IN PARTNERSHIP WITH:
CNRS

Université Pierre et Marie Curie (Paris 6)

Activity Report 2013

Project-Team REGAL

Large-Scale Distributed Systems and Applications

IN COLLABORATION WITH: Laboratoire d'informatique de Paris 6 (LIP6)

RESEARCH CENTER
Paris - Rocquencourt

THEME
Distributed Systems and middleware

Table of contents

Project-Team REGAL

Keywords: Distributed Algorithms, Fault Tolerance, Operating System, Data Consistency, Cloud Computing

Creation of the Project-Team: 2005 July 01.

1. Members

Research Scientists

Julia Lawall [Inria, Senior Researcher]
Mesaac Makpangou [Inria, Researcher, HdR]
Gilles Muller [Inria, Senior Researcher, HdR]
Marc Shapiro [Inria, Senior Researcher, HdR]

Faculty Members

Pierre Sens [Team leader, UPMC, Professor, HdR]
Luciana Bezerra Arantes [UPMC, Associate Professor]
Swan Dubois [UPMC, Associate Professor, from Sep. 2013]
Bertil Folliot [UPMC, Professor, HdR]
Olivier Marin [UPMC, Associate Professor]
Sébastien Monnet [UPMC, Associate Professor (on leave at Inria Rocquencourt)]
Franck Petit [UPMC, Professor, HdR]
Julien Sopena [UPMC, Associate Professor]
Gaël Thomas [UPMC, Associate Professor, HdR]

Engineers

Véronique Simon [UPMC, funded by FUI Odisea]
Harris Bakiras [Inria, until Sep. 2013]
Ikram Chabbouh [Inria, until Oct. 2013]
Christian Clausen [Inria, until Mar. 2013]
Peter Senna Tschudin [UPMC, from Feb. 2013]

PhD Students

Koutheir Attouchi [UPMC, CIFRE Orange]
Antoine Blin [UPMC, since May. 2013]
Pierpaolo Cincilla [Inria]
Rudyar Cortes [Inria, funded by CONICYT, from Nov. 2013]
Guthemberg Da Silva Silvestre [UPMC, CIFRE Orange, until Nov. 2013]
Florian David [UPMC]
Raluca Diaconu [UPMC, CIFRE Orange]
Lokesh Gidra [Inria, funded by ANR ConcoRDanT]
Lisong Guo [Inria, CORDI-S]
Ruijing Hu [UPMC, until Aug. 2013]
Mohamed-Hamza Kaaouachi [UPMC]
Mohsen Koohi-Esfahani [UPMC, until May 2013]
Jonathan Lejeune [UPMC]
Maxime Lorrillere [UPMC, funded by Nuage project]
Jean-Pierre Lozi [UPMC]
Mahsa Najafzadeh [Inria]
Julien Peeters [CEA grant until Aug. 2013, ATER UPMC since]
Karine Pires [UPMC, funded by FUI Odisea]
Masoud Saeida Ardekani [Inria, funded by ANR ConcoRDanT project]

Suman Saha [UPMC, until Mar. 2013]
Maxime Véron [CNAM]
Marek Zawirski [Inria, funded by Google Inc. and ANR ConcoRDanT]
Thomas Preud'Homme [UPMC, funded by Nuage Project until Jul. 2013]

Post-Doctoral Fellow

Tyler Crain [Inria, funded by FP7 SyncFree project, from Jul. 2013]

Visiting Scientists

Nuno Preguiça [U. Nova de Lisboa]
Rachid Guerraoui [Professor EPFL, Switzerland, Apr. 2013]
Luiz Antonio Rodrigues [PhD Student, Brazil, until Oct. 2013]
Erika Rosas Olivos [Assistant Professor, Chile, Jul. 2013]
Kenji Kono [Professor, Keio University, Japan, until Mar. 2013]

Administrative Assistant

Hélène Milome [Inria]

Others

Burcu Külahçioglu Özkan [Inria PhD Intern, Jun.–Aug. 2013]
Dang Nhan Nguyen [Euro-TM PhD Intern, Jan.–Apr. 2013]
Mudit Verma [Inria Masters' Intern, Jan.–Jul. 2013]

2. Overall Objectives

2.1. Overall Objectives

The research of the Regal team addresses the theory and practice of *Computer Systems,* including infrastructure software (operating systems and execution environments), large-scale parallel systems (multicore computers), and distributed systems (dynamic networks, P2P or cloud computing). It addresses the challenges of automated administration of highly dynamic networks, of fault tolerance, of consistency in large-scale distributed systems, of information sharing in collaborative groups, and of dynamic content distribution. It aims to design efficient, robust and flexible operating systems for multicore computers.

Regal is a joint research team between LIP6 and Inria Paris-Rocquencourt.

2.2. Highlights of the Year

- Suman Saha received the William C. Carter Award from DSN 2013 . The award recognizes an outstanding paper based on a graduate dissertation, and is the only form of best paper award given at DSN. The award was given for the paper Hector: Detecting Resource-Release Omission Faults in Error-Handling Code for Systems Software.
- Nicolas Geoffray received the 2nd prize for the best PhD thesis in Operating System, from the French Chapter of ACM SIGOPS for his thesis titled "Fostering Systems Research with Managed Runtimes".
- Inria is the leader of the new European project SyncFree, started in October 2013, described in more detail in Section 7.2.1.1. SyncFree is based on the CRDT (see Section 5.3.5) and SwiftCloud (Section 4.2) technologies, invented here. CRDTs are data types that are guaranteed to ensure eventual consistency by construction. SwiftCloud is a distributed store that leverages CRDTs to support fast and reliable updates to shared data. This European project, which involves several internet start-ups and academic partners, aims to develop cloud-scale applications that are simpler, more scalable and cheaper.

BEST PAPER AWARD :

[44] **Hector: Detecting resource-release omission faults in error-handling code for systems software in DSN 2013 - 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)**. S. SAHA, J.-P. LOZI, G. THOMAS, J. LAWALL, G. MULLER.

3. Research Program

3.1. Research rationale

As society relies more and more on computers, responsiveness, correctness and security are increasingly critical. At the same time, systems are growing larger, more parallel, and more unpredictable. Our research agenda is to design Computer Systems that remain correct and efficient despite this increased complexity and in spite of conflicting requirements. The term *"Computer Systems"* is interpreted broadly and includes systems architectures, operating systems, distributed systems, and computer networks.[1] The Regal group covers the whole spectrum, with a "bimodal" focus on distributed systems and infrastructure software. This holistic approach allows us to address related problems at different levels. It also permits us to efficiently share knowledge and expertise, and is a source of originality.

Computer Systems is a rapidly evolving domain, with strong interactions with industry. Two main evolutions in the Computer Systems area have strongly influenced our research activities:

3.1.1. Modern computer systems are increasingly parallel and distributed.

Ensuring the persistence, availability and consistency of data in a distributed setting is a major requirement: the system must remain correct despite slow networks, disconnection, crashes, failures, churn, and attacks. Ease of use, performance and efficiency are equally important for systems to be accepted. These requirements are somewhat conflicting, and there are many algorithmic and engineering trade-offs, which often depend on specific workloads or usage scenarios.

Years of research in distributed systems are now coming to fruition, and are being used by millions of users of web systems, peer-to-peer systems, gaming and social applications, or cloud computing. These new usages bring new challenges of extreme scalability and adaptation to dynamically-changing conditions, where knowledge of system state can only be partial and incomplete. The challenges of distributed computing listed above are subject to new trade-offs.

Innovative environments that motivate our research include cloud computing, geo-replication, edge clouds, peer-to-peer (P2P) systems, dynamic networks, and manycore machines. The scientific challenges are scalability, fault tolerance, security, dynamicity and the virtualization of the physical infrastructure. Algorithms designed for classical distributed systems, such as resource allocation, data storage and placement, and concurrent access to shared data, need to be revisited to work properly under the constraints of these new environments.

Regal focuses in particular on two key challenges in these areas: the adaptation of algorithms to the new dynamics of distributed systems and data management on large configurations.

3.1.2. Multicore architectures are everywhere.

The fine-grained parallelism offered by multicore architectures has the potential to open highly parallel computing to new application areas. To make this a reality, however, many issues, including issues that have previously arisen in distributed systems, need to be addressed. Challenges include obtaining a consistent view of shared resources, such as memory, and optimally distributing computations among heterogeneous architectures, such as CPUs, GPUs, and other specialized processors. As compared to distributed systems, in the case of multicore architectures, these issues arise at a more fine-grained level, leading to the need for different solutions and different cost-benefit trade-offs.

[1] As defined by the journal ACM Transactions on Computer Systems; see http://tocs.acm.org/.

Recent multicore architectures are highly diverse. Compiling and optimizing programs for such architectures can only be done for a given target. In this setting, managed runtime environments (MREs) are an elegant approach since they permit distributing a unique binary representation of an application, to which architecture-specific optimizations can be applied late on the execution machine. Finally, the concurrency provided by multicore architectures also induces new challenges for software robustness. We consider this problem in the context of systems software, using static analysis of the source code and the technology developed in the Coccinelle tool.

4. Software and Platforms

4.1. Coccinelle

Participants: Christian Clausen, Julia Lawall [correspondent], Gilles Muller [correspondent], Suman Saha, Gaël Thomas.

Coccinelle is a program matching and transformation engine which provides the language SmPL (Semantic Patch Language) for specifying desired matches and transformations in C code. Coccinelle was initially targeted towards performing collateral evolutions in Linux. Such evolutions comprise the changes that are needed in client code in response to evolutions in library APIs, and may include modifications such as renaming a function, adding a function argument whose value is somehow context-dependent, and reorganizing a data structure.

Beyond collateral evolutions, Coccinelle has been successfully used for finding and fixing bugs in systems code. One of the main recent results is an extensive study of bugs in Linux 2.6 that has permitted us to demonstrate that the quality of code has been improving over the last six years, even though the code size has more than doubled.

Coccinelle is freely available at http://coccinelle.lip6.fr under a GPL v2 license.

4.2. SwiftCloud

Participants: Mahsa Najafzadeh, Burcu Külahçioglu Özkan, Marc Shapiro [correspondent], Marek Zawirski.

Cloud computing infrastructures improve latency and provide high availability by geo-replicating data at different locations across the world. This improves user experience, which is important for services such as social networks, online shops and games. Nevertheless, the distance to the closest data centre is still far from optimal for many users.

SwiftCloud is the first system to bring geo-replication all the way to the client machine, in order to provide the best possible latency and availability. This raises two main challenges. One, how to provide, efficiently, convenient programming guarantees, including access to consistent data in read-write transactions, and ensuring session guarantees. Two, to continue providing these guarantees, despite failures that require a client to switch to a different data centre.

Our research report [61] presents the design of SwiftCloud and the algorithms it uses to achieve the desired properties, while aiming for efficiency and for scalability to large numbers of clients. Our evaluation confirms that its programming model is practical, and that its performance and fault tolerance objectives are met.

SwiftCloud is supported by the ConcoRDanT ANR project (Section 7.1.6), by a Google European Doctoral Fellowship, and by the new FP7 grant SyncFree (Section 7.2.1.1).

The code is freely available on http://gforge.inria.fr/ under a BSD license.

4.3. JESSY

Participants: Masoud Saeida Ardekani, Marc Shapiro [correspondent].

A large family of distributed transactional protocols have a common structure, called Deferred Update Replication (DUR). DUR provides dependability by replicating data, and performance by not re-executing transactions but only applying their updates. Protocols of the DUR family differ only in behaviors of few generic functions. Based on this insight, we offer a generic DUR framework, called Jessy, along with a library of finely-optimized plug-in implementations of the required behaviors. Our empirical study shows that:

1. The framework provides a fair, apples-to-apples comparison between transactional protocols;
2. By replacing plugs-ins, developers can use Jessy to understand bottlenecks in their protocols;
3. This in turn enables the improvement of existing protocols; and
4. Given a protocol, Jessy allows to evaluate the cost of ensuring various degrees of dependability.

Articles related to Jessy were published in an Inria research report [60], in the Symp. on Reliable Distr. Sys. (SRDS) [43] and in the Euro-Par conference [42] Jessy is supported by a UPMC PhD scholarship to Masoud Saeida Ardekani, and by the ConcoRDanT ANR project (Section 7.1.6).

Jessy is freely available on github under http://Github.com/msaeida/jessy under an Apache license.

4.4. Java and .Net runtimes for LLVM

Participants: Koutheir Attouchi, Harris Bakiras, Bertil Folliot, Julia Lawall, Gilles Muller, Thomas Preud'Homme, Gaël Thomas [correspondent].

Many systems research projects now target managed runtime environments (MREs) because they provide better productivity and safety compared to native environments. Still, developing and optimizing an MRE is a tedious task that requires many years of development. Although MREs share some common functionalities, such as a Just In Time Compiler or a Garbage Collector, this opportunity for sharing implementations has not been yet exploited in implementing MREs. We are working on VMKit, a first attempt to build a common substrate that eases the development and experimentation of high-level MREs and systems mechanisms.

VMKit has been used to implement a JVM and a CLI Virtual Machine (Microsoft .NET is an implementation of the CLI) using the LLVM compiler framework and the MMTk garbage collectors. The JVM, called J3, executes real-world applications such as Tomcat, Felix or Eclipse and the DaCapo benchmark. It uses the GNU Classpath project for the base classes. The CLI implementation, called N3, is its in early stages but can execute simple applications and the "pnetmark" benchmark. It uses the pnetlib project or Mono as its core library. The VMKit VMs compare in performance with industrial and top open-source VMs on CPU-intensive applications. VMKit is publicly available under the LLVM license.

http://vmkit2.gforge.inria.fr/

5. New Results

5.1. Introduction

In 2013, we focused our research on the following areas:

- *Distributed algorithms for dynamic and large networks.*
- *Management of distributed data.*
- *Performance and robustness of Systems Software in multicore architectures.*

5.2. Distributed algorithms for dynamic networks

Participants: Luciana Bezerra Arantes [correspondent], Rudyar Cortes, Guthemberg Da Silva Silvestre, Raluca Diaconu, Ruijing Hu, Anissa Lamani, Jonathan Lejeune, Olivier Marin, Sébastien Monnet, Franck Petit [correspondent], Karine Pires, Maria Potop-Butucaru, Pierre Sens, Véronique Simon, Julien Sopena.

This objective aims to design distributed algorithms adapted to new large scale or dynamic distributed systems, such as mobile networks, sensor networks, P2P systems, Grids, Cloud environments, and robot networks. Efficiency in such demanding environments requires specialised protocols, providing features such as fault or heterogeneity tolerance, scalability, quality of service, and self-stabilization. Our approach covers the whole spectrum from theory to experimentation. We design algorithms, prove them correct, implement them, and evaluate them in simulation, using OMNeT++ or PeerSim, and on large-scale real platforms such as Grid'5000. The theory ensures that our solutions are correct and whenever possible optimal; experimental evidence is necessary to show that they are relevant and practical.

Within this thread, we have considered a number of specific applications, including massively multi-player on-line games (MMOGs) and peer certification.

Since 2008, we have obtained results both on fundamental aspects of distributed algorithms and on specific emerging large-scale applications.

We study various key topics of distributed algorithms: mutual exclusion, failure detection, data dissemination and data finding in large scale systems, self-stabilization and self-* services.

5.2.1. *Mutual Exclusion and Failure Detection.*

Mutual Exclusion and Fault Tolerance are two major basic building blocks in the design of distributed systems. Most of the current mutual exclusion algorithms are not suitable for modern distributed architectures because they are not scalable, they ignore the network topology, and they do not consider application quality of service constraints. Under the ANR Project *MyCloud* and the FSE *Nu@age*, we study locking algorithms fulfilling some QoS constraints often found in Cloud Computing [46], [38].

A classical way for a distributed system to tolerate failures is to detect them and then recover. It is now well recognized that the dominant factor in system unavailability lies in the failure detection phase. Regal has worked for many years on practical and theoretical aspects of failure detections and pioneered hierarchical scalable failure detectors. [2] Since 2008, we have studied the adaptation of failure detectors to dynamic networks. In 2013, we studied Ω, the eventual leader election failure detector. Ω ensures that, eventually, each process in the system will be provided by an unique leader, elected among the set of correct processes in spite of crashes and uncertainties. It is known to be weakest failure detector to solve agreement protocols such as Paxos. Then, a number of eventual leader election protocols were suggested. Nonetheless, as far as we are aware of, no one of these protocols tolerates a free pattern of node mobility. In [27] we propose a new protocol for this scenario of dynamic and mobile unknown networks.

5.2.2. *Self-Stabilization and Self-* Services.*

We have also approached fault tolerance through self-stabilization. Self-stabilization is a versatile technique to design distributed algorithms that withstand transient faults. In particular, we have worked on the unison problem, [3] i.e., the design of self-stabilizing algorithms to synchronize a distributed clock. As part of the ANR project *SPADES*, we have proposed several snap-stabilizing algorithms for the message forwarding problem that are optimal in terms of number of required buffers. A snap-stabilizing algorithm is a self-stabilizing algorithm that stabilizes in 0 steps; in other words, such an algorithm always behaves according to its specification.

Finally, we have applied our expertise in distributed algorithms for dynamic and self-* systems in domains that at first glance seem quite far from the core expertise of the team, namely ad-hoc systems and swarms of mobile robots. In the latter, as part of ANR project *R-Discover*, we have studied various problems such as exploration and gathering.

5.2.3. *Dissemination and Data Finding in Large Scale Systems.*

In the area of large-scale P2P networks, we have studied the problems of data dissemination and overlay maintenance, i.e., maintenance of a logical network built over the a P2P network. In 2013, we have proposed

[2] Recent work by Leners et al published in SOSP 2011 uses our DSN 2003 paper as basis for performance comparison

[3] C. Boulinier, F. Petit, and V. Villain. Synchronous vs. asynchronous unison. Algorithmica, 51(1):61-80, 2008

a new distributed algorithm suitable for scale-free random topologies which model some complex real world networks [37], [52].

5.2.4. *Peer certification.*

In a distributed system, the certification of transactions makes it possible to circumscribe malicious behaviors. Certification requires the use of a trusted third party which must be centralized to guarantee safety. At a large scale, however, centralized certification represents a bottleneck and a single point of attack or failure.

We proposed two decentralized approaches towards certifying transactions with a high probability of success. The first approach replicates transactions over multiple peers and retains identical results from a qualified majority to certify that a service has been carried out for a given client at a given time [30]. The second approach uses distributed reputations to identify trusted nodes and use them as game referees to detect and prevent cheating [57].

5.3. Management of distributed data

Participants: Pierpaolo Cincilla, Guthemberg Da Silva Silvestre, Raluca Diaconu, Jonathan Lejeune, Mesaac Makpangou, Olivier Marin, Sébastien Monnet, Dang Nhan Nguyen, Burcu Külahçioglu Özkan, Karine Pires, Masoud Saeida Ardekani, Thomas Preud'Homme, Pierre Sens, Marc Shapiro, Véronique Simon, Julien Sopena, Gaël Thomas, Mathieu Valero, Mudit Verma, Marek Zawirski.

Storing and sharing information is one of the major reasons for the use of large-scale distributed computer systems. Replicating data at multiple locations ensures that the information persists despite the occurrence of faults, and improves application performance by bringing data close to its point of use, enabling parallel reads, and balancing load. This raises numerous issues:

- where to store or replicate the data, in order to ensure that it is available quickly and remains persistent despite failures and disconnections;
- how many copies are needed to face dynamically-changing demand (load) and offer (elasticity);
- how to parallelize writes and hence how to ensure consistency between replicas;
- tradeoffs between synchronised, consistent but slow updates, and fast but weakly-consistent ones;
- when and how to move data to computation, or computation to data, in order to improve response time while minimizing storage or energy usage;
- etc.

5.3.1. *Long term durability*

To tolerate failures, distributed storage systems replicate data. However, despite the replication, pieces of data may be lost (i.e. all the copies are lost). We have previously proposed a mechanism, RelaxDHT, to make distributed hash tables (DHT) resilient to high churn rates.

Well sized systems rarely loose data, still, data may be lost: the more the time passes, the greater is the risk of loss. It is thus necessary to study data durability on a long term. To do so, we have implemented an efficient simulator, we can simulate a 100 node system over years within several hours. We have observe that a given system with a given replication mechanism can store a certain amount of data above which the loss rate would be greater than an "acceptable"/fixed threshold. This amount of data can be used as a metric to compare replication strategies. We have studied the impact of the data distribution layout upon the loss rate. The way the replication mechanism distribute the data copies among the nodes has a great impact. If node contents are very correlated, the number of available sources to heal a failure is low. On the opposite, if the data copies are shuffled among the nodes, many source nodes may be available to heal the system, and thus, the system losses less pieces of data. We are also studying the impact of other parameters, like the replication degree or the way a new storer node is chosen.

5.3.2. *Adaptative replication*

Different pieces of data have different popularity: some data are stored but never accessed while other pieces are very "hot" and are requested concurrently by many clients. This implies that different pieces of data with different popularity should have a different number of copies to efficiently serve the requests without wasting resources. Furthermore, for a given piece of data, the popularity may vary drastically among time. It is thus important that the replication mechanism dynamically adapt the number of replicas to the demand. In the context of the ODISEA2 FUI project, we have made two main contributions. First, we have studied the popularity distribution and evolution of live video streams (Karine Pires thesis). Second, we have designed replication mechanisms able to gracefully adapt the replication degree to the demand, one based on bandwidth reservation, and one using statistical learning (Guthemberg Silvestre thesis).

5.3.3. *Strong consistency*

When data is updated somewhere on the network, it may become inconsistent with data elsewhere, especially in the presence of concurrent updates, network failures, and hardware or software crashes. A primitive such as consensus (or equivalently, total-order broadcast) synchronises all the network nodes, ensuring that they all observe the same updates in the same order, thus ensuring strong consistency. However the latency of consensus is very large in wide-area networks, directly impacting the response time of every update. Our contributions consist mainly of leveraging application-specific knowledge to decrease the amount of synchronisation.

To reduce the latency of consensus, we study *Generalised Consensus* algorithms, i.e., ones that leverage the commutativity of operations or the spontaneous ordering of messages by the network. We propose a novel protocol for generalised consensus that is optimal, both in message complexity and in faults tolerated, and that switches optimally between its fast path (which avoids ordering commuting requests) and its classical path (which generates a total order). Experimental evaluation shows that our algorithm is much more efficient and scales better than competing protocols.

When a database is very large, it pays off to replicate only a subset at any given node; this is known as partial replication. This allows non-overlapping transactions to proceed in parallel at different locations and decreases the overall network traffic. However, this makes it much harder to maintain consistency. We designed and implemented two *genuine* consensus protocols for partial replication, i.e., ones in which only relevant replicas participate in the commit of a transaction.

Another research direction leverages isolation levels, particularly Snapshot Isolation (SI), in order to parallelize non-conflicting transactions on databases. We prove a novel impossibility result: under standard assumptions (data store accesses are not known in advance, and transactions may access arbitrary objects in the data store), it is impossible to have both SI and GPR. Our impossibility result is based on a novel decomposition of SI which proves that, like serializability, SI is expressible on plain histories. These results are published at the Euro-Par conference [42].

We designed an efficient protocol that maintains side-steps this impossibility but maintains the most important features of SI:

1. (Genuine Partial Replication) only replicas updated by a transaction T make steps to execute T;
2. (Wait-Free Queries) a read-only transaction never waits for concurrent transactions and always commits;
3. (Minimal Commit Synchronization) two transactions synchronize with each other only if their writes conflict.

The protocol also ensures Forward Freshness, i.e., that a transaction may read object versions committed after it started.

Non-Monotonic Snapshot Isolation (NMSI) is the first strong consistency criterion to allow implementations with all four properties. We also present a practical implementation of NMSI called Jessy, which we compare experimentally against a number of well-known criteria. Our measurements show that the latency and throughput of NMSI are comparable to the weakest criterion, read-committed, and between two to fourteen times faster than well-known strong consistencies. This was published in the Symp. on Reliable Distr. Sys. (SRDS) [43].

5.3.4. Distributed Transaction Scheduling

Parallel transactions in distributed DBs incur high overhead for concurrency control and aborts. Our Gargamel system proposes an alternative approach by pre-serializing possibly conflicting transactions, and parallelizing non-conflicting update transactions to different replicas. This system provides strong transactional guarantees. In effect, Gargamel partitions the database dynamically according to the update workload. Each database replica runs sequentially, at full bandwidth; mutual synchronisation between replicas remains minimal. Our simulations show that Gargamel improves both response time and load by an order of magnitude when contention is high (highly loaded system with bounded resources), and that otherwise slow-down is negligible.

Our current experiments aim to compare the practical pros and cons of different approaches to designing large-scale replicated databases, by implementing and benchmarking a number of different protocols.

5.3.5. Eventual consistency

Eventual Consistency (EC) aims to minimize synchronisation, by weakening the consistency model. The idea is to allow updates at different nodes to proceed without any synchronisation, and to propagate the updates asynchronously, in the hope that replicas converge once all nodes have received all updates. EC was invented for mobile/disconnected computing, where communication is impossible (or prohibitively costly). EC also appears very appealing in large-scale computing environments such as P2P and cloud computing. However, its apparent simplicity is deceptive; in particular, the general EC model exposes tentative values, conflict resolution, and rollback to applications and users. Our research aims to better understand EC and to make it more accessible to developers.

We propose a new model, called *Strong Eventual Consistency* (SEC), which adds the guarantee that every update is durable and the application never observes a roll-back. SEC is ensured if all concurrent updates have a deterministic outcome. As a realization of SEC, we have also proposed the concept of a Conflict-free Replicated Data Type (CRDT). CRDTs represent a sweet spot in consistency design: they support concurrent updates, they ensure availability and fault tolerance, and they are scalable; yet they provide simple and understandable consistency guarantees.

This new model is suited to large-scale systems, such as P2P or cloud computing. For instance, we propose a “sequence” CRDT type called Treedoc that supports concurrent text editing at a large scale, e.g., for a wikipedia-style concurrent editing application. We designed a number of CRDTs such as counters (supporting concurrent increments and decrements), sets (adding and removing elements), graphs (adding and removing vertices and edges), and maps (adding, removing, and setting key-value pairs).

On the theoretical side, we identified sufficient correctness conditions for CRDTs, viz., that concurrent updates commute, or that the state is a monotonic semi-lattice. CRDTs raise challenging research issues: What is the power of CRDTs? Are the sufficient conditions necessary? How to engineer interesting data types to be CRDTs? How to garbage collect obsolete state without synchronisation, and without violating the monotonic semi-lattice requirement? What are the upper and lower bounds of CRDTs? We co-authored an innovative approach to these questions, to be published at Principles of Programming Languages (POPL) 2014 [29].

We are currently developing an extreme-scale CRDT platform called SwiftCloud; see Section 4.2.

5.3.6. Mixing commutative and non-commutative updates: reservations

Asynchronous updates are desirable because they ensure the system is available, fast and scalable. CRDTs are asynchronous, but cannot guarantee strong invariants, such as ensuring that a shared counter never goes negative. To solve this problem, we define a novel hybrid model that supports both synchronous and asynchronous updates, “red-blue-purple” consistency. The RPB model classifies updates into commutative, partially-commutative and non-commutative, and distinguishes the (global) states where partially-commutative operations can safely run asynchronously. We use reservation techniques to ensure operation in such states. A reservation promises, to a cache that holds it, that the system is in a state that allows the cache server to perform purple updates asynchronously. Reservations ensure that data is in a known state by caching both data and access permissions over data to make updates. This approach strengthens the safety guarantees in addition to eventual consistency [40].

5.4. Performance and Robustness of Systems Software in Multicore Architectures

Participants: Koutheir Attouchi, Harris Bakiras, Antoine Blin, Florian David, Bertil Folliot, Lokesh Gidra, Julia Lawall, Jean-Pierre Lozi, Gilles Muller [correspondent], Dang Nhan Nguyen, Thomas Preud'Homme, Suman Saha, Peter Senna Tschudin, Marc Shapiro, Julien Sopena, Gaël Thomas, Mudit Verma.

5.4.1. Managed Runtime Environments

Today, multicore architectures are becoming ubiquitous, found even in embedded systems, and thus it is essential that managed runtime environments can scale on multicore processors. We have found that two major scalability bottlenecks are the implementation of highly contented locks and of garbage collectors. On a multicore, a single lock can overload the bus because the cache line that contains the lock bounces between the cores, eliminating all the performance benefits from adding more cores. To address this issue, as part of the PhD of Jean-Pierre Lozi, we have developed remote core locking (RCL), in which highly contended locks are implemented on a dedicated server, minimizing bus traffic and improving application scalability. This work initially targeted C code but is now being adapted to the needs of Java applications in the PhD of Florian David. For garbage collectors, as the memory is physically distributed among a set of memory controllers, a collection saturates the bus when the collector threads access remote memory. This saturation prevents the garbage collector from scaling with the number of cores, making the garbage collector a major bottleneck of managed runtime environments on multicore hardware. As part of the PhD of Lokesh Gidra, we have identified memory placement schemes that decrease the number of remote memory accesses during a collection in OpenJDK 7, thus preventing the bottleneck caused by bus saturation [36].

5.4.2. System software robustness

A widely recognized problem in the area of finding bugs in API usage in systems code is to know what APIs are expected and to identify contexts where these expectations are not satisfied. Indeed, systems code, such as an operating systems kernel, is typically voluminous, amounting to millions of lines of code, and uses many different highly specialized APIs, making it impossible for most developers to keep the usage protocols of all of them in mind. To address this issue, we have developed an approach to inferring API function usage protocols from software, relying on knowledge of common code structures (Software – Practice and Experience [26]). Building on this experience, we have developed an approach to finding resource-release omission faults in systems code that leverages information local to a single function [44]. This approach permits finding hundreds of faults in Linux kernel code as well as a variety of other systems software, with a low rate of false positives. Finally, we have initiated an effort on understanding the range and scope of the oops reports collected in the recently revived Linux kernel oops repository [59].

Beyond finding faults in existing code, we have also considered how systems code is constructed. Specifically, in the context of Linux device drivers, we have identified the notion of a *gene*, as a sequence of code fragments that express a particular device or operating system functionality. We have performed an initial partial sequencing of the genes making up the probe functions of Linux platform drivers [45]. Relatedly, in the context of a Merlion collaboration grant with David Lo of Singapore Management University, we have considered the problem of recommending APIs to developers. We propose one approach based on the set of libraries used by other software having similar properties [47], and a second approach based on the set of libraries used to implement related feature requests [48].

5.4.3. Domain-specific languages for systems software

A challenge in the management of a datacenter is the placement of application replicas, both to avoid a single point of failure and to limit communication costs. We have proposed a novel approach, BtrPlace [23], based on the use of a domain-specific language to express constraints derived from properties of the application and of the datacenter, and the use of a constraint solver to efficiently resolve these constraints. Simulations show that BtrPlace is able to repair a configuration involving 5000 servers after a server failure in 3 minutes.

While the use of domain-specific languages such as that of BtrPlace can ease programming, it is well known that developing, and especially maintaining, a domain-specific language over time is time-consuming and challenging. This is particularly the case when the domain-specific language provides domain-specific verifications, as the code implementing these verifications has to be maintained along with the rest of the language implementation. Furthermore, new domain-specific languages typically must evolve frequently, as the language developer comes to better understand the range and scope of the domain. To address these issues, we have proposed a methodology for domain-specific language implementation development for C-like domain-specific languages [19], based on the use of rewriting rules implemented using Coccinelle. We apply this approach to our previously developed domain specific language z2z for developing network gateways, and find that the resulting language implementation is more concise and easier to extend with new language features.

6. Bilateral Contracts and Grants with Industry

6.1. Bilateral Contracts with Industry

- Metaware Technologies, 31,250 euros for the development of Coccinelle. Metaware offers software renovation services. It is using Coccinelle to modernize a large legacy C application for a client.
- Orange Lab, 90,000 euros for 3 PhD Students (CIFRE), Ralucca Diaconu, Guthemberg Da Silva Silvestre, and Koutheir Attouchi
- Renault, 60,000 over 3 years (2013 - 2016) for a CIFRE. In the context of a Cifre cooperation with Renault, we are supervising the PhD of Antoine Blin on the topic of scheduling processes on a multicore machine for the automotive industry. The goal is to allow real-time and multimedia applications to cohabit on a single processor. The challenge here is to control resource consumption of non real-time processes so as to preserve the real-time behavior of critical ones. As part of this cooperation, we will use the Bossa DSL framework for implementing process schedulers that we have previously developed.

6.2. Bilateral Grants with Industry

6.2.1. Joint PhD: CRDTs for Large-Scale Storage Systems, with Scality SA

We are starting a research project (CIFRE: industrial PhD) with the French start-up company Scality, on CRDTs for large scale storage systems.

Storage architectures for large enterprises are evolving towards a hybrid cloud model, mixing private storage (pure SSD solutions, virtualization-on-premise) with cloud-based service provider infrastructures. Users will be able to both share data through the common cloud space, and to retain replicas in local storage. In this context we need to design data structures suitable for storage, access, update and consistency of massive amounts of data at the object, block or file system level.

Current designs consider only data structures (e.g., trees or B+-Trees) that are strongly consistent and partition-tolerant (CP). However, this means that they are not available when there is a network problem, and that replicating a CP index across sites is painful. The traditional approaches include locking, journaling and replaying of logs, snapshots and Merkle trees. All of these are difficult to scale using generic approaches, although it is possible to scale them in some specific instances. For instance, synchronization in a single direction (the Active/Passive model) is relatively simple but very limited. A multi-master (Active/Active) model, where updates are allowed at multiple replicas and synchronization occurs in both directions, is difficult to achieve with the above techniques.

Our previous work has shown that many storage indexing operations commute; this enables a the highly-scalable CRDT approach. For those that do not, Red-Blue-Purple approach (Section 5.3.6) appears promising.

The objective of the joint research will be to design new algorithms for object, block and file storage systems. Note that these thee kinds of systems, although related, support different kinds of operations, and have different consistency requirements.

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. InfraJVM - (2012–2015)

Members: LIP6 (Regal), Ecole des Mines de Nanes (Constraint), IRISA (Triskell), LaBRI (LSR).

Funding: ANR Infra.

Objectives: The design of the Java Virtual Machine (JVM) was last revised in 1999, at a time when a single program running on a uniprocessor desktop machine was the norm. Today's computing environment, however, is radically different, being characterized by many different kinds of computing devices, which are often mobile and which need to interact within the context of a single application. Supporting such applications, involving multiple mutually untrusted devices, requires resource management and scheduling strategies that were not planned for in the 1999 JVM design. The goal of InfraJVM is to design strategies that can meet the needs of such applications and that provide the good performance that is required in an MRE.

The coordinator of InfraJVM is Gaël Thomas. Infra-JVM brings a grant of 202 000 euros from the ANR to UPMC over three years.

7.1.2. Nuage - (2012–2014)

Members: Non Stop Systems (NSS), Oodrive, Alphalink (Init SYS), CELESTE, DotRiver, NewGeneration, LIP6 (Regal et Phare)

Funding: Fonds National pour la Société Numérique, CDC

Objectives: The Nuage project aims at designing and building an open source, energy-aware, cloud based on OpenStack. In this project, the Regal group contributes on the storage axis. In clouds, virtualization forms the basis to ensure flexibility, portability and isolation. However, the price to pay for flexibility and isolation is memory fragmentation. We thus propose to pool unused memory by allowing nodes to use memory of other nodes to extend their cache, at the kernel level.

It involves a grant of 153 000 euros over 2,5 years.

7.1.3. ODISEA - (2011–2014)

Members: Orange, LIP6 (Regal), UbiStorage, Technicolor, Institut Telecom

Funding: FUI project, Ile de France Region

Objectives: ODISEA aims at designing new on-line data storage and data sharing solutions. Current solutions rely on big data centers, which induce many drawbacks: (i) a high cost, (ii) proprietary solutions, (iii) inefficiency (one single location, not necessarily close to the user). The goal is to tackle these issues by designing a distributed/decentralized solution that leverage edge resources like set-top boxes.

It involves a grant of 159 000 euros from Region Ile de France over three years.

7.1.4. Richelieu - (2012–2014)

Members: LIP6 (Regal), Scilab Entreprise, Silkan, OCaml Pro, Inria Saclay, Arcelor Mittal, CNES, Dassault Aviation.

Funding: FUI.

Objectives: The goal of Richelieu is to design a new runtime for the Scilab language based on VMKit. Scilab is a scientific language and its runtime relies on a costly interpretation loop. In the Richelieu project, we propose to replace the interpretation loop by VMKit, which provides both an efficient Just In Time Compiler and advanced memory management techniques.

It involves a grant of 135 000 euros from Region Ile de France over two years.

7.1.5. MyCloud (2011–2014)

Members: Inria Rhones-Alpes (SARDES), LIP6 (REGAL), EMN, WeAreCloud, Elastic Cloud.

Funding: MyCloud project is funded by ANR Arpège.

Objectives: Cloud Computing is a paradigm for enabling remote, on-demand access to a set of configurable computing resources. The objective of the MyCloud project is to define and implement a novel cloud model: SLAaaS (SLA aware Service). Novel models, control laws, distributed algorithms and languages will be proposed for automated provisioning, configuration and deployment of cloud services to meet SLA requirements, while tackling scalability and dynamics issues. It involves a grant of 155 000 euros from ANR to LIP6 over three years.

7.1.6. ConcoRDanT (2010–2014)

Members: Inria Regal, project leader; LORIA, Universdide Nova de Lisboa

Funding: ConcoRDanT is funded by ANR Blanc.

Objectives: CRDTs for consistency without concurrency control in Cloud and Peer-To-Peer systems. Massive computing systems and their applications suffer from a fundamental tension between scalability and data consistency. Avoiding the synchronisation bottleneck requires highly skilled programmers, makes applications complex and brittle, and is error-prone. The ConcoRDanT project investigates a promising new approach that is simple, scales indefinitely, and provably ensures eventual consistency. A Commutative Replicated Data Type (CRDT) is a data type where all concurrent operations commute. If all replicas execute all operations, they converge; no complex concurrency control is required. We have shown in the past that CRDTs can replace existing techniques in a number of tasks where distributed users can update concurrently, such as co-operative editing, wikis, and version control. However CRDTs are not a universal solution and raise their own issues (e.g., growth of meta-data). The ConcoRDanT project engages in a systematic and principled study of CRDTs, to discover their power and limitations, both theoretical and practical. Its outcome will be a body of knowledge about CRDTs and a library of CRDT designs, and applications using them. We are hopeful that significant distributed applications can be designed using CRDTs, a radical simplification of software, elegantly reconciling scalability and consistency. ConcoRDanT involves a grant of 192 637 euros from ANR to Inria over three and a half years.

7.1.7. STREAMS (2010–2014)

Members: LORIA (Score, Cassis), Inria (Regal, ASAP), Xwiki.

Funding: STREAMS is funded by ANR Arpège.

Objectives: Solutions for a peer-To-peer REAl-tiMe Social web The STREAMS project proposes to design peer-to-peer solutions that offer underlying services required by real-time social web applications and that eliminate the disadvantages of centralised architectures. These solutions are meant to replace a central authority-based collaboration with a distributed collaboration that offers support for decentralisation of services. The project aims to advance the state of the art on peer-to-peer networks for social and real-time applications. Scalability is generally considered as an inherent characteristic of peer-to-peer systems. It is traditionally achieved using replication techniques. Unfortunately, the current state of the art in peer-to-peer networks does not address replication of continuously updated content due to real-time user changes. Moreover, there exists a tension between sharing data with friends in a social network deployed in an open peer-to-peer network and ensuring privacy. One of the most challenging issues in social applications is how to balance collaboration with access control to shared objects. Interaction is aimed at making shared objects available to all who need them, whereas access control seeks to ensure this availability only to users with proper authorisation. STREAMS project aims at providing theoretical solutions to these challenges as well as practical experimentation. It involves a grant of 57 000 euros from ANR to Inria over three and a half years.

7.1.8. ABL - (2009–2013)

Members: Gilles Muller, Julia Lawall, Gaël Thomas, Suman Saha.

Funding: ANR Blanc.

Objectives: The goal of the “A Bug’s Life” (ABL) project is to develop a comprehensive solution to the problem of finding bugs in API usage in open source infrastructure software. The ABL project has grown out of our experience in using the Coccinelle code matching and transformation tool, which we have developed as part of the former ANR project Blanc Coccinelle, 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.

ABL concluded in 2013 with the defense of the PhD thesis of Suman Saha in March and the publication of Saha’s PhD work at the IEEE conference Dependable Systems and Networks (DSN) in June. At DSN, Saha received the William C. Carter Award for best student paper. This is the only best paper award given at DSN and was the first time that the recipient was from a French university. Saha has since taken a postdoc position jointly at Harvard and Lehigh Universities.

7.2. European Initiatives

7.2.1. FP7 Projects

7.2.1.1. SyncFree

Type: COOPERATION

Challenge: Pervasive and Trusted Network and Service Infrastructures

Instrument: Specific Targeted Research Project

Objectives: ICT-2013.1.2 “Software Engineering, Services and Cloud Computing,” ICT-2013.1.6 “Connected and Social Media”

Duration: October 2013 - September 2016

Coordinator: Marc Shapiro (Inria)

Partners: Inria (Regal & Score), Basho Technologies Inc., Trifork A/S, Rovio Entertainment Oy, U. Nova de Lisboa, U. Catholique de Louvain, Koç U., Technische U. Kaiserslautern.

Inria contact: Marc Shapiro

Abstract: The goal of SyncFree is to enable large-scale distributed applications *without global synchronisation*, by exploiting the recent concept of *Conflict-free Replicated Data Types* (CRDTs). CRDTs allow unsynchronised concurrent updates, yet ensure data consistency. This radical new approach maximises responsiveness and availability; it enables locating data near its users, in decentralised clouds.

Global-scale applications, such as virtual wallets, advertising platforms, social networks, online games, or collaboration networks, require consistency across distributed data items. As networked users, objects, devices, and sensors proliferate, the consistency issue is increasingly acute for the software industry. Current alternatives are both unsatisfactory: either to rely on synchronisation to ensure strong consistency, or to forfeit synchronisation and consistency altogether with ad-hoc eventual consistency. The former approach does not scale beyond a single data centre and is expensive. The latter is extremely difficult to understand, and remains error-prone, even for highly-skilled programmers.

SyncFree avoids both global synchronisation and the complexities of ad-hoc eventual consistency by leveraging the formal properties of CRDTs. CRDTs are designed so that unsynchronised concurrent updates do not conflict and have well-defined semantics. By combining CRDT objects from a standard library of proven datatypes (counters, sets, graphs, sequences, etc.), large-scale distributed programming is simpler and less error-prone. CRDTs are a practical and cost-effective approach.

The SyncFree project will develop both theoretical and practical understanding of large-scale synchronisation-free programming based on CRDTs. Project results will be new industrial applications, new application architectures, large-scale evaluation of both, programming models and algorithms for large-scale applications, and advanced scientific understanding.

7.2.2. Collaborations in European Programs, except FP7

Program: COST Action IC1001

Project acronym: Euro-TM

Project title: Transactional Memories: Foundations, Algorithms, Tools, and Applications

Duration: 2011–2014

Coordinator: Dr. Paolo Romano (INESC)

Other partners: Austria, Czech Republic, Denmark, France, Germany, Greece, Israel, Italy, Norway, Poland, Portugal, Serbia, Spain, Sweden, Switzerland, Turkey, United Kingdom.

Inria contact: Marc Shapiro

Abstract: Parallel programming (PP) used to be an area once confined to a few niches, such as scientific and high-performance computing applications. However, with the proliferation of multicore processors, and the emergence of new, inherently parallel and distributed deployment platforms, such as those provided by cloud computing, parallel programming has definitely become a mainstream concern. Transactional Memories (TMs) answer the need to find a better programming model for PP, capable of boosting developers' productivity and allowing ordinary programmers to unleash the power of parallel and distributed architectures avoiding the pitfalls of manual, lock based synchronization. It is therefore no surprise that TM has been subject to intense research in the last years. This Action aims at consolidating European research on this important field, by coordinating the European research groups working on the development of complementary, interdisciplinary aspects of Transactional Memories, including theoretical foundations, algorithms, hardware and operating system support, language integration and development tools, and applications.

7.2.3. Collaborations with Major European Organizations

Center for Informatics and Information Technologies (CITI) of Universidade Nova de Lisboa

Commutative Replicated Data Type (CRDT)

7.3. International Initiatives

7.3.1. Inria International Partners

7.3.1.1. Declared Inria International Partners

7.3.1.1.1. Dependability of dynamic distributed systems for ad-hoc networks and desktop grid (ONDINA) (2011-2013)

Members: Inria Paris Rocquencourt (REGAL), Inria Rhone-Alpes (Avalon), UFBA (Bahia, Brazil))

Funding: Inria

Objectives: Modern distributed systems deployed over ad-hoc networks, such as MANETs (wireless mobile ad-hoc networks), WSNs (wireless sensor networks) or Desktop Grid are inherently dynamic and the issue of designing reliable services which can cope with the high dynamics of these systems is a challenge. This project studies the necessary conditions, models and algorithms able to implement reliable services in these dynamic environments.

7.3.1.1.2. Enabling Collaborative Applications For Desktop Grids (ECADeG) (2011–2013)

Members: Inria Paris Rocquencourt (REGAL), USP (Sao Paulo, Brazil))

Funding: Inria

Objectives: The overall objective of the ECADeG research project is the design and implementation of a desktop grid middleware infrastructure for supporting the development of collaborative applications and its evaluation through a case study of a particular application in the health care domain.

7.3.2. Participation in other International Programs

7.3.2.1. Improving Clone Detection for Systems Software, Merlion Project - (2013)

Members: Julia Lawall, Gilles Muller, Lisong Guo, Peter Senna Tschudin.

Funding: Institut Français de Singapour.

Objectives: Clone detection is a technique for finding similar code fragments scattered across a code base. Clone detection is potentially very relevant to operating systems code, as many operating system services, such as drivers for related devices, have similar functionalities, and thus similar implementations. Nevertheless, the application of clone detection to systems code has achieved only moderate success, finding clone rates of only 10-20% in Linux kernel code. The purpose of this project is to consider how clone detection can be more effectively used in systems code development, for *e.g.*, code understanding or bug finding.

7.4. International Research Visitors

7.4.1. Visits of International Scientists

- Rachid Guerraoui, Professor, 1 month from EPFL, Switzerland.
- Kenji Kono, Professor, 3 months from University Keio, Japan.
- David Lo, Assistant Professor, 1 week, and Yuan Tian, PhD student, 1 month, both from Singapore Management University, in the context of a Merlion France-Singapore collaboration grant.
- Luis R. Rodriguez, 2 months, from Qualcomm, USA.

7.4.2. Internships

Participant: Dang Nhan Nguyen.

Subject: Scalable old-generation garbage collection for NUMA multicores.

Date: from Jan 2013 until Jun 2013

Institution: Chalmers U. (Sweden)

Participant: Mudit Verma.
Subject: Relaxed synchronization for library datatypes in NUMA multicores.
Date: from Jan 2013 until Jun 2013
Institution: Int. Masters in Distr. Computing / KTH (Sweden)

Participant: Burcu Külahçioglu Özkan.
Subject: Verifying distributed systems based on CRDTs
Date: from Jan 2013 until Jun 2013
Institution: Koç U., Turquie.

7.4.3. Visits to International Teams

- Julia Lawall, 2 weeks, to Singapore Management University, in the context of a Merlion France-Singapore collaboration grant.

8. Dissemination

8.1. Scientific Animation

Bertil Folliot is:

- Co-program chair du 13th IEEE International Symposium on Parallel and Distributed Computing (ISPDC 2014), Porquerolles Island, France.
- Co-editor (with Juhnyoung Lee, IBM Watson) of International Journal of Networked and Distributed Computing (IJNDC),
- PC Member of 12th International Symposium on Parallel and Distributed Computing (ISPDC 2013), Bucharest, Roumanie, June 2013.
- PC Member of The 2013 International Conference on High Performance Computing & Simulation (HPCS 2013), Helsinki, Finlande, July 2013.
- PC Member of 13th IEEE International Symposium on Parallel and Distributed Computing (ISPDC 2014), Porquerolles Island, France, June 2014.

Swan Dubois is:

- PC member of the 1st International Workshop on Models and Algorithms for Reliable and Open Computing.
- External reviewer for the 31st Symposium on Theoretical Aspects of Computer Science (STACS 2014).

Julia Lawall is:

- Member of the steering committee of Generative Programming and Component Engineering.
- Secretary of IFIP TC2.
- Member-at-large of the SIGPLAN Executive Committee.
- Member of the editorial board of Science of Computer Programming.
- Associate editor of Higher Order and Symbolic Computation.
- Expert reviewer of DAC 2013, Design Automation Conference, Austin, TX, USA, June 2013.
- PC Member of GPCE 2013, Generative Programming and Component Engineering, Indianapolis, IN, October 2013.
- PC Member of OOPSLA 2013, formerly the ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, Indianapolis, IN, USA, October 2013.
- External review committee of ASE 2013, the 28th IEEE/ACM International Conference on Automated Software Engineering, Palo Alto, CA, USA, November 2013.
- PC Member for various workshops: MISS, PLOS, Hotswup, Scheme Workshop, Real Time Linux Workshop.
- Member of IFIP WG 2.11 (Program Generation).

Gilles Muller is:

- Member of the EuroSys steering committee
- Workshop co-organizer of PLOS 2013, Nemacolin Woodlands Resort, Pennsylvania, USA, November 2013.
- PC member of DSN 2013, Budapest, Hungary, June 2013.
- PC member of EuroSys 2013, April 2013, Prague, Czech Republic.
- PC member of ICDCN 2013, January 2013, Mumbai, India.

Franck Petit is:

- Invited Editor of Journal of Theoretical Computer Sciences (TCS), Special Issue on Distributed Computing (ICDCN 2013)
- Invited Editor of Journal of Theoretical Computer Sciences (TCS), Special Issue on Stabilization, Safety, and Security (SSS 2011)
- PC Member of 14th International Conference on Distributed Computing and Networking (ICDCN 2013)
- PC Member of 22th Conférence d'informatique en Parallélisme, Architecture et Système (ComPass 2013)

Pierre Sens is:

- PC Member of 28th IEEE International Parallel & Distributed Processing Symposium (IPDPS'2014)
- PC Member of 2nd IEEE/SAE International Conference on Connected Vehicles and Expo (ICCVE 2013)
- PC Member of 42nd IEEE International Conference on Parallel Processing (ICPP'2013)
- PC Member of 27th IEEE International Parallel & Distributed Processing Symposium (IPDPS'2013)
- Project reviewer for Swiss National Science Foundation (Switzerland).
- Project reviewer for Agence Nationale de la Recherche (France).

In 2013, Marc Shapiro served as:

- Invited speaker at Int. W. on Exploiting Concurrency Efficiently and Correctly ((EC)2), http://zvonimir.info/events/ec2-2013/.
- Member of the Scientific Advisory Board of the Center for Informatics and Information Technologies (CITI) of Universidade Nova de Lisboa.
- Member of the ACM Europe Council.
- Member of the board (Conseil d'Administration) of Société Informatique de France (SIF), the French learned society in Informatics.
- Member of the board of the Euro. Forum for Info. and Comm. Sciences and Technologies (EFICST) .
- Member of the "Consolidator Grants" for the PE6 (Computer Science) selection panel of the European Research Council (ERC).
- Co-chair of Dagstuhl workshop "Consistency in Distributed Systems" [24].
- Coordinator of FP7 project SyncFree (Section 7.2.1.1).
- Coordinator of ANR project (Section 7.1.6ConcoRDanT).
- PC co-chair for Int. Conf. on Principles of Distr. Sys. (OPODIS) 2014.
- PC member for for Int. Conf. on Principles of Distr. Sys. (OPODIS) 2013.
- Review committee member of ACM Symp. on Principles and Practice of Parallel Programming (PPoPP) 2014.
- Reviewer for Int. Symp. on DIStributed Computing (DISC) 2013.
- Reviewer for ACM Symp. on Principles of Dist. Comp. (PODC) 2013.
- Project reviewer for Swiss National Science Foundation (Switzerland).
- Project reviewer for Agence Nationale de la Recherche (France).
- Project reviewer for US-Israel Binational Science Foundation.

Gaël Thomas is:

- PC Member of 7th Workshop on Programming Languages and Operating Systems (PLOS'2013)

Olivier Marin is:

- PC Member of the 1st Workshop on Real Applications for P2P Networks 2013 (RAPP'2013)

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Master: Julia Lawall, Coccinelle, 4h, niveau M2, ENSEIRB, Bordeaux.

Licence: Gaël Thomas, Introduction to the C programming language, L1, Université Paris 6

Licence: Swan Dubois, Franck Petit, C programming language, L2, Université Paris 6

Licence: Gaël Thomas, Introduction to computer architecture, L2, Université Paris 6

Licence: Luciana Arantes, Bertil Folliot, Julien Sopena, Franck Petit, Pierre Sens, Principles of operating systems, L3, Université Paris 6

Licence: Mesaac Makpangou, Client/server architecture, L3 professionelle, Université Paris 6

Licence: Sébastien Monnet, Gaël Thomas, System and Internet programmation, L2, Université Paris 6

Licence: Sébastien Monnet, Computer science initiation, L1, Université Paris 6

Master: Luciana Arantes, Sébastien Monnet, Pierre Sens, Julien Sopena, Gaël Thomas, Operating systems kernel, M1, Université Paris 6

Master: Luciana Arantes, O. Marin, Maria Potop-Butucaru, Distributed algorithms, M1, Université Paris 6

Master: Luciana Arantes, Oliver Marin, Pierre Sens, Advanced distributed algorithms, M2, Université Paris 6

Licence: Luciana Arantes, Bertil Folliot, Olivier Marin, POSIX Advanced C system programming, M1 d'Informatique, Université Paris 6

Master: Bertil Folliot, Julien Sopena, Distributed systems and client/serveur, M1 , Université Paris 6

Licence: Bertil Folliot, C programming & systems, L2, Université Paris 6

Licence: Bertil Folliot, Directed projects, L2, Université Paris 6

Licence: Bertil Folliot, Head of the Computer Courses "Applications of Computer Technology and Communication", L2, Université Paris 6

Master: Franck Petit, Resistance of Distributed Attacks, M2, Université Paris 6,

Master: Franck Petit, Embedded communicant systems, M2, UniversitéParis 6

Master: Luciana Arantes, Sébastien Monnet, Julien Sopena, Gaël Thomas, Middleware for advanced computing systems, M2, Université Paris 6"

Master: Marc Shapiro, Julien Sopena, Gaël Thomas, multicore kernels and virtualisation, M2, Université Paris 6

Master of Engineering: Marc Shapiro, OS-User, PolyTech-UPMC I4.

8.2.2. Supervision

PhD: Ruijing Hu, "Epidemic dissemination algorithms in large-scale networks: comparison and adaption to topologies", UPMC, 2/12/2013, Luciana Arantes, Isabelle Demeure, Julien Sopena, Pierre Sens.

PhD: Guthemberg Da Silva Silvestre, "Designing Adaptive Replication Schemes for Efficient Content Delivery in Edge Networks", UPMC, 18/10/2013, Sébastien Monnet, Pierre Sens.

PhD: Massata Ndiaye, “Techniques de gestion des défaillances dans les grilles informatiques tolérantes aux fautes”, UPMC, 17/09/2013, Pierre Sens.

PhD: Thomas Preud'homme, “Optimized inter-core communication for pipeline parallelism”, UPMC, 10/06/2013, Bertil Folliot, Gael Thomas, Julien Sopena.

PhD: Suman Saha, Improving the Quality of Error-Handling Code in Systems Software using Function-Local Information, UPMC, 25/03/2013, Giller Muller and Julia Lawall.

PhD: Anissa Lamani, “Algorithmes avec optimisation de ressources pour des problèmes et des systèmes distribués variés”, UPJV, 19/03/2013, Franck Petit et Vincent Villain.

8.2.3. Juries

Bertil Folliot was member of the jury of:

- Alexandre Carbon. Accélérations matérielles couplées au processeur en charge de la compilation dynamique afin d'en accroÓtre les performances. Thèse de Doctorat de l'Université Paris VI, CEA†Nano-INNOV, Palaiseau, October 2013 (président du jury).
- Ruijng Hu. Un système adaptatif de publication-abonnement pour des réseaux mobiles. Thèse de Doctorat de l'Université Paris VI, November 2013 (président du jury).
- Ridha Benosman. Conception et évaluation de performances d'un bus applicatif, parallèle et orienté services. Thèse de Doctorat du Conservatoire National des Arts et Métiers, Paris, December 2013 (président du jury).

Julia Lawall was member of the jury of:

- Tegawendé F. Bissyandé. PhD University of Bordeaux (Advisor: Laurent Réveillère)

Gilles Muller was the reviewer of:

- Sylvain Geneves, PhD U. of Grenoble (Advisor: Vivien Quema)
- Emmanuel Sifakis, PhD U. of Grenoble (Advisor: Laurent Mounier)

Gilles Muller was member of the jury of:

- Preston Rodrigues, PhD University of Bordeaux (Advisor: Laurent Réveillère)
- A MCF position at INSA of Lyon.
- Four committees for engineer positions at the Inria Paris-Rocquencourt.

Franck Petit was the reviewer of:

- Yvan Rivierre. PhD VERIMAG, Grenoble (Advisors: Florence Maraninchi, Fabienne Carrier, Stéphane Devismes)
- Sylvie Delaët, HDR LRI, Orsay.
- Hung Tran-The, PhD LIAFA, Paris (Advisors: C. Delporte-Gallet, H. Fauconnier)

Franck Petit was member if the jury of:

- Ndeye Massata Ndiaye, PhD LIP6, Paris (Advisors: Pierre Sens, Ousmane Thiare)

Pierre Sens was the reviewer of:

- Martin Quinson. HDR Loria, Nancy
- Houssem Chihoub. PhD ENS Cachan, Rennes (Advisors: Luc Bougé, Gabriel Antoniu)
- Aeiman Gadafi. PhD ENSHEIT, Toulouse (Advisor: Daniel Hagimont)
- Marko Obravac. PhD Rennes 1, Rennes (Advisor: Thierry Priol, Cedric Tedeschi)
- Mehdi Diouri. PhD ENS Lyon, Lyon (Advisor: Eddy Caron)

Gaël Thomas was the reviewer of:

- François Goicho, PhD Université de Lyon (Advisors: Stéphane Frénot, Guillaume Salagnac)
- Quentin Sabah, PhD Université de Grenoble (Advisor: Jean-Bernard Stefani)
- Konstantinos Kloudas, PhD Université de Rennes (Advisor: Anne-Marie Kermarrec)
- Geoffroy Cogniaux, PhD Université de Lilles (Advisor: Gilles Grimaud, Michael Hauspie)

8.3. Popularization

- Marc Shapiro presented a tutorial on “From strong to eventual consistency: getting it right” at OPODIS 2014.
- Marc Shapiro organized the Dagstuhl workshop on “Consistency in Distributed Systems,” in February 2013 [24].

9. Bibliography

Major publications by the team in recent years

[1] E. Anceaume, R. Friedman, M. Gradinariu Potop-Butucaru. *Managed Agreement: Generalizing two fundamental distributed agreement problems*, in "Inf. Process. Lett.", 2007, vol. 101, n^{o} 5, pp. 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", 2001, vol. 11, n^{o} 1, pp. 65–76

[3] J. Beauquier, M. Gradinariu Potop-Butucaru, C. Johnen. *Randomized self-stabilizing and space optimal leader election under arbitrary scheduler on rings*, in "Distributed Computing", 2007, vol. 20, n^{o} 1, pp. 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", 2006, vol. 66, pp. 128–144

[5] M. Bertier, O. Marin, P. Sens. *Implementation and performance of an adaptable failure detector*, 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] B. Ducourthial, S. Khalfallah, F. Petit. *Best-effort group service in dynamic networks*, in "22nd Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)", 2010, pp. 233-242

[8] 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

[9] S. Legtchenko, S. Monnet, G. Thomas. *Blue banana: resilience to avatar mobility in distributed MMOGs*, in "The 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)", July 2010

[10] J.-P. LOZI, F. DAVID, G. THOMAS, J. LAWALL, G. MULLER. *Remote Core Locking: Migrating Critical-Section Execution to Improve the Performance of Multithreaded Applications*, in "USENIX Annual Technical Conference", USENIX, June 2012, pp. 65-76

[11] 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

[12] N. PALIX, G. THOMAS, S. SAHA, C. CALVÈS, J. LAWALL, G. MULLER. *Faults in Linux: Ten Years Later*, in "Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2011)", Newport Beach, CA, USA, March 2011

[13] N. SCHIPER, P. SUTRA, F. PEDONE. *P-Store: Genuine Partial Replication in Wide Area Networks*, in "Symp. on Reliable Dist. Sys. (SRDS)", New Dehli, India, IEEE Comp. Society, October 2010, pp. 214–224

Publications of the year

Doctoral Dissertations and Habilitation Theses

[14] G. DA SILVA SILVESTRE. , *Designing Adaptive Replication Schemes for Efficient Content Delivery in Edge Networks*, Université Pierre et Marie Curie - Paris VI, October 2013, http://hal.inria.fr/tel-00931562

[15] R. HU. , *Algorithmes de dissémination épidémiques dans les réseaux à grande échelle : comparaison et adaptation aux topologies*, Université Pierre et Marie Curie - Paris VI, December 2013, avec la mention : très honorable, http://hal.inria.fr/tel-00931796

[16] N. M. NDIAYE. , *Techniques de gestion des défaillances dans les grilles informatiques tolérantes aux fautes*, Université Pierre et Marie Curie - Paris VI, September 2013, http://hal.inria.fr/tel-00931839

[17] T. PREUD'HOMME. , *Communication inter-cœurs optimisée pour le parallélisme de flux*, Université Pierre et Marie Curie - Paris VI, June 2013, http://hal.inria.fr/tel-00931833

[18] S. SAHA. , *Amélioration de la qualité des codes de gestion d'erreur dans les logiciels système en utilisant des informations locales aux fonctions*, Université Pierre et Marie Curie - Paris VI, March 2013, http://hal.inria.fr/tel-00937807

Articles in International Peer-Reviewed Journals

[19] T. F. BISSYANDÉ, L. RÉVEILLÈRE, J. LAWALL, D. BROMBERG, G. MULLER. *Implementing an Embedded Compiler using Program Transformation Rules*, in "Software: Practice and Experience", September 2013, pp. 1–20, (To appear), http://hal.inria.fr/hal-00844536

[20] A. COURNIER, S. DUBOIS, A. LAMANI, F. PETIT, V. VILLAIN. *Snap-stabilizing message forwarding algorithm on tree topologies*, in "Theoretical Computer Science", 2013, vol. 496, pp. 89-112, http://hal.inria.fr/hal-00933930

[21] S. DEVISMES, F. PETIT, S. TIXEUIL. *Optimal probabilistic ring exploration by semi-synchronous oblivious robots*, in "Theoretical Computer Science", August 2013, vol. 498, pp. 10-27 [*DOI :* 10.1016/J.TCS.2013.05.031], http://hal.inria.fr/hal-00930045

[22] Y. DIEUDONNÉ, F. LEVÉ, F. PETIT, V. VILLAIN. *Deterministic Geoleader Election in Disoriented Anonymous Systems*, in "Theoretical Computer Science", 2013, pp. 43-54, http://hal.inria.fr/hal-00933915

[23] F. HERMENIER, J. LAWALL, G. MULLER. *BtrPlace: A Flexible Consolidation Manager for Highly Available Applications*, in "IEEE Transactions on Dependable and Secure Computing", 2013, vol. 10, n° 5, pp. 273-286 [*DOI :* 10.1109/TDSC.2013.5], http://hal.inria.fr/hal-00916311

[24] B. KEMME, G. RAMALINGAM, A. SCHIPER, M. SHAPIRO, K. VASWANI. *Consistency in Distributed Systems*, in "Dagstuhl Reports", June 2013, vol. 3, n° 2, pp. 92-126 [*DOI :* 10.4230/DAGREP.3.2.92], http://hal.inria.fr/hal-00932737

[25] W. KOLBERG, P. D. B. MARCOS, J. C. S. ANJOS, A. K. S. MIYAZAKI, C. R. GEYER, L. ARANTES. *MRSG - A MapReduce Simulator over SimGrid*, in "Parallel Computing", 2013, vol. 39, n° 4-5, pp. 233–244 [*DOI :* 10.1016/J.PARCO.2013.02.001], http://hal.inria.fr/hal-00931855

[26] J. LAWALL, J. BRUNEL, N. PALIX, R. RYDHOF HANSEN, H. STUART, G. MULLER. *WYSIWIB: exploiting fine-grained program structure in a scriptable API-usage protocol-finding process*, in "Software, Practice and Experience", January 2013, vol. 43, n° 1, pp. 67-92 [*DOI :* 10.1002/SPE.2102], http://hal.inria.fr/hal-00918830

International Conferences with Proceedings

[27] L. ARANTES, F. GREVE, P. SENS, V. SIMON. *Eventual Leader Election in Evolving Mobile Networks*, in "OPODIS 2013 - 17th International Conference Principles of Distributed Systems", Nice, France, R. BALDONI, N. NISSE, M. VAN STEEN (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8304, pp. 23-37 [*DOI :* 10.1007/978-3-319-03850-6_3], http://hal.inria.fr/hal-00927651

[28] E. BAMPAS, A. LAMANI, F. PETIT, M. VALERO. *Self-Stabilizing Balancing Algorithm for Containment-Based Trees*, in "15th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS 2013", OSAKA, Japan, Springer Berlin / Heidelberg, 2013, vol. 8255, pp. 191-205, http://hal.inria.fr/hal-00933151

[29] S. BURCKHARDT, A. GOTSMAN, H. YANG, M. ZAWIRSKI. *Replicated Data Types: Specification, Verification, Optimality*, in "POPL 2014: 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages", San Diego, CA, United States, ACM, January 2014, 14 p. , http://hal.inria.fr/hal-00934311

[30] R. CORTES, X. BONNAIRE, F. KORDON, O. MARIN. *A Scalable Architecture for Highly Reliable Certification*, in "TrustCom'2013 - IEEE International Conference on Trust, Security and Privacy in Computing and Communications", Melbourne, Australia, IEEE, July 2013, pp. 328-335 [*DOI :* 10.1109/TRUSTCOM.2013.44], http://hal.inria.fr/hal-00931444

[31] R. CORTES, X. BONNAIRE, F. KORDON, O. MARIN. *Verification of a Quasi certification Protocol over a DHT*, in "Dagstuhl Seminar "Formal Verification of Distributed Algorithms"", France, April 2013, http://hal.inria.fr/hal-00931415

[32] G. DA SILVA SILVESTRE, S. MONNET. *Performing accurate simulations for deadline-aware applications*, in "HPCS 2013 - The 2013 International Conference on High Performance Computing & Simulation", Helsinki, Finland, July 2013, 10 p. , http://hal.inria.fr/hal-00861970

[33] A. DATTA, A. LAMANI, L. LARMORE, F. PETIT. *Brief Announcement: Ring Exploration by Oblivious Robots With Vision Limited to 2 or 3*, in "15th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS 2013", Osaka, Japan, Springer Berlin / Heidelberg, 2013, vol. 8255, pp. 363-366, http://hal.inria.fr/hal-00933714

[34] A. DATTA, A. LAMANI, L. LARMORE, F. PETIT. *Ring Exploration by Oblivious Agents with Local Vision*, in "33rd International Conference on Distributed Computing (ICDCS)", Philadelphia, United States, 2013, pp. 347-356, http://hal.inria.fr/hal-00933909

[35] A. DATTA, A. LAMANI, L. LARMORE, F. PETIT. *Ring Exploration with Oblivious Myopic Robots (Invited Paper)*, in "Workshop on Architecting Safety in Collaborative Mobile Systems (ASCoMS)", Toulouse, France, 2013, pp. 335-342, http://hal.inria.fr/hal-00933893

[36] L. GIDRA, G. THOMAS, J. SOPENA, M. SHAPIRO. *A study of the scalability of stop-the-world garbage collectors on multicores*, in "ASPLOS 13 - Proceedings of the eighteenth international conference on Architectural support for programming languages and operating systems", Houston, United States, ACM, March 2013, pp. 229-240 [*DOI :* 10.1145/2451116.2451142], http://hal.inria.fr/hal-00868012

[37] R. HU, J. SOPENA, L. ARANTES, P. SENS, I. DEMEURE. *Efficient Dissemination Algorithm for Scale-Free Topologies*, in "ICPP'13 - 42th International Conference on Parallel Processing", Lyon, France, IEEE Computer Society, 2013, http://hal.inria.fr/hal-00839060

[38] J. LEJEUNE, L. ARANTES, J. SOPENA, P. SENS. *A prioritized distributed mutual exclusion algorithm balancing priority inversions and response time*, in "ICPP'13 - 42th International Conference on Parallel Processing", Lyon, France, IEEE Computer Society, 2013, http://hal.inria.fr/hal-00839058

[39] S. MONNET, G. DA SILVA SILVESTRE, B. DAVID, P. SENS. *Predicting Popularity and Adapting Replication of Internet Videos for High-Quality Delivery*, in "ICPADS 2013 - 19th IEEE International Conference on Parallel and Distributed Systems", Seoul, Korea, Republic Of, December 2013, http://hal.inria.fr/hal-00930200

[40] M. NAJAFZADEH, M. SHAPIRO, V. BALEGAS, N. PREGUIÇA. *Improving the scalability of geo-replication with reservations*, in "DCC 2013 : Workshop on Distributed Cloud Computing", Dresden, Germany, IEEE Computer Society, December 2013, http://hal.inria.fr/hal-00932657

[41] D. NAVALHO, S. DUARTE, N. PREGUIÇA, M. SHAPIRO. *Incremental Stream Processing using Computational Conflict-free Replicated Data Types*, in "CloudDP '13 - 3rd International Workshop on Cloud Data and Platforms", Prague, Czech Republic, ACM, April 2013, pp. 31-36 [*DOI :* 10.1145/2460756.2460762], http://hal.inria.fr/hal-00932788

[42] M. SAEIDA ARDEKANI, P. SUTRA, M. SHAPIRO, N. PREGUIÇA. *On the Scalability of Snapshot Isolation*, in "Euro-Par 2013 - 19th International Conference Parallel Processing", Aachen (Aix-la-Chapelle), Germany, F. WOLF, B. MOHR, D. MEY (editors), LNCS - Lecture Notes in Computer Science, Springer, October 2013, vol. 8097, pp. 369-381 [*DOI :* 10.1007/978-3-642-40047-6_39], http://hal.inria.fr/hal-00932781

[43] M. SAEIDA ARDEKANI, P. SUTRA, M. SHAPIRO. *Non-Monotonic Snapshot Isolation: scalable and strong consistency for geo-replicated transactional systems*, in "SRDS 2013 -IEEE 32nd International Symposium on Reliable Distributed Systems", Braga, Portugal, IEEE Computer Society, October 2013, pp. 163-172 [*DOI :* 10.1109/SRDS.2013.25], http://hal.inria.fr/hal-00932758

[44] *Best Paper*
S. SAHA, J.-P. LOZI, G. THOMAS, J. LAWALL, G. MULLER. *Hector: Detecting resource-release omission faults in error-handling code for systems software*, in "DSN 2013 - 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)", Budapest, Hungary, IEEE Computer Society, June 2013, pp. 1-12 [*DOI :* 10.1109/DSN.2013.6575307], http://hal.inria.fr/hal-00918079.

[45] P. SENNA, L. RÉVEILLÈRE, L. JIANG, D. LO, J. LAWALL, G. MULLER. *Understanding the genetic makeup of Linux device drivers*, in "PLOS'13 - 7th Workshop on Programming Languages and Operating Systems", Nemacolin Woodlands Resort, Pennsylvania, United States, ACM, 2013 [*DOI :* 10.1145/2525528.2525536], http://hal.inria.fr/hal-00927070

[46] D. SERRANO, S. BOUCHENAK, Y. KOUKI, T. LEDOUX, J. LEJEUNE, J. SOPENA, L. ARANTES, P. SENS. *Towards QoS-Oriented SLA Guarantees for Online Cloud Services*, in "IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2013", Delft, Netherlands, May 2013, pp. 50-57, http://hal.inria.fr/hal-00780000

[47] F. THUNG, D. LO, J. LAWALL. *Automated library recommendation*, in "WCRE 2013 - 20th Working Conference on Reverse Engineering", Koblenz, Germany, R. LÄMMEL, R. OLIVETO, R. ROBBES (editors), IEEE, October 2013, pp. 182-191 [*DOI :* 10.1109/WCRE.2013.6671293], http://hal.inria.fr/hal-00918076

[48] F. THUNG, S. WANG, D. LO, J. LAWALL. *Automatic recommendation of API methods from feature requests*, in "ASE 2013 - 28th IEEE/ACM International Conference on Automated Software Engineering", Palo Alto, California, United States, E. DENNEY, T. BULTAN, A. ZELLER (editors), IEEE, November 2013, http://hal.inria.fr/hal-00918828

[49] M. VÉRON, O. MARIN, S. MONNET. *Matchmaking in multi-player on-line games: studying user traces to improve the user experience*, in "NOSSDAV 2014 - ACM Workshop on Network and Operating Systems Support for Digital Audio and Video", Singapore, March 2014, 26 p. , http://hal.inria.fr/hal-00940774

National Conferences with Proceedings

[50] F. DAVID. *Profiler dynamique de contention pour les verrous des applications Java*, in "ComPAS", Grenoble, France, January 2013, http://hal.inria.fr/hal-00937220

[51] L. GUO. *Pinpoint the Offending Code in a Kernel Oops*, in "ComPAS", Grenoble, France, January 2013, http://hal.inria.fr/hal-00937216

[52] R. HU, J. SOPENA, L. ARANTES, P. SENS, I. DEMEURE. *Comparaisons équitables des algorithmes de gossip sur les topologies aléatoires à grande-échelle*, in "ComPAS'2013 - 9ème Conférence Française sur les Systèmes d'Exploitation (CFSE'13), Chapitre français de l'ACM-SIGOPS, GDR ARP", Grenoble, France, 2013, http://hal.inria.fr/hal-00839059

[53] Y. KOUKI, T. LEDOUX, D. SERRANO, S. BOUCHENAK, J. LEJEUNE, L. ARANTES, J. SOPENA, P. SENS. *SLA et qualité de service pour le Cloud Computing*, in "Conférence d'informatique en Parallélisme, Architecture et Système, ComPAS 2013", Grenoble, France, January 2013, pp. 1-11, http://hal.inria.fr/hal-00764951

[54] J. Lejeune, L. Arantes, J. Sopena, P. Sens. *Un algorithme équitable d'exclusion mutuelle distribuée avec priorité*, in "9ème Conférence Française sur les Systèmes d'Exploitation (CFSE'13), Chapitre français de l'ACM-SIGOPS, GDR ARP", Grenoble, France, 2013, http://hal.inria.fr/hal-00839061

[55] M. Lorrillere, J. Sopena, S. Monnet, P. Sens. *Vers un cache réparti adapté au cloud computing*, in "Conférence d'informatique en Parallélisme, Architecture et Système (ComPAS'2013) - 9ème Conférence Française sur les Systèmes d'Exploitation (CFSE'13)", Grenoble, France, January 2013, pp. 1-12, http://hal.inria.fr/hal-00832976

[56] S. Saha, J.-P. Lozi. *EHCtor: Detecting resource-release omission faults in error-handling code for systems software*, in "ComPAS", Grenoble, France, January 2013, http://hal.inria.fr/hal-00937218

[57] M. Véron, O. Marin, S. Monnet, Z. Guessoum. *Vers un système d'arbitrage décentralisé pour les jeux en ligne*, in "RenPar'21 - Rencontres francophones du Parallelisme", Grenoble, France, 2013, 9 p. , http://hal.inria.fr/hal-00931400

Research Reports

[58] L. Arantes, J. Sopena. , *Easily rendering token-ring algorithms of distributed and parallel applications fault tolerant*, Inria, September 2013, n^{o} RR-8359, 23 p. , http://hal.inria.fr/hal-00859863

[59] L. Guo, P. Senna Tschudin, K. Kono, G. Muller, J. Lawall. , *Oops! What about a Million Kernel Oopses?*, Inria, June 2013, n^{o} RT-0436, 27 p. , http://hal.inria.fr/hal-00838528

[60] M. Saeida Ardekani, P. Sutra, N. Preguiça, M. Shapiro. , *Non-Monotonic Snapshot Isolation*, Inria, June 2013, n^{o} RR-7805, 45 p. , http://hal.inria.fr/hal-00643430

[61] M. Zawirski, A. Bieniusa, V. Balegas, S. Duarte, C. Baquero, M. Shapiro, N. Preguiça. , *SwiftCloud: Fault-Tolerant Geo-Replication Integrated all the Way to the Client Machine*, Inria, October 2013, n^{o} RR-8347, http://hal.inria.fr/hal-00870225

Other Publications

[62] S. Devismes, A. Lamani, F. Petit, S. Tixeuil. , *Optimal Torus Exploration by Oblivious Mobile Robots*, 2014, http://hal.inria.fr/hal-00926573