusse
  – Graph algorithms for data visualization (2nd year MASTER "Models and Algorithms" - 2005 and 2006)
  – Distributed computing (2nd year MASTER "Models and Algorithms" - 2006)
- Cyril Gavoille
  – Introduction to Distributed Computing (2nd year MASTER "Models and Algorithms" - 2005, 2006)
  – Algorithms and Communications in Networks (2nd year MASTER "Models and Algorithms" - 2005, 2006)

– Communication and Routing (last year of engineering school ENSEIRB 2005, 2006)

- Olivier Beaumont
  – Routing and P2P Networks (last year of engineering school ENSEIRB, 2005)
- Philippe Duchon
  – Randomized Algorithms (2nd year MASTER "Models and Algorithms" - 2006)

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 heterogeneousplatforms*, 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^{o} 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^{o} 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^{o} 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.

[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", PODC 20-Year Special Issue, vol. 16, May 2003, p. 111-120.

[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

Books and Monographs

[11] S. BANDYOPADHYAY. *Dissemination of Information in Optical Networks: From Technology to Algorithms*, Springer Monograph, in cooperation with Ralf Klasing, to appear, Springer-Verlag, 2007, http://www.springer.com/france/home?SGWID=7-102-22-173750238-0&changeHeader=true.

Articles in refereed journals and book chapters

[12] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Centralized Versus Distributed Schedulers Multiple Bag-of-Tasks Applications*, in "IEEE Trans. Parallel Distributed Systems", 2007.

[13] C. COOPER, R. KLASING, T. RADZIK. *A randomized algorithm for the joining protocol in dynamic distributed networks*, in "Theoretical Computer Science", to appear, 2007, http://www.elsevier.com/locate/tcs.

[14] M. FLAMMINI, R. KLASING, A. NAVARRA, S. PÉRENNES. *Improved approximation results for the Minimum Energy Broadcasting problem in Wireless Ad Hoc Networks*, in "Algorithmica", Online First, 2007, http://dx.doi.org/10.1007/s00453-007-9077-7.

[15] M. FLAMMINI, R. KLASING, A. NAVARRA, S. PÉRENNES. *Tightening the upper bound for the Minimum Energy Broadcasting*, in "Wireless Networks", Online First, 2007, http://dx.doi.org/10.1007/s11276-006-0007-4.

[16] L. GASIENIEC, R. KLASING, R. MARTIN, A. NAVARRA, X. ZHANG. *Fast Periodic Graph Exploration with Constant Memory*, in "Journal of Computer and System Sciences", to appear, 2007, http://dx.doi.org/10.1016/j.jcss.2007.09.004.

[17] S. GRAVIER, R. KLASING, J. MONCEL. *Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs*, in "Algorithmic Operations Research", to appear, 2007, http://journals.hil.unb.ca/index.php/AOR.

[18] R. KLASING, E. MARKOU, A. PELC. *Gathering asynchronous oblivious mobile robots in a ring*, in "Theoretical Computer Science", to appear, 2007, http://www.elsevier.com/locate/tcs.

[19] R. KLASING, E. MARKOU, T. RADZIK, F. SARRACCO. *Approximation bounds for Black Hole Search problems*, in "Networks", to appear, 2007, http://eu.wiley.com/WileyCDA/WileyTitle/productCd-NET.html.

[20] R. KLASING, E. MARKOU, T. RADZIK, F. SARRACCO. *Hardness and approximation results for black hole search in arbitrary graphs*, in "Theoretical Computer Science", vol. 384, n^{o} 2–3, October 2007, p. 201–221, http://dx.doi.org/10.1016/j.tcs.2007.04.024.

[21] R. KLASING, N. MORALES, S. PÉRENNES. *On the Complexity of Bandwidth Allocation in Radio Networks*, in "Theoretical Computer Science", to appear, 2007, http://www.elsevier.com/locate/tcs.

Publications in Conferences and Workshops

[22] 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.

[23] O. Beaumont, N. Bonichon, L. Eyraud-Dubois. *Scheduling Divisible Workload on Heterogeneous Platforms under Bounded Multi-Port Model*, in "International Heterogeneity in Computing Workshop", To appear, 2008.

[24] 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.

[25] 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.

[26] O. Beaumont, A. Kermarrec, E. Rivière. *Peer to peer multidimensional overlays: Approximating complex structures*, in "OPODIS", 2007.

[27] B. Courcelle, C. Gavoille, M. Kanté, D. A. Twigg. *Forbidden-Set Labeling on Graphs*, in "2nd Workshop on Locality Preserving Distributed Computing Methods (LOCALITY)", Co-located with PODC 2007, August 2007.

[28] 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.

[29] P. Duchon, N. Eggemann, N. Hanusse. *Non-Navigability of random scale-free graphs*, in "OPODIS - International Conference Of Principles of Distributed Systems", 2007.

[30] P. Duchon, N. Eggemann, N. Hanusse. *Non-searchability of random scale-free graphs*, in "PODC - Principles Of Distributed Computing", 2007, p. 380-381.

[31] 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.

[32] L. Gasieniec, R. Klasing, R. Martin, A. Navarra, X. Zhang. *Fast Periodic Graph Exploration with Constant Memory*, in "Proceedings of the 14th Colloquium on Structural Information and Communication Complexity (SIROCCO 2007)", Lecture Notes in Computer Science, vol. 4474, Springer Verlag, June 2007, p. 26–40, http://dx.doi.org/10.1007/978-3-540-72951-8_4.

[33] C. Gavoille. *An Overview on Compact Routing*, in "Workshop on Peer-to-Peer, Routing in Complex Graphs, and Network Coding", March 2007.

[34] C. Gavoille. *Localized Data Structures (Keynote Talk)*, in "2nd Workshop on Locality Preserving Distributed Computing Methods (LOCALITY)", Co-located with PODC 2007, August 2007.

[35] C. Gavoille, R. Klasing, A. Kosowski, A. Navarra. *On the Complexity of Distributed Greedy Coloring*, in "Proceedings of the 21st International Symposium on Distributed Computing (DISC 2007)",

Lecture Notes in Computer Science, vol. 4731, Springer Verlag, September 2007, p. 482–484, http://dx.doi.org/10.1007/978-3-540-75142-7_37.

[36] 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.

[37] 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.

[38] 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.

[39] R. KLASING, A. KOSOWSKI, A. NAVARRA. *Cost minimisation in multi-interface networks*, in "Proceedings of the 1st Annual International Conference on Network Control and Optimization (NET-COOP 2007)", Lecture Notes in Computer Science, vol. 4465, Springer Verlag, June 2007, p. 276–285, http://dx.doi.org/10.1007/978-3-540-72709-5_29.

Internal Reports

[40] H.-J. BÖCKENHAUER, D. BONGARTZ, J. HROMKOVIČ, R. KLASING, G. PROIETTI, S. SEIBERT, W. UNGER. *On k-Connectivity Problems with Sharpened Triangle Inequality*, Technical Report, n^{o} RR-1430-07, LaBRI, May 2007, http://www.labri.fr/perso/lepine/Rapports_internes.

[41] C. COOPER, R. KLASING, T. RADZIK. *A randomized algorithm for the joining protocol in dynamic distributed networks*, Technical Report, n^{o} RR-1432-07, LaBRI, June 2007, http://www.labri.fr/perso/lepine/Rapports_internes.

[42] L. GASIENIEC, R. KLASING, R. MARTIN, A. NAVARRA, X. ZHANG. *Fast Periodic Graph Exploration with Constant Memory*, Technical Report, n^{o} RR-1426-07, LaBRI, April 2007, http://www.labri.fr/perso/lepine/Rapports_internes.

[43] R. KLASING, A. KOSOWSKI, A. NAVARRA. *Cost Minimisation in Wireless Networks with a Bounded and Unbounded Number of Interfaces*, Technical Report, n^{o} RR-1436-07, LaBRI, September 2007, http://www.labri.fr/perso/lepine/Rapports_internes.

[44] R. KLASING, Z. LOTKER, A. NAVARRA, S. PÉRENNES. *From Balls and Bins to Points and Vertices*, Technical Report, n^{o} RR-1437-07, LaBRI, October 2007, http://www.labri.fr/perso/lepine/Rapports_internes.

[45] R. KLASING, E. MARKOU, A. PELC. *Gathering asynchronous oblivious mobile robots in a ring*, Technical Report, n^{o} RR-1422-07, LaBRI, January 2007, http://www.labri.fr/perso/lepine/Rapports_internes.

Miscellaneous

[46] H. LARCHEVÊQUE. *Bin covering distribué avec et sans diamètre contraint pour les plates formes à grande échelle*, Technical report, Université Bordeaux 1, 2007.

References in notes

[47] S. Abiteboul, S. Alstrup, H. Kaplan, T. Milo, T. Rauhe. *Compact Labeling Schemes for Ancestor Queries*, in "SIAM Journal on Computing", 2005.

[48] I. Abraham, C. Gavoille, A. V. Goldberg, D. Malkhi. *Routing in Networks with Low Doubling Dimension*, Technical report, n^o MSR-TR-2005-175, Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052 - http://www.research.microsoft.com, December 2005.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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^o 4, 2004, p. 319-330.

[56] 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.

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

[58] 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.

[59] 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.

[60] 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.

[61] 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)", 270-284, vol. 3353 of Lecture Notes in Computer Science, Springer, June 2004.

[62] 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.

[63] E. Darve, A. Pohorille. *Calculating free energies using average force*, in "Journal of Chemical Physics", vol. 115, 2001, p. 9169-9183.

[64] 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^{o} 2, 2002, p. 181-201.

[65] 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^{o} 1, 1988, p. 8-32.

[66] 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.

[67] 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 Pess, ACM SIGACT - ACM SIGARCH, July 2006, p. 225-232, http://www.labri.fr/publications/combalgo/2006/DHLS06.

[68] 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.

[69] C. Gavoille. *Routing in Distributed Networks: Overview and Open Problems*, in "ACM SIGACT News - Distributed Computing Column", vol. 32, n^{o} 1, March 2001, p. 36-52.

[70] C. Gavoille, N. Hanusse. *On Compact Encoding of Pagenumber k Graphs*, in "Discrete Mathematics & Theoretical Computer Science", To appear, 2005.

[71] 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.

[72] C. Gavoille, D. Peleg. *Compact and Localized Distributed Data Structures*, in "Journal of Distributed Computing", PODC 20-Year Special Issue, vol. 16, May 2003, p. 111-120.

[73] 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.

[74] J. HÉNIN, C. CHIPOT. *Overcoming free energy barriers using unconstrained molecular dynamics simulations*, in "Journal of Chemical Physics", vol. 121, 2004, p. 2904–2914.

[75] R. M. KARP, C. SCHINDELHAUER, S. SHENKER, B. VÖCKING. *Randomized Rumor Spreading.*, in "FOCS", 2000, p. 565-574.

[76] M. KATZ, N. A. KATZ, A. KORMAN, D. PELEG. *Labeling schemes for flow and connectivity*, in "SIAM Journal on Computing", vol. 34, n^o 1, 2004, p. 23-40.

[77] 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.

[78] 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.

[79] 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.

[80] D. KRIOUKOV, K. FALL, X. YANG. *Compact Routing on Internet-Like Graphs*, in "IEEE INFOCOM", To appear, 2004.

[81] M. LAPORTE, N. NOVELLI, R. CICCHETTI, 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.

[82] A. LEGRAND, F. MAZOIT, M. QUINSON. *An Application-level network mapper*, Research Report, n^o RR-2003-09, LIP, ENS Lyon, France, feb 2003.

[83] S. MILGRAM. *The small world problem*, in "Psychology Today", vol. 61, n^o 1, 1967.

[84] 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.

[85] A. SLIVKINS. *Distance Estimation and Object Location via Rings of Neighbors*, in "24th Annual ACM Symposium on Principles of Distributed Computing (PODC)", Appears earlier as Cornell CIS technical report TR2005-1977, ACM Press, July 2005, p. 41-50.

[86] H. TANG, A. GULBEDEN, J. ZHOU, L. CHU, T. YANG. *Sorrento: a self-organizing storage cluster for parallel data-intensive applications*, in "International Conf for High Performance Computing Networking and Storage, Pittsburgh, PA, USA", 2004.

[87] 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.

[88] S. WEIL, S. BRANDT, E. MILLER, C. MALTZAHN. *CRUSH: Controlled, scalable, decentralized placement of replicated data*, in "Proceedings of the 2006 ACM,IEEE Conference on Supercomputing (SC06)".