INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Team Regal*

# *Resource management in large scale distributed systems*

*Paris - Rocquencourt*

THEME COM

Activity Report

2008

# Table of contents

*Regal is a common project with CNRS and Unviersit of Paris 6 through the "Laboratoire d'Informatique de Paris 6",* LIP6 *(UMR 7606).*

# 1. Team

**Research Scientist**

Marc Shapiro [ Research Director (DR) INRIA, HdR ]
Mesaac Makpangou [ Research associate (CR) Inria, HdR ]

**Faculty Member**

Pierre Sens [ Team Leader, Professor Univeristé Paris 6, HdR ]
Luciana Arantes [ Associate professor Université Paris 6 ]
Maria Gradinariu [ Associate professor Université Paris 6 ]
Bertil Folliot [ Professor Université Paris 6, HdR ]
Olivier Marin [ Associate professor Université Paris 6 ]
Sebastien Monnet [ Associate professor Université Paris 6 ]
Gaël Thomas [ Associate professor Université Paris 6 ]
Ikram Chabbouh [ Assistant professor Université (ATER) Paris 6 ]
Gilles Muller [ Professor ENM, Nantes, France, Delegation in Regal since September ]

**Technical Staff**

Jean-Michel Busca [ Research Engineer ]
Véronique Martin [ Associate Engineer ]

**PhD Student**

Lamia Benmouffok [ Microsoft research grant - Université Paris 6 ]
Mathieu Bouillaguet [ Université Paris 6 ]
Charles Clement [ Université Paris 6 ]
Nicolas Geoffray [ Université Paris 6 ]
Nicolas Hidalgo [ Université Paris 6 ]
Corentin Mehat [ Université Paris 6 ]
Thomas Preud'homme [ Université Paris 6 ]
Erika Rosas [ Université Paris 6 ]
Julien Sopena [ Université Paris 6 ]
Pierre Sutra [ Université Paris 6 ]
Mathieu Valéro [ Université Paris 6 ]

**Administrative Assistant**

Nadia Mesrar [ Secretary Inria ]

# 2. Overall Objectives

## 2.1. Overall Objectives

The main focus of the Regal team is research on large-scale distributed computing systems, and addresses the challenges of automated adminstration of highly dynamic networks, of fault tolerance, of consistency in large-scale distributed systems, of information sharing in collaborative groups, of dynamic content distribution, and of operating system adaptation. Regal is a joint research team between LIP6 and INRIA-Paris-Rocquencourt.

## 2.2. Highlights

- Participation to 2 new projects funding by ANR: SHAMAN and R-DISCOVER.
- Majors publications in 2008 : Sofware Pratice and Experience Journal, Multiagent and Grid Systems Journal [15], [16], Euro-Par 2008 [26], [27],IEEE NCA 2008, [18], OPODIS 2008 [24].
- Gilles (Professor) has joined the team in September.
- 1 PhD defense : Julien Sopena (dec. 2007).

# 3. Scientific Foundations

## 3.1. Scientific Foundations

**Keywords:** *Grid computing*, *Peer-to-peer*, *consistency*, *distributed system*, *dynamic adaptation*, *fault tolerance*, *large scale environments*, *replication*. .

Scaling to large configurations is one of the major challenges addressed by the distributed system community lately. The basic idea is how to efficiently and transparently use and manage resources of millions of hosts spread over a large network. The problem is complex compared to classical distributed systems where the number of hosts is low (less than a thousand) and the inter-host links are fast and relatively reliable. In such "classical" distributed architectures, it is possible and reasonable to build a single image of the system so as to "easily" control resource allocation.

In large configurations, there is no possibility to establish a global view of the system. The underlying operating system has to make decisions (on resource allocation, scheduling ...) based only on partial and possibly wrongs view of the resources usage.

Scaling introduces the following problems:

- Failure: as the number of hosts increases, the probability of a host failure converges to one. [1] Compared to classical distributed systems, failures are more common and have to be efficiently processed.
- Asynchronous networks: on the Internet, message delays vary considerably and are unbounded.
- Impossibility of consensus:In such an asynchronous network with failures, consensus cannot be solved deterministically (the famous Fischer-Lynch-Patterson impossibility result of 1985).The system can only approximate, suspecting hosts that are not failed, or failing to suspect hosts that have failed. As a result, no host can form a consistent view of system state.
- Failure models: the classical view of distributed systems considers only crash and omission failures. In the context of large-scale, open networks, the failure model must be generalised to include stronger attacks. For instance, a host can be taken over ("zombie") and become malicious. Arbitrary faults, so-called Byzantine behaviours, are to be expected and must be tolerated.
- Managing distributed state: In contrast to a local-area network, establishing a global view of a large distributed system system is unfeasible. The operating system must make its decisions, regarding resource allocation or scheduling, based on partial and incomplete views of system state.

Two architectures in relation with the scaling problem have emerged during the last years:

Grid computing: Grid computing offers a model for solving massive computational problems using large numbers of computers arranged as clusters interconnected by a telecommunications infrastructure as internet, renater or VTHD.

[1] For instance if we consider a classical host MTBF (Mean Time Between Failure) equals to 13 days, in a middle scale system composed of only 10000 hosts, a failure will occur every 4 minutes.

If the number of involved hosts can be high (several thousands), the global environment is relatively controlled and users of such systems are usually considered safe and only submitted to host crash failures (typically, Byzantine failures are not considered).

Peer-to-peer overlay network: Generally, a peer-to-peer (or P2P) computer network is any network that does not rely on dedicated servers for communication but, instead, mostly uses direct connections between clients (peers). A pure peer-to-peer network does not have the notion of clients or servers, but only equal peer nodes that simultaneously function as both "clients" and "servers" with respect to the other nodes on the network.

This model of network arrangement differs from the client-server model where communication is usually relayed by the server. In a peer-to-peer network, any node is able to initiate or complete any supported transaction with any other node. Peer nodes may differ in local configuration, processing speed, network bandwidth, and storage capacity.

Different peer-to-peer networks have varying P2P overlays. In such systems, no assumption can be made on the behavior of the host and Byzantine behavior has to be considered.

Regal is interested in how to adapt distributed middleware to these large scale configurations. We target Grid and Peer-to-peer configurations. This objective is ambitious and covers a large spectrum. To reduce its spectrum, Regal focuses on fault tolerance, replication management, and dynamic adaptation.

We concentrate on the following research themes:

Data management: the goal is to be able to deploy and locate effectively data while maintening the required level of consistency between data replicas.

System monitoring and failure detection: we envisage a service providing the follow-up of distributed information. Here, the first difficulty is the management of a potentially enormous flow of information which leads to the design of dynamic filtering techniques. The second difficulty is the asynchronous aspect of the underlying network which introduces a strong uncertainty on the collected information.

Adaptive replication: we design parameterizable techniques of replication aiming to tolerate the faults and to reduce information access times. We focus on the runtime adaptation of the replication scheme by (1) automatically adjusting the internal parameters of the strategies and (2) by choosing the replication protocol more adapted to the current context.

The dynamic adaptation of application execution support: the adaptation is declined here to the level of the execution support (in either of the high level strategies). We thus study the problem of dynamic configuration at runtime of the low support layers.

# 4. Application Domains

## 4.1. Application Domains

**Keywords:** *Internet services*, *data storage*, *data-sharing*, *multi-agent systems*.

As we already mentioned, we focus on two kinds of large scale environments: computational grids and peer-to-peer (P2P) systems. Although both environments have the same final objective of sharing large sets of resources, they initially emerged from different communities with different context assumptions and hence they have been designed differently. Grids provide support for a large number of services needed by scientific communities. They usually target thousands of hosts and hundreds of users. Peer-to-peer environments address millions of hosts with hundreds of thousands of simultaneous users but they offer limited and specialized functionalities (file sharing, parallel computation).

In peer-to-peer configurations we focus on the following applications:

- Internet services such as web caches or content distribution network (CDN) which aim at reducing the access time to data shared by many users,
- Data storage of mutable data. Data storage is a classical peer-to-peer application where users can share documents (audio and video) across the Internet. A challenge for the next generation of data sharing systems is to provide update management in order to develop large cooperative applications.
- multi-player games. The recent involvement of REGAL in the PLAY ALL project gives us the opportunity to consider distributed interactive video games. Theses applications are very interesting for us since they bring new constraints, most specifically on latency.

In Grid configurations we address resource management for two kinds of applications:

- Multi-agent applications which model complex cooperative behaviors.
- Application Service Provider (ASP) environments in cooperation with the DIET project of the GRAAL team.

Our third application domain is based on data sharing. Whereas most work on P2P applications focuses on write-once single-writer multiple-reader applications, we consider the (more demanding) applications that share mutable data in large-scale distributed settings. Some examples are co-operative engineering, collaborative authoring, or entreprise information libraries: for instance co-operative code development tools or decentralized wikis. Such applications involve users working from different locations and at different times, and for long durations. In such settings, each user *optimistically* modifies his private copy, called a replica, of a shared datum. As replicas may diverge, this poses the problem of reconciliation. Our research takes into account a number of issues not addressed by previous work, for instance respecting application semantics, high-level operations, dependence, atomicity and conflict, long session times, etc.

# 5. Software

## 5.1. Pastis: A peer-to-peer file system

**Participants:** Pierre Sens [correspondent], Jean-Michel Busca.

Pastis is a distributed multi-writer file system. It aims at making use of the aggregate storage capacity of hundreds of thousands of PCs connected to the Internet by means of a completely decentralized peer-to-peer (P2P) network. Replication allows persistent storage in spite of a highly transient node population, while cryptographic techniques ensure the authenticity and integrity of file system data.

Routing and data storage in Pastis are handled by the Pastry routing protocol and the PAST distributed hash table (DHT). The good locality properties of Pastry/PAST allow Pastis to minimize network access latencies, thus achieving a good level of performance when using a relaxed consistency model. Moreover, Pastis does not employ heavy protocols such as BFT (Byzantine Fault Tolerance), like other P2P multi-writer file systems do. In Pastis, for a file system update to be valid, the user must provide a certificate signed by the file owner which proves that he has write access to that file.

## 5.2. LS3: Large Scale Simulator

**Participants:** Pierre Sens [correspondent], Jean-Michel Busca.

LS3 is a discrete event simulator originally developed for Pastis, a peer-to-peer file system based on Pastry. LS3 allows to build a network of tenths of thousands nodes on a single computer, and simulate its execution by taking into account message transmission delays. LS3 transparently simulates communication layers between nodes, and executes the same application code (including Pastry, Past and higher layers) as in a real execution of the system.

LS3's modular design consists of three independent layers, allowing the simulator to be reused in areas other than Pastis and Pastry:

- At the kernel level, the system being simulated is described in a generic way in terms of entities triggered by events. Each entity has a current state and a current virtual time, and can be programmed either in synchronous mode (blocking wait of the next event) or in asynchronous mode (activation of an event handler). A multi-threaded event engine delivers events in chronological order to each entity by applying a conservative scheduling policy, based on the analysis of event dependencies.
- At the network level, the system being simulated is modeled in terms of nodes sending and receiving messages, and connected through a network. The transmission delay of a message is derived from the distance between the sending and the receiving nodes in the network, according to the selected topology. Three topologies can be used: local network (all nodes belong to the same local network), two-level hierarchy (nodes are grouped into LANs connected through WANs) and sphere (nodes are located on a sphere). It is possible to set the jitter rate of transmission delays, as well as the rate of message loss in the network.
- The stubs Pastry level interfaces Pastry with LS3: it defines a specialization of Pastry nodes that allows them to interface with standard LS3 nodes. Several parameters and policies that drive the behaviour and the structure of a Pastry network can be set at this level, including: the distribution of node ids, the selection of boostrap nodes, the periodicity of routing tables checks and the rate of node churn. It is also possible to simulate the ping messages that nodes send to supervise each other, and set failure detection thresholds.

Some figures: LS3 can simulate a network of 20 000 Pastry nodes with no application within 512 Mb of RAM, and it takes approximately 12 minutes on a single processor Pentium M 1,7 GHz to build such network. When simulating the Pastis application, event processing speed is about 500 evt/s. The speedup factor depends on the simulated load: as an example, speedup ranges from 20 for a single user to 0,05 for 400 simultaneous users.

## 5.3. Telex

**Participants:** Marc Shapiro [correspondent], Lamia Benmouffok, Jean-Michel Busca, Pierre Sutra, Georgios Tsoukalas.

Developing write-sharing applications is challenging. Developers must deal with difficult problems such as managing distributed state, disconnection, and conflicts. Telex is an application-independent platform to ease development and to provide guarantees. Telex is guided by application-provided parameters: actions (operations) and constraints (concurrency control statements). Telex takes care of replication and persistence, drives application progress, and ensures that replicas eventually agree on a correct, common state. Telex supports partial replication, i.e., sites only receive operations they are interested in. The main data structure of Telex is a large, replicated, highly dynamic graph; we discuss the engineering trade-offs for such a graph and our solutions. Our novel agreement protocol runs Telex ensures, in the background, that replicas converge to a safe state. We conducted an experimental evaluation of the Telex based on a cooperative calendar application and on benchmarks.

# 6. New Results

## 6.1. Introduction

In 2008, we focused our research on the following areas:

- distributed algorithms for large and dynamic networks,
- Peer-to-peer storage
- dynamic adaptation of virtual machines,
- services management in large scale environments,
- Formal and practical study of optimistic replication, incorporating application semantics.
- Decentralized commitment protocols for semantic optimistic replication.
- dynamic replication in distributed multi-agent systems.

## 6.2. Distributed algorithms

**Participants:** Luciana Arantes [correspondent], Maria Gradinariu [correspondent], Mathieu Bouillaguet, Pierre Sens, Julien Sopena.

Our current research in the context of distributed algorithms focuses on two main axes. We are interested in providing fault-tolerant and self*(self-organizing, self-healing and self-stabilizing) solutions for fundamental problems in distributed computing. More precisely, we target the following basic blocks: mutual exclusion, resources allocation, agreement and communication primitives. We propose solutions for both static (eg. grid) and dynamic networks (P2P and mobile networks).

### *6.2.1. Static systems*

We have been interested in scalability aspects of distributed mutual exclusion algorithms, failure detection mechanisms and transformers in order to easy build distributed systems.

- In mutual exclusion algorithm, we have been interested in scalability, fault-tolerance and latency tolerance aspects of distributed mutual exclusion algorithms.

  The majority of current distributed mutual exclusion algorithms are not suited for distributed applications on top of large-scale or heterogeneous systems such as Grid or peer-to-peer systems since they do not consider the latency heterogeneity of the platform or scalable fault tolerance mechanisms. In [26], [25] we propose a new composition approach to mutual exclusion algorithms for applications spread over a grid which is composed of a federation of clusters. Taking into account the heterogeneity of communication latency, our hierarchical architecture combines intra and inter cluster algorithms. We focus on token-based algorithms and study different compositions of algorithms. Performance evaluation tests have been conducted on a national grid testbed whose results show that our approach is scalable and that the choice of the most suitable inter cluster algorithm depends on the behavior of the application.
- In 2008, we study a generalisation of the mutual exclusion problem, namely the $k$-Mutex problem. The $k$-mutual exclusion problem is a fundamental distributed problem which guarantees the integrity of the $k$ units of a shared resource by restricting the number of process that can simultaneous access them. We propose a fault tolerant permission-based k-mutual exclusion algorithm, which is an extension of Raymond's algorithm. Tolerating up to $n-1$ failures, our algorithm keeps its effectiveness despite failures. It uses information provided by unreliable failure detectors to dynamically detect crashes of nodes. Performance evaluation experiments show the performance of our algorithm compared to Raymond's when faults are injected. This work was published in [17].

### *6.2.2. Dynamic systems*

In this context we are interested in designing building blocks for distributed applications such as: failure detectors, adequate communication primitives (publish/subscribe) and overlays. Moreover, we are interested in solving fundamental problems such as leader election and naming.

- Since 2006, we consider the problem of failure detection in dynamic network such as MANET or Peer-to-peer overlay [24]. Most of implementations consider a set of known processes fully connected by reliable links. Such an assumption is not applicable in dynamic environments. Furthermore, the majority of current failure detector implementations are timer-based ones while in dynamic networks there is not an upper bound for communication delays. We propose an asynchronous implementation of a failure detector for dynamic environments. Our implementation is an query-response algorithm which does not rely on timers to detect failures. We assume that neither the identity nor the number of nodes are initially known. We also prove that our algorithm can implement failure detectors of class $\diamond S$ when some behavioral properties are satisfied by the underlying system.

  Furthermore in mobile environment, nodes can move around and voluntarily leave or join the network. Furthermore, they can crash or be disconnected from the network due to the absence of network signals. Therefore, failure, disconnection and mobility may create partitions in wireless networks which should be detected for fault and disconnection tolerance reasons. In [18], we propose an architecture of local and distributed detectors for mobile networks that detect failures, disconnections, and partitions. It is basically composed of three unreliable detectors: a heartbeat failure detector, a vector-based disconnection detector, and an eventually perfect partition detector. The latter ensures the convergence of detection information within a partition where all participants suspect the same sets of disconnected, unreachable, and faulty processes.

- The main challenges of our research activity over 2008 year were to develop self* (self-stabilizing, self-organizing and self-healing) algorithms for static and dynamic networks (P2P, sensor and robot networks). Each of these systems has its own specificity in terms of network characteristics. Therefore algorithms for these systems should be adapted to the network specificity. For example in P2P systems nodes can communicate with the nodes in the acquittance neighborhood while in sensor and robot networks the communication is restricted to the communication range limited by the physical communication power of each node. We addressed fundamental problems such as communication primitives and infrastructures in P2P and sensor networks, leader election and pattern formation in robot networks and conflict managers and leader election for static networks. All the proposed algorithms are self* and cope with the network dynamicity in the case of P2P, sensor or robot networks.

## 6.3. Peer-to-peer systems

**Participants:** Pierre Sens [correspondent], Jean-Michel Busca, Nicolas Hidalgo, Sebastien Monnet, Véronique Simon, Mathieu Valéro.

### *6.3.1. Peer-to-peer storage*

Since 2003, we develop Pastis [7] is a new completely decentralized multi-user read-write peer-to-peer file system. Pastis is based on the FreePastry Distributed Hash Table (DHT) of the Rice University. DHTs provide a means to build a completely decentralized, large-scale persistent storage service from the individual storage capacities contributed by each node of the peer-to-peer overlay

However, persistence can only be achieved if nodes are highly available, that is, if they stay most of the time connected to the overlay. Churn (i.e., nodes connecting and disconnecting from the overlay) in peer-to-peer networks is mainly due to the fact that users have total control on theirs computers, and thus may not see any benefit in keeping its peer-to-peer client running all the time. Since 2007, we study the effects of churn on Pastis, a DHT-based peer-to-peer file system. We evaluate the behavior of Pastis under churn, and investigate whether it can keep up with changes in the peer-to-peer overlay. We used a modified version of the PAST DHT to provide better support for mutable data and to improve tolerance to churn. Our replica regeneration protocol distinguishes between mutable blocks and immutable blocks to minimize the probability of data loss. Read-write quorums provide a good compromise to ensure replica consistency under the presence of node failures. Our experiments use Modelnet to emulate wide-area latencies and the asymmetric band- width of ADSL client links. The results show that Pastis preserves data consistency even at relatively high levels of churn. However, when connection/disconnection frequency is too high in the system, data-blocks may be lost. This is true for most current DHT-based system's implementations. To avoid this problem, it is necessary to build really efficient replication and maintenance mechanisms. We actually study the effect of churn on an existing DHT-based P2P system namely PAST/Pastry. We then propose solutions to enhance churn support and evaluate then through discrete event simulations.

### *6.3.2. Peer-to-peer overlay*

A malleable peer-to-peer overlay to take into account applications needs. Peer-to-peer overlays allow distributed applications to work in a wide-area, scalable, and fault-tolerant manner. However, most structured and unstructured overlays present in literature today are inflexible from the application viewpoint. In other words, the application has no control over the structure of the overlay itself. This paper proposes the concept of an application-malleable overlay, and the design of the first malleable overlay which we call MOve. In

MOve, the communication characteristics of the distributed application using the overlay can influence the overlay's structure itself, with the twin goals of (1) optimizing the application performance by adapting the overlay, while also (2) retaining the scale and fault-tolerance of the overlay approach. The influence could either be explicitly specified by the application or implicitly gleaned by our algorithms. Besides neighbor list membership management, MOve also contains algorithms for resource discovery, update propagation, and churn-resistance. The emergent behavior of the implicit mechanisms used in MOve manifest in the following way: when application communication is low, most overlay links keep their default configuration; however, as application communication characteristics become more evident, the overlay gracefully adapts itself to the application.

We are currently considering a new class of target applications : massively multiplayer online games (MMOG) such as virtual worlds. Within the context of a project funded by the LIP6, we have modeled a P2P distribution of such applications. Following this model, groups of object replicas are moving among peers while players evolve in the virtual world. For this kind of applications, it is important that the underlying overlay is flexible in order to remain adapted to the changing application structure.

## 6.4. Virtual machine (VM)

**Participants:** Gaël Thomas [correspondent], Bertil Folliot, Charles Clement, Nicolas Goeffray, Gilles Muller, Thomas Preud'homme.

Our research interest are in computer systems, particularly operating systems and virtual machines. We focuss on resource management, isolation and concurrency management in virtual machines. Since September 2008, we start with Gilles Muller a new complementary research theme on dynamic patches of operating systems.

### *6.4.1. Virtual machines*

Isolation in OSGi: The OSGi framework is a Java-based, centralized, component oriented platform. It is being widely adopted as an execution environment for the development of extensible applications. However, current Java Virtual Machines are unable to isolate components from each others. By modifying shared variables or allocating too much memory, a malicious component can freeze the complete platform. I work on I-JVM, a Java Virtual Machines that provides a lightweight approach to isolation while preserving the compatibility with legacy OSGi applications. Our evaluation of I-JVM shows that it solves the 15 known OSGi vulnerabilities due to the Java Virtual Machine with an overhead below $20\%$.

VMKit: Developing and optimizing a virtual machine (VM) is a tedious task that requires many years of development. Although VMs share some common principles, such as a Just In Time Compiler or a Garbage Collector, this opportunity for sharing hash not been yet exploited in implementing VMs. Our work on VMKit is a first attempt to build a common substrate that eases the development of high-level VMs. VMKit has been successfully used to build three VMs: a Java Virtual Machine, a Common Language Runtime and a lisp-like language with type inference $uvml$. Our precise contribution is an extensive study of the lessons learnt in implementing such common infrastructure from a performance and an ease of development standpoint. Our performance study shows that VMKit does not degrade performance on CPU-intensive applications, but still requires engineering efforts to compete with other VMs on memory-intensive applications.

### *6.4.2. Semantic patches*

Open source infrastructure software, such as the Linux operating system, Web browsers and n-tier servers, has become a well-recognized solution for implementing critical functions of modern life. Furthermore, companies and local governments are finding that the use of open source software reduces costs and allows them to pool their resources to build and maintain infrastructure software in critical niche areas. Nevertheless, the increasing reliance on open source infrastructure software introduces new demands in terms of security and safety. In principle, infrastructure software contains security features that protect against data loss, data corruption, and inadvertent transmission of data to third parties. In practice, however, these security features are compromised by a simple fact: software contains bugs.

We are developing a comprehensive solution to the problem of finding bugs in API usage in open source infrastructure software based on our experience in using the Coccinelle code matching and transformation tool, and our interactions with the Linux community. Coccinelle targets the problem of documenting and automating collateral evolutions in C code, specifically Linux code. A collateral evolution is a change that is needed in the clients of an API when the API changes in some way that affects its interface. Coccinelle provides a language for expressing collateral evolutions by means of Semantic Patches, and a transformation tool for performing them automatically. Recently, we have begun using Coccinelle to generate traditional patches for improving the safety of Linux. Some Linux developers have also begun to use the tool. Over 170 of these patches developed using Coccinelle have been integrated into the mainline Linux kernel, and more have been accepted by Linux maintainers and are pending integration. Our current work is to build on the results of Coccinelle by designing libraries of semantic patches to identify API protocols and detect violations in their usage. One of the novelty of this work is to explore how to develop these semantic patches in collaborative manner with the community of Linux open-source developers as a target. In this context, we will investigate the usage of the Telex framework for supporting collaborative developments.

## 6.5. Service management in large-scale configuration

**Participants:** Mesaac Makpangou [correspondent], Ikram Chabbouh.

Today, the vast majority of content distributed on the web are produced by the so-called web applications. Examples of web applications include e-commerce web sites, e-learning services, search engines, and emerging collaborative applications (e.g. multi-players games). There is also a continuous growing interest on service oriented architecture. These applications or services often rely on databases. Hence, the success of these applications/services depends mainly on the scalability and the performance of the database backend.

One well-established solution to improve the scalability and the performance of the database backend is to replicate the database at several locations spread out on Internet. Database replication, especially in large-scale settings, raises a number of difficult issues: which consistency model suffices to offer an acceptable functioning of the application; what internal replication management policies (i.e. number of replicas, replica life cycle, replica placement, request redirection, replica consistency management protocol) to implement in order to deliver the expected quality of service? How to accommodate the diversity of requirements due to the heterogeneity of applications?

In 2008, we focus on the design of a Replica Control Middleware for Scalable Wide Area Partially Replicated Databases. This work aims mainly to address the scalability and the consistency issues of a database be widely replicated in order to serve high loads of both read-only and update transactions submitted by clients geographically distributed over the world. We have adopted a middleware based replica control approach. There are two levels of concurrency control: Conflicts between local transactions on one edge server are handled by the local DataBase Management System. The conflicts between local and remote transactions, and the synchronisation of database replicas are handled by the replica control middleware.

The objective of our replica control middleware is to provide a *1-copy isolation* guarantee for a partially replicated relational database, while offering good response time for both read-only and update transactions. Our replica control middleware relies on global timestamps to globally schedule update transactions. Timestamps are managed by a set of cooperative schedulers associated with each replicated database. We distinguish two kinds of schedulers: *replica scheduler* and *global scheduler*.

On one hand, each replica scheduler is associated with one database replica. A replica scheduler and its associated database replica are co-located within the same edge server. Hence, there are as many replica schedulers as the number of database replicas. With respect to clients, a replica scheduler represents its associated database replica. The replica scheduler implements the interface to connect to its associated database replica, to begin transactions, to request database access operations, and to commit or to abort transactions.

On the other hand, each replicated database, as a whole, is associated with a group of global schedulers deployed on a high speed local area network. Global schedulers communicate with one another thanks to the `total-order multicast`, the `reliable multicast`, and `the reliable unicast` communication primitives offered by an underlying group communication system. We assume the support of the group membership management for the group of global schedulers. The primary role of this group of global schedulers is to certify and to schedule globally each update transaction, while preserving the so-called "first-committed-wins" conflict resolution strategy. For the sake of performance and scalability, the group of global schedulers implements a distributed fragment-based conflict detection algorithm which permits to balance the conflict detection load among the group of global schedulers: each global scheduler is responsible of detecting conflicts on a subset of fragments assigned to it. To tolerate global scheduler crashes, each fragment is assigned to a subgroup of global schedulers. In addition of enforcing a total order within certified update transactions, the group of global schedulers attached to a replicated database is also in charge of initiating the replica synchronisation protocol: upon a successful certification of an update transaction, an update notification is disseminated to all interested replicas such as to permit their synchronization.

We are now developing the first prototype of this replica control middleware in order to conduct some performance evaluation of our proposal.

## 6.6. Fault-Tolerant Partial Replication in Large-Scale Database Systems

**Participants:** Marc Shapiro [correspondent], Lamia Benmouffok, Pierre Sutra.

In distributed systems, information is replicated. Data replication enables cooperative work, improves access latency to data shared through the network, and improves availability in the presence of failures. When the information is updated, maintaining consistency between replicas is a major challenge.

Previous studies of data replication considered different areas separately, often ignoring the requirements of other areas. For instance, OS researchers often assume updates are independent; CSCW researchers ignore conflicts; algorithms research mostly ignores semantics; peer-to-peer systems often ignore mutable data and hence consistency; none of the above have addressed partial replication.

We study optimistic replication for multi-user collaborative applications such as co-operative engineering (e.g., co-operative code development), collaborative authoring (e.g., a decentralized wikipedia), or entreprise information libraries. We propose a general-purpose approach, subsuming the previous work in different areas. It takes addresses respecting application semantics, high-level operations, dependence, atomicity and conflict, long session times, etc.

In 2008, we investigated a decentralized approach to committing transactions in a replicated database, under partial replication. Previous protocols either reexecute transactions entirely and/or compute a total order of transactions. In contrast, ours applies update values, and generate a partial order between mutually conflicting transactions only. Transactions execute faster, and distributed databases commit in small committees. Both effects contribute to preserve scalability as the number of databases and transactions increase. Our algorithm ensures serializability, and is live and safe in spite of faults.

A Commutative Replicated Data Type (CRDT) is one where all concurrent operations commute. The replicas of a CRDT converge automatically, without complex concurrency control. This paper describes Treedoc, a novel CRDT design for cooperative text editing. An essential property is that the identifiers of Treedoc atoms are selected from a dense space. We discuss practical alternatives for implementing the identifier space based on an extended binary tree. We also discuss storage alternatives for data and meta-data, and mechanisms for compacting the tree. In the best case, Treedoc incurs no overhead with respect to a linear text buffer. We validate the results with traces from existing edit histories.

The work described above takes place in the context of several joint projects: Grid4All, Respire and Recall.

## 6.7. Optimistic approaches on collaborative editing

**Participant:** Marc Shapiro [correspondent].

In recent years, the Web has seen an explosive growth of massive collaboration tools, such as wiki and weblog systems. By the billions, users may share knowledge and collectively advance innovation, in various fields of science and art. Existing tools, such as the MediaWiki system for wikis, are popular in part because they do not require any specific skills. However, they are based on a centralised architecture and hence do not scale well. Moreover, they provide limited functionality for collaborative authoring of shared documents.

At the same time, peer-to-peer (P2P) techniques have grown equally explosively. They enable massive sharing of audio, video, data or any other digital content, without the need for a central server, and its attendant administration and hardware costs. P2P systems provide availability and scalability by replicating data, and by balancing workload among peers. However, current P2P networks are designed to distribute only immutable documents.

A natural research direction is to use P2P techniques to distribute collaborative documents. This raises the issue of supporting collaborative edits, and of maintaining consistency, over a massive population of users, shared documents, and sites. Within the Recall collaboration, we studied a number of alternative P2P, decentralized approaches, applied to collaborative wiki editing, contrasted with current centralized systems. Specifically, we detail P2P broadcast techniques, and we compare the existing centralized approach (MediaWiki) with several distributed, peer-to-peer approaches, namely: an operational transformation approach (MOT2), a commutativity-oriented approach (WOOTO) and a serialisation and conflict resolution approach (ACF). We evaluate each approach according to a number of specified qualitative and quantitative metrics.

A Commutative Replicated Data Type (CRDT) is one where all concurrent operations commute. The replicas of a CRDT converge automatically, without complex concurrency control. We propose Treedoc, a novel CRDT design for cooperative text editing. An essential property is that the identifiers of Treedoc atoms are selected from a dense space. We study practical alternatives for implementing the identifier space based on an extended binary tree. We also focus storage alternatives for data and meta-data, and mechanisms for compacting the tree. In the best case, Treedoc incurs no overhead with respect to a linear text buffer. We validate the results with traces from existing edit histories.

## 6.8. Fault tolerance in multi-agent systems

**Participants:** Olivier Marin [correspondent], Erika Rosas, Corentin Méhat.

Distributed agent systems stand out as a powerful tool for designing scalable software. The general outline of distributed agent software consists of computational entities which interact with one another towards a common goal that is beyond their individual capabilities

Our research focuses on middleware to deploy agent on large-scale environments and mobile networks. Our main topics of interest comprise: fault tolerance, process replication, and dynamicity with respect to both environments and applications. The ongoing research projects we are working on are all related to these topics.

The FRAME (Failure Resilient Agents in Mobile Environments) project – funded by LIP6 in 2006 and 2007 – aims at designing a middleware for the deployment of distributed algorithms among mobile devices. The originality of our approach is double: (i) we view partial and total disconnection as types of failures and aim to integrate fault tolerance solutions in order to guarantee the continuity of the computation in such a context, and (ii) we provide a modeling language which is close to Pi-calculus and yet focuses on communication channels in order to represent replicated applications and introduce failures. The ongoing PhD effort of Corentin Méhat is at the core of this project.

Our current work addresses the resiliency of group communications among mobile devices: the exchanged messages are transparently rerouted inside a structured P2P overlay – in our case Pastry – and can thus be accessed asynchronously. This has lead to a new platform design: we are presently implementing the resulting design over FreePastry in order to evaluate its performances.

The DARX (Dynamic Agent Replication eXtension) project aims at building an architecture for fault-tolerant agent computing in multi-cluster networks. The originality of our approach lies in two features: (i) an automated replication service which chooses for the application which of its computational components are to be made dependable, to which degree, and at what point of the execution, and (ii) the hierarchic architecture of the middleware which ought to provide suitable support for large-scale applications. DARX is now a component of the FACOMA project, which is supported in the context of the ANR-SETIN frame until 2009.

The latest advances include building a distributed exception-handling system which can be shared by the agent application and the dynamic replication service, and integrating heuristics on the system-level servers in order to drive the load-balancing decisions related to replicating agents.

The DDEFCON (Dependable DEployment oF Code in Open eNvironments) project addresses the safe and secure deployment of collaborative software components over large-scale networks. We seek to achieve a deployment platform that can be implemented on top of a structured peer to peer overlay. This project is funded by the LIP6 for the year 2008, and serves as a basis for the PhD these of Erika Rosas.

We are currently working on an algorithm for establishing a super-group of trusted peers, where local decisions regarding resource allocation can be taken and enacted quickly. Concurrently, we are building a service for registering resources and allowing multi-criteria searches of these resources; this service is being implemented and is awaiting proper evaluation on Grid5000.

# 7. Other Grants and Activities

## 7.1. National initiatives

### *7.1.1. SHAMAN - (2009–2011)*

Members: LIP6 (NPA), Inria Saclay (Grand-Large), Inria Bretagne (ASAP), LIP6 (Regal)

Funding: SHAMAN project is funded by ANR TELECOM

Objectives: Large-scale networks (e.g. sensor networks, peer-to-peer networks) typically include several thousands (or even hundred thousand) basic elements (computers, processors) endowed with communication capabilities (low power radio, dedicated fast network, Internet). Because of the large number of involved components, these systems are particularly vulnerable to occurrences of failures or attacks (permanent, transient, intermittent). Our focus in this project is to enable the sustainability of autonomous network functionalities in spite of component failures (lack of power, physical damage, software or environmental interference, etc.) or system evolution (changes in topology, alteration of needs or capacities). We emphasize the self-organization, fault-tolerance, and resource saving properties of the potential solutions. In this project, we will consider two different kinds of large-scale systems: on one hand sensor networks, and on the other hand peer to peer networks.

### *7.1.2. R-DISCOVER - (2009–2011)*

Members: MIS, LASMEA, GREYC, LIP6 (Regal), Thales

Funding: R-DISCOVER project is funded by ANR CONTINT

Objectives: This project considers a set of sensors and mobile robots arbitrarily deployed in a geographical area. Sensors are static. The robots can move and observe the positions of other robots and sensors in the plane and based on these observations they perform some local computations. This project addresses the problem of topological and cooperative navigation of robots in such complex systems.

### *7.1.3. SPREADS - (2008–2010)*

Members: UbiStorage, LACL, Inria Sophia, Inria (Regal)

Funding: SPREADS project is funded by ANR TELECOM

Objectives: This project proposes a collaborative research effort to study and design highly dynamic secure P2P storage systems on large scale networks like the Internet. The scientific program covered by this proposal is mainly the design of new mathematical safety, security and performance models, secure patterns, simulation to evaluate the quality of service of a peer-to-peer storage system in the context of a dynamic large scale network. These models and simulations will eventually be corroborated by experimentation on the Grid 5000 and Grid eXplorer Platforms.

### *7.1.4. Facoma - (2007–2009)*

Members: LIP6, LIRMM, Regal

Funding: Facoma project is funded by ANR SETIN

Objectives: The fault tolerance research community has developped solutions (algorithms and architectures), mostly based on the concept of replication, applied for instance to data bases. But, these techniques are almost always applied explicitely and statically. This is the responsability of the designer of the application to identify explicitly which critical servers should be made robust and also to decide which strategies (active or passive replication) and their configurations (how many replicas, their placement). Meanwhile, regarding new cooperative applications, which are very dynamic, for instance: decision support systems, distributed control, electronic commerce, crisis management systems, and intelligent sensors networks, - such applications increasingly modeled as a set of cooperative agents (multi-agent systems) -, it is very difficult, or even impossible, to identify in advance the most critical agents of the application. This is because the roles and relative importances of the agents can greatly vary during the course of computation, interaction and cooperation, the agents being able to change roles, strategies, plans, and new agents may also join or leave the application (open system). Our approach is in consequence to give the capacity to the multi-agent system itself to dynamically identify the most critical agents and to decide which abilisation strategies to apply to them.

### *7.1.5. Respire - (2005–2008)*

Members: LIP6, Atlas (IRISA), Paris (IRISA), Regal

Funding: RESPIRE project is funded by ANR (ARA MDSA)

Objectives: The Respire project aims to develop support for sharing information (including either data or meta-data) and services for managing this information in a Peer-to-Peer (P2P) environment. A P2P architecture is potentially more scalable than previous client-server approaches, but raises many interesting scientific issues. Peers are autonomous and may join or leave the network at any time. Peers publish resources for sharing, such as data or services, and may use resources published by other peers. Users may collaborate without any explicit or implicit hierarchy. We target applications that enable world-wide spread professional communities (such as a group of researchers) to collaborate, or learning scenarios. These applications manipulate heterogenous, semantically rich data. Therefore they require more advanced functionality than existing P2P file systems.

Respire is a collaboration between research teams from different areas, distributed databases and distributed systems, which have until now largely ignored each other. This synergetic approach enables each community to question hidden assumptions, and to take into account new approaches and requirements.

### *7.1.6. PlayAll - (2007–2009)*

Members: PME : Darkwoks, Atonce, Bionatics, Fandango Games, Load Inc, Kilotonn, Sixela, SpirOps, Voxler, White Birds, Wizrbox - Public : CNAM, ENST, ENJMIN, LIP6 (REgal), LIRIS

Funding: PLAYALL project is funded by Pôle de Compitivité - Cap Digital

Objectives: The goal is the build a middleware adapted to the different game platforms (Sony Play Station 3, Nitendo DS, Wii, Xbox, PC). The contribution of Regal concerns distributed algorithms taking into account QoS contraints.

### *7.1.7. Recall - (2005–2008)*

Members: LIP6, Atlas (IRISA), Paris (IRISA), Regal

Funding: Recall is funded by INRIA (Action de Recherche Coopérative)

Objectives: Recall aims to develop optimistic replication algorithms for supporting massive collaborative editing applications. The goal is to enable classical collaborative applications to scale and to tolerate faults, by deploying them above peer-to-peer networks, and without expensive hardware requirements. This project will show that P2P networks are a viable solution, not only for distributing content, but also for creating and editing it.

### *7.1.8. Fracas - (2007–2009)*

Members: ARES (Rhones-Alpes), DIONYSOS (IRISA), Grand-Large (Futurs), Regal(Paris-Rocquencourt)

Funding: Fracas is funded by INRIA (Action de Recherche Coopérative)

Objectives: We propose to define a new middleware dedicated for sensor networks. This middleware must tolerate failures and specific attacks these networks are subject.

## 7.2. European initiatives

### *7.2.1. Grant from Microsoft Research Cambridge*

Data replication enables cooperative work, improves access latency to data shared through the network, and improves availability in the presence of failures. This grant supports a doctoral student for studying consistency between replicas of mutable, semantically-rich data in a peer-to-peer fashion. This study should enable to engineer distributed systems and applications based on them, supporting cooperative applications in large-scale collaboration networks. It includes a systematic exploration of the solution space, in order to expose the cost vs. performance vs. availability vs. quality trade-offs, and understanding fault tolerance and recovery aspects. This work combines formal approaches, simulation, implementation, and measurement.

### *7.2.2. Grid4All - (2006-2009)*

Members: France Télécom Recherche et Développement, INRIA (Regal, Atlas and Grand-Large), SICS, KTH, ICCS, UPRC, UPC, Redidia.

Funding: European Commission, 6th Framework Programme, STREP (Specific Targeted Research Project)

Objectives: Grid4All embraces the vision of a “democratic” Grid as a ubiquitous utility whereby domestic users, small organizations and enterprises may draw on resources on the Internet without having to individually invest and manage computing and IT resources. This project is funded by the 6th Framework Programme of the European Commission. It involves institutional and industrial partners. Its budget is slightly over 4.8 million euros.

Grid4All has the following objectives:

- To alleviate administration and management of large scale distributed IT infrastructure, by pioneering the application of component based management architectures to self-organizing peer-to-peer overlay services.
- To provide self-management capabilities, to improve scalability, resilience to failures and volatility thus paving the way to mature solutions enabling deployment of Grids on the wide Internet.

- To widen the scope of Grid technologies by enabling on-demand creation and maintenance of dynamically evolving scalable virtual organisations even short lived.
- To apply advanced application frameworks for collaborative data sharing applications executing in dynamic environments.
- To capitalize on Grids as revenue generating sources to implement utility models of computing but using resources on the Internet.

Grid4All will help to bring global computing to the broader society beyond that of academia and large enterprises by providing an opportunity to small organisations and individuals to reap the cost benefit of resource sharing without however the burdens of management, security, and administration.

The consortium will demonstrate this by applying Grid4All in two different application domains: collaborative tools for e-learning targeting schools and digital content processing applications targeting residential users.

## 7.3. International initiatives

JAIST (Japon). With the group of Prof. Xavier Defago we investigate various aspects of self-organization and fault tolerance in the context of robots networks.

UNLV (SUA) With the group of Prof. Ajoy Datta we collaborate in designing self* solutions for the computations of connected covers of query regions in sensor networks.

Technion (Israel). We collaborate with Prof. Roy Friedman on divers aspects of dynamic systems ranging from the computation of connected covers to the design of agreement problems adequate for P2P networks.

COFECUB (Brazil). With the group of Prof. F. Greve. (Univ. Federal of Bahia), we investigate various aspects of failure detection for dynamic environement such as MANET of P2P systems.

CONYCIT (Chili). Since 2007, we start on new collaboration with the group of X. Bonnaire Fabre (Universidad Técnica Federico Santa María - Valparaiso). The main goal is to implement trusted services in P2P environment. Even if it is near impossible to fully trust a node in a P2P system, managing a set of the most trusted nodes in the system can help to implement more trusted and reliable services. Using these nodes, can reduce the probability to have some malicious nodes that will not correctly provide the given service. The project will have the following objectives: 1. To design a distributed membership algorithm for structured Peer to Peer networks in order to build a group of trusted nodes. 2. To design a maintenance algorithm to periodically clean the trusted group so as to avoid nodes whose reputation has decreased under the minimum value. 3. To provide a way for a given node X to find at least one trusted node. 4. To design a prototype of an information system, such as a news dissemination system, that relies on the trusted group.

Informal collaboration with INESC. João. Barreto, a PhD student at INESC and Instituto Superior Técnico spent six months in Projet Regal, working on ubiquitous information sharing in mobile ad-hoc networks. He took an important role in the design of our decentralised commitment algorithm .

# 8. Dissemination

## 8.1. Program committees and responsibilities

Luciana Arantes is:

- Member of the program committee of the 6ème Conférence française sur les systèmes d'exploitation, CFSE-6, Friburg, Switzerland, february 2008.

Bertil Folliot is:

- Member of the program committee of 6th International Symposium on Parallel and Distributed Computing, Hagenberg, Austria, july 2007.
- Member of the program committee of the 6ème Conférence française sur les systèmes d'exploitation, CFSE-6, Friburg, Switzerland, february 2008.
- Elected member of the Commission de spécialistes of the Paris 6 University.
- Co-chair of the middleware group of GdR ASR (Hardware, System and Network).
- Elected member of the IFIP WG10.3 working group (International Federation for Information Processing - Concurrent systems).
- Member of the « Advisory Board » of EuroPar (International European Conference on Parallel and Distributed Computing), IFIP/ACM.
- Member of the « steering committee of the International Symposium on Parallel and Distributed Computing".
- Reviewer for the IET Software Journal.

Maria Gradinariu is:

- Member of the program committee ISORC 2008 (International Symposium Object/component/service oriented real-time distributed computing)
- Member of the program committee Algotel 2008 (10eme rencontres francophones sur les aspects algorithmiques de telecommunications)

Mesaac Makpangou is:

- Member of the "Commission de spécialistes" of the University of Marne La Vallée. item Membre comité de programme: Second Workshop sur la Cohérence des Données en Univers Réparti (CDUR'O8)

Olivier Marin is:

- Member of the board of Distributed Systems Online (DSO)
- Community editor for DSO Distributed Agents

Gilles Muller is:

- Member of PC of the 7ème Conférence française sur les systèmes d'exploitation, CFSE-7, October, 2009.
- Member of PC of workshop ACP4IS 2009 http://www.aosd.net/workshops/acp4is/2009/
- Member of the jury of the best european thesis on operating systeme (eurosys) 2009.

Pierre Sens is:

- Chair of PC of the 6ème Conférence française sur les systèmes d'exploitation, CFSE-6, Friburg, Switzerland, february 2008.
- co-Chair of DAMAP 2009: Data Management in Peer-to-peer system, Workshop in conjunction EDBT, St. Perterburg, March, 2009
- Member of PC of OPODIS 2009 (International Conference On Principles Of Distributed Systems)
- Member of PC of the 7ème Conférence française sur les systèmes d'exploitation, CFSE-7, October, 2009.
- Member of PC of COLIBRI, COLloque d'Informatique: Brésil / INRIA, Coopérations, Avancées et Défis, 2009
- Member of the "Commission de spécialistes" of ENS-Cachan and Paris 11 - Orsay University.
- Elected member of the "Institut de Formation Doctorale" of Paris 6 University.
- Member of scientific council of AFNIC
- Vice-chair of the LIP6 laboratory.
- Reviewer for JPDC and TPDS journals

Marc Shapiro is:

- Member of the PC European Computer Science Summit 2008, Zürich October 2008
- Reviewer for Swiss National Science Foundation
- Reviewer for European Research Council
- Reviewer for Springer Distributed Computing
- Reviewer for Swedish Research Council (Vetenskapsrådet)
- reviewer for ICDE
- Reviewer for Transactions on Parallel and Distributed Systems
- PC chair, topic "Distributed algorithms and systems" at Europar 2008
- Organiser of Workshop on Decentralised Mechanisms for User Communities at SASO, Venice, Italy, Oct. 2008
- Member of Steering Committee of EuroSys conference
- Invited to Dagstuhl seminar "Transactional Memory: From Implementation to Application", June 2008.
- Organiser of INRIA Workshop on Massively Multiprocessor and Multicore Computers Paris, 4-5 February 2009
- Chair, EuroSys (European Chapter of ACM Sigops)
- Member, ACM Taskforce on Chapters
- Member, ACM European Taskforce
- Member, committee on “ICT scientific societies at the dawn of the 21st century” advising the European Commission Directorate General Information Society and Media.
- Invited speaker, INRIA colloquium “Le modèle et l’algorithme,” May 2008.

## 8.2. PhD reviews

Bertil Folliot was PhD rewiever of:

- Nabil Elmarzouqui. La prise de conscience dans les environnements virtuels de collaboration : Application aux interactions/manipulations distribuées collaboratives asynchrones. Thèse de Doctorat de l’Université de Franche-Comté (directeurs : Eric Garcia, Jean-Christophe Lapayre), Besançon, novembre 2008.

Gilles Muller was HDR rewiever of

- Gilles Grimaud, université de Lille, 21 novembre 2008.

Pierre Sens was PhD reviewer of:

- M. Mushtaq, Transport Adaptatif et contrôle de la qualité de services vidéo sur les réseaux pair-à-pair (Advisor: F. Krief, LABRI)
- L. Nussbaum. Contributions à lexpérimentation sur les systèmes distribués de grande taille (Advisors: O. Richard, JF. Mehaut, LIG)
- G. Le Mahec. Gestion des bases de données biologiques sur grilles de calcul (Advisor: F. Desprez, ENS-Lyon)
- Y. Busnel. Systèmes dinformation collaboratifs et auto-organisants pour réseaux de capteurs large échelle ; De la théorie à la pratique (Advisor : AM. Kermarrec, M. Bertier, IRISA)
- B. Quetier. ÉMUGRID : études des mécanismes de virtualisation pour lémulation conforme de grilles à grande échelle (Advisor: F. Cappello, LRI)
- T-M-H Nguyen. Une architecture orientée services pour la gestion de données sur grilles (Advisor: F. Magoules, Ecole Centrale Paris)
- W. Hoarau. Injection de Fautes dans les Systèmes Distribués (Advisors: F. Cappello, S. Tixeuil, LRI)
- C. Vittoria. Etudes et principes de conception dune machine langage Java : le processeur bytecode (Advisor: M. Banâtre, IRISA)

Marc Shapiro was PhD reviewer of:

- Felix Hupfeld, ZIB Berlin, Jan. 2009
- João Barreto, INESC Lisboa, Portugal, March 2009.

## 8.3. Teaching

- Bertil Folliot
  - Principles of operating systems in Licence d'Informatique, Université Paris 6
  - Distributed algorithms and systems in Master Informatique, Université Paris 6
  - Distributed systems and client/serveur in Master Informatique, Université Paris 6
  - Projects in distributed programming in Master Informatique, Université Paris 6
- Mesaac Makpangou
  - Systems and networks, Master, Pôle Universitaire Leonard de Vinci
- Oliver Marin
  - Operating system programming, Master d'Informatique, Université Paris 6
  - Operating system Principles, Licence d'Informatique, Université Paris 6
  - Parallel and distributed systems, Master d'Informatique, Université Paris 6
  - Client/server arcitecture, Licence professionelle d'Informatique, Université Paris 6
- Pierre Sens
  - Responsible of "Principles of operating systems" in Licence d'Informatique, Université Paris 6
  - Responsible of "Operating systems kernel in Master Informatique", Université Paris 6
  - Distributed systems and algorithms in Master Informatique, Université Paris 6
- Gaël Thomas
  - Responsible for the Master 1 module "Systèmes Répartis Clients/Serveurs" in Master Informatique at the Univeristy Université Paris 6
  - Responsible for the Master 2 module "Middleware Orientés Composants" in Master Informatique at the Univeristy Université Paris 6
  - Responsible for the Master 2 module "Répartition et Client/Serveur" in Master Informatique at the Univeristy Université Paris 6
  - "Noyau des Systèmes d'exploitation" in Master Informatique at the Université Paris 6
  - "Systèmes" at PolyTech' Paris

# 9. Bibliography

## Major publications by the team in recent years

[1] E. ANCEAUME, R. FRIEDMAN, M. GRADINARIU. *Managed Agreement: Generalizing two fundamental distributed agreement problems*, in "Inf. Process. Lett.", vol. 101, n$^{o}$ 5, 2007, p. 190-198.

[2] L. ARANTES, D. POITRENAUD, P. SENS, B. FOLLIOT. *The Barrier-Lock Clock: A Scalable Synchronization-Oriented Logical Clock*, in "Parallel Processing Letters", vol. 11, n$^{o}$ 1, 2001, p. 65–76.

[3] J. BEAUQUIER, M. GRADINARIU, C. JOHNEN. *Randomized self-stabilizing and space optimal leader election under arbitrary scheduler on rings*, in "Distributed Computing", vol. 20, n$^{o}$ 1, 2007, p. 75-93.

[4] M. BERTIER, L. ARANTES, P. SENS. *Distributed Mutual Exclusion Algorithms for Grid Applications: A Hierarchical Approach*, in "JPDC: Journal of Parallel and Distributed Computing", vol. 66, 2006, p. 128–144.

[5] M. BERTIER, O. MARIN, P. SENS. *Implementation and performance of an adaptable failure detecto r*, in "Proceedings of the International Conference on Dependable Systems and Networks (DSN '02)", June 2002.

[6] M. BERTIER, O. MARIN, P. SENS. *Performance Analysis of Hierarchical Failure Detector*, in "Proceedings of the International Conference on Dependable Systems and Networks (DSN '03), San-Francisco (USA)", IEEE Society Press, June 2003.

[7] J.-M. BUSCA, F. PICCONI, P. SENS. *Pastis: a Highly-Scalabel Multi-User Peer-to-Peer File Systems*, in "Euro-Par'05 - Parallel Processing, Lisboa, Portugal", Lecture Notes in Computer Science, Springer-Verlag, August 2005.

[8] A.-M. KERMARREC, A. ROWSTRON, M. SHAPIRO, P. DRUSCHEL. *The IceCube approach to the reconciliation of divergent replicas*, in "20th Symp. on Principles of Dist. Comp. (PODC), Newport RI (USA)", ACM SIGACT-SIGOPS, August 2001.

[9] N. KRISHNA, M. SHAPIRO, K. BHARGAVAN. *Brief announcement: Exploring the Consistency Problem Space*, in "Symp. on Prin. of Dist. Computing (PODC), Las Vegas, Nevada, USA", ACM SIGACT-SIGOPS, July 2005.

[10] O. MARIN, M. BERTIER, P. SENS. *DARX - A Framework For The Fault-Tolerant Support Of Agent S oftware*, in "Proceedings of the 14th IEEE International Symposium on Sofwat are Reliability Engineering (ISSRE '03), Denver (USA)", IEEE Society Press, November 2003.

[11] F. OGEL, G. THOMAS, A. GALLAND, B. FOLLIOT. *MVV : une Plate-forme à Composants Dynamiquement Reconfigurables — La Machine Virtuelle Virtuelle*, 2004.

[12] Y. PADIOLEAU, J. L. LAWALL, R. R. HANSEN, G. MULLER. *Documenting and Automating Collateral Evolutions in Linux Device Drivers*, in "EuroSys 2008, Glasgow, Scotland", March 2008, p. 247–260.

[13] Y. PADIOLEAU, J. L. LAWALL, G. MULLER. *Understanding Collateral Evolution in Linux Device Drivers*, in "The first ACM SIGOPS EuroSys conference (EuroSys 2006), Leuven, Belgium", Also available as INRIA Research Report RR-5769, April 2006, p. 59-71, http://hal.inria.fr/inria-00070251/en/.

## Year Publications

### Doctoral Dissertations and Habilitation Theses

[14] J. SOPENA. *Exclusion mutuelle répartie : tolérance aux fautes et adaptation aux cgrilles*, Ph. D. Thesis, Université Pierre et Marie Curie (Paris 6), 4, place Jussieu, Paris, december 2008.

### Articles in International Peer-Reviewed Journal

[15] Z. GUESSOUM, J.-P. BRIOT, N. FACI, O. MARIN. *Towards Reliable Multi-Agent Systems - An Adaptive Replication Mechanism*, in "Multiagent and Grid Systems", 2008.

[16] G. THOMAS, N. GEOFFRAY, C. CLÉMENT, B. FOLLIOT. *Designing highly flexible virtual machines: the JnJVM experience*, in "Software Practice Expererience", vol. 38, n$^{o}$ 15, 2008, p. 1643–1675.

### International Peer-Reviewed Conference/Proceedings

[17] M. BOUILLAGUET, L. ARANTES, P. SENS. *Fault Tolerant K-Mutual Exclusion Algorithm Using Failure Detector*, in "International Symposium on Parallel and Distributed Computing - ISPDC", 2008.

[18] D. CONAN, P. SENS, L. ARANTES, M. BOUILLAGUET. *Failure, Disconnection and Partition Detection in Mobile Environment*, in "The 7th IEEE International Symposium on Network Computing and Application - NCA", 2008, p. 119-127.

[19] A. DE LUNA ALMEIDA, S. AKNINE, J.-P. BRIOT, N. HABIBI, L. ARANTES. *Heuristiques d'allocation dynamique de ressources pour la fiabilisation des systèmes multi-agents*, in "Actes du 6ème AFRIF-AFIA congrès francophone de reconnaissance des formes et intelligence artificielle (RFIA'08), Amiens, France", 1 2008.

[20] N. GEOFFRAY, G. THOMAS, C. CLÉMENT, B. FOLLIOT. *A lazy developer approach: building a JVM with third party software*, in "PPPJ '08: Proceedings of the 6th international symposium on Principles and practice of programming in Java, New York, NY, USA", ACM, 2008, p. 73–82.

[21] N. GEOFFRAY, G. THOMAS, C. CLÉMENT, B. FOLLIOT. *Towards a new Isolation Abstraction for OSGi*, in "Proceedings of the First Workshop on Isolation and Integration in Embedded Systems (IIES 2008), Glasgow, Scotland, UK", April 2008, p. 41-45.

[22] F. HERMENIER, X. LORCA, J.-M. MENAUD, G. MULLER, J. LAWALL. *Entropy: a Consolidation Manager for Clusters*, in "the 2009 International Conference on Virtual Execution Environments (VEE'09)", To Appear, March 2009.

[23] M. NGUYEN-DUC, Z. GUESSOUM, O. MARIN, J.-F. PERROT, J.-P. BRIOT, V. DUONG. *Towards a reliable air traffic control*, in "AAMAS (Industrial Track)", 2008, p. 101-104.

[24] P. SENS, L. ARANTES, M. BOUILLAGUET, V. SIMON, F. GREVE. *An Unreliable Failure Detector for Unknown and Mobile Networks*, in "OPODIS", 2008, p. 555-559.

[25] J. SOPENA, L. ARANTES, F. LEGOND-AUBRY, P. SENS. *Composition Générique d'Algorithmes d'Exclusion Mutuelle pour les Grilles de Calcul*, in "6ème Conférence Francophone en Systèmes d'Exploitation -CFSE'6", 2008.

[26] J. SOPENA, L. ARANTES, F. LEGOND-AUBRY, P. SENS. *The Impact of Clustering on Token-Based Mutual Exclusion Algorithms*, in "Euro-Par", 2008, p. 565-575.

[27] P. SUTRA, M. SHAPIRO. *Fault-Tolerant Partial Replication in Large-Scale Database Systems*, in "Euro-Par, Las Palmas de Gran Canaria, Spain", August 2008, p. 404–413, http://pagesperso-systeme.lip6.fr/Marc.Shapiro/papers/Fault-Tolerant-Partial-Replication-in-Large-Scale-Database-Systems_Sutra_Shapiro_EuroPar-2008.pdf.

### Scientific Books (or Scientific Book chapters)

[28] X. BONNAIRE, P. SENS. *Concepts de systèmes pair-à-pair à large échelle*, in "Systèmes répartis en action : de l'embarqué aux systèmes à large échelle", F. KORDON, L. PAUTET, L. PETRUCCI (editors), n$^{\mathrm{o}}$ 11, Traité IC2 - Hermes, 2008, p. 199-221.

[29] O. MARIN, J.-M. BUSCA. *ePOST, une expérience de courrier électronique pair-à-pair*, in "Systèmes répartis en action : de l'embarqué aux systèmes à large échelle", F. KORDON, L. PAUTET, L. PETRUCCI (editors), n$^{\mathrm{o}}$ 9, Traité IC2 - Hermes, 2008, p. 257–282.

[30] E. SAINT-JAMES, G. THOMAS. *Applications pair-à-pair de partage de données*, in "Systèmes répartis en action : de l'embarqué aux systèmes à large échelle", F. KORDON, L. PAUTET, L. PETRUCCI (editors), n$^{\mathrm{o}}$ 11, Traité IC2 - Hermes, 2008, p. 223-256.

### Research Reports

[31] L. BENMOUFFOK, J.-M. BUSCA, J. MANUEL MARQUÈS, M. SHAPIRO, P. SUTRA, G. TSOUKALAS. *Telex: Principled System Support for Write-Sharing in Collaborative Applications*, Technical report, n$^{\mathrm{o}}$ 6546, INRIA, May 2008, http://hal.inria.fr/inria-00281329/en/.

[32] N. PREGUIÇA, M. SHAPIRO, J. LEGATHEAUX MARTINS. *Designing a commutative replicated data type for cooperative editing systems*, Technical report, n$^{\mathrm{o}}$ TR-02-2008 DI-FCT-UNL, Universidade Nova de Lisboa, Dep. Informática, FCT, 2008.

[33] P. SUTRA, M. SHAPIRO. *Fault-Tolerant Partial Replication in Large-Scale Database Systems*, Technical report, n$^{\mathrm{o}}$ 6440, INRIA, February 2008, http://hal.inria.fr/inria-00232662/en/.