IN PARTNERSHIP WITH:
CNRS

Université Pierre et Marie Curie (Paris 6)

Activity Report 2014

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, Cloud Computing, Data Management

Creation of the Project-Team: 2005 July 01.

1. Members

Research Scientists

Mesaac Makpangou [Inria, 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]
Olivier Marin [UPMC, Associate Professor]
Sébastien Monnet [UPMC, Associate Professor]
Franck Petit [UPMC, Professor, HdR]
Julien Sopena [UPMC, Associate Professor]

Engineers

Salvatore Pileggi [Inria, since Dec 2014, funded by FP7 SyncFree]
Marek Zawirski [Inria, funded by Google]
Véronique Simon [UPMC, until Jun 2014]

PhD Students

Masoud Saeida Ardekani [Inria, until Oct 2014, funded by Google]
Jonathan Lejeune [UPMC, until Sep 2014]
Pierpaolo Cincilla [Inria, until Sep 2014]
Rudyar Cortes [UPMC]
Raluca Diaconu [UPMC, until Aug 2014]
Lokesh Gidra [UPMC]
Mohamed-Hamza Kaaouachi [UPMC]
Maxime Lorrillere [UPMC]
Mahsa Najafzadeh [Inria]
Karine Pires [UPMC, Telecom Bretagne]
Alejandro Tomsic [Inria, since Feb 2014, funded by FP7 SyncFree]
Maxime Véron [CNAM]
Vinh Tao Thanh [UPMC, since Feb 2014, CIFRE with Scality]
Bassirou Ngom [U. Cheikh Anta Diop Dakar, Sénégal, Joint PhD Student, until Nov 2014]

Post-Doctoral Fellows

Tyler Crain [Inria, funded by FP7 SyncFree]
André Leon Gradvohl [University of Campinas, Assistant professor]
Janine Kniess [Universidad do Estado de Santa Catarina - UDESC]

Visiting Scientists

Hermes Senger [Prof., U. of São Carlos, Brazil, Oct–Nov 2014]
Serdar Tasiran [Professor, Koç U., Turkey, Jul 2014–Sep 2014]

Administrative Assistant

Hélène Milome [Inria]

2. Overall Objectives

2.1. Overall Objectives

The research of the Regal team addresses the theory and practice of *Computer Systems,* including 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 scalability and consistency of distributed systems, of information sharing in collaborative groups, of dynamic content distribution, and multi- and many-core computing.

Regal is a joint research team between LIP6 and Inria Paris-Rocquencourt. In 2014, 4 permanent members of Regal created Whisper team with a focuss on infrastructure (system) software.

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,[1] and includes system architecture, operating systems, distributed systems, multiprocessor systems, and touches on related areas such as computer networks, distributed databases or support for big data. The interests of the Regal group cover the whole spectrum from theory to experimentation, with a strong focus on algorithm design and implementation.

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 and consistent access to shared data, need to be revisited to work properly under the constraints of these new environments.

[1] This follows the definition from the journal of reference in our field, ACM Transactions on Computer Systems.

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.

Of particular interest to Regal are topics related to memory management in high-end multicore computers, such as garbage collection of very large memories and system support for massive databases of highly-structured data.

4. New Software and Platforms

4.1. NumaGiC

Participants: Lokesh Gidra, Marc Shapiro, Julien Sopena [correspondent], Gaël Thomas.

NumaGiC is a version of the HotSpot garbage collector (GC) adapted to many-core computers with very large main memories. In order to maximise GC throughput, it manages the trade-off between memory locality (local scans) and parallelism (work stealing) in a self-balancing manner. Furthemore, the collector features several memory placement heuristics that improve locality. NumaGiC is described in a paper accepted for publication at ASPLOS 2015 [29].

4.2. G-DUR

Participants: Masoud Saeida Ardekani, Dastagiri Reddy Malikireddy, 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 middleware, called G-DUR, along with a library of finely-optimized plug-in implementations of the required behaviors. This paper presents the middleware, the plugins, and an extensive experimental evaluation in a geo-replicated environment. Our empirical study shows that:

1. G-DUR allows developers to implement various transactional protocols under 600 lines of code;
2. It provides a fair, apples-to-apples comparison between transactional protocols;
3. By replacing plugs-ins, developers can use G-DUR to understand bottlenecks in their protocols;
4. This in turn enables the improvement of existing protocols; and
5. Given a protocol, G-DUR helps evaluate the cost of ensuring various degrees of dependability.

G-DUR and the results of the comparison campaign are described in a paper to Middleware 2014 [33]. This research is supported in part by ConcoRDanT ANR project (Section 7.1.7) and by the FP7 grant SyncFree (Section 7.2.1.1).

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

4.3. SwiftCloud

Participants: Mahsa Najafzadeh, Marc Shapiro [correspondent], Serdar Tasiran, Marek Zawirski.

Client-side (e.g., mobile or in-browser) apps need local access to shared cloud data, but current technologies either do not provide fault-tolerant consistency guarantees, or do not scale to high numbers of unreliable and resource-poor clients, or both. Addressing this issue, the SwiftCloud distributed object database supports high numbers of client-side partial replicas. SwiftCloud offers fast reads and writes from a causally-consistent client-side cache. It is scalable, thanks to small and bounded metadata, and available, tolerating faults and intermittent connectivity by switching between data centres. The price to pay is a modest amount of staleness. A recent Inria Research Report (submitted for publication) presents the SwiftCloud algorithms, design, and experimental evaluation, which shows that client-side apps enjoy the same guarantees as a cloud data store, at a small cost.

SwiftCloud is supported by the ConcoRDanT ANR project (Section 7.1.7), by a Google Research Award, and by the FP7 grant SyncFree (Section 7.2.1.1).

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

4.4. Antidote

Participants: Tyler Crain, Marc Shapiro [correspondent], Serdar Tasiran, Alejandro Tomsic.

Antidote is the flexible cloud database platform currently under development in the SyncFree European project (Section 7.2.1.1). Antidote aims to be both a research platform for studying replication and consistency at the large scale, and an instrument for exploiting research results. The platform supports replication of CRDTs, in and between sharded (partitioned) data centres (DCs). The current stable version supports strong transactional consistency inside a DC, and causal transactional consistency between DCs. Ongoing research includes support for explicit consistency [23], for elastic version management, for adaptive replication, for partial replication, and for reconfigurable sharding.

5. New Results

5.1. Highlights of the Year

- *Garbage collection for big data on large-memory NUMA machines*. We developed NumaGiC, a high-throughput garbage collector for big-data algorithms running on large-memory NUMA machines (see Section 4.1). This result, a collaboration with the Whisper team, will be presented at ASPLOS 2015 [29].
- *Explicit consistency*. We propose an alternative approach to the strong-vs.-weak consistency conundrum, *explicit consistency*. Static analysis identifies precisely what is the minimal amount of synchronisation that is necessary to maintain the invariants required by an application (see Section 5.3.11). This result will be presented at EuroSys 2015 [53].
- *Lower bounds and optimality for CRDTs*. This is the first paper to study the inherent lower bounds of replicated data types. The contribution includes derivation of lower bounds for several data types, improvement of some implementations, and proved optimality of others (see Section 5.3.10). This result was presented at POPL 2014 [25].

5.2. Distributed algorithms for dynamic networks

Participants: Luciana Bezerra Arantes [correspondent], Rudyar Cortes, Raluca Diaconu, Jonathan Lejeune, Olivier Marin, Sébastien Monnet, Franck Petit [correspondent], Karine Pires, Pierre Sens, Véronique Simon, Julien Sopena.

Nowadays, distributed systems are more and more heterogeneous and versatile. Computing units can join, leave or move inside a global infrastructure. These features require the implementation of dynamic systems, that is to say they can cope autonomously with changes in their structure in terms of physical facilities and software. It therefore becomes necessary to define, develop, and validate distributed algorithms able to managed such dynamic and large scale systems, for instance mobile *ad hoc* networks, (mobile) sensor networks, P2P systems, Cloud environments, robot networks, to quote only a few.

Efficiency in such environments requires specialised protocols, providing features such as fault or heterogeneity tolerance, scalability, quality of service, and self-*. 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.

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. Self-Stabilization.

We have also approached fault tolerance through self-stabilization. Self-stabilization is a versatile technique to design distributed algorithms that withstand transient faults.

In [43], we proposed a silent self-stabilizing leader election algorithm (SSLE, for short) for bidirectional connected identified networks of arbitrary topology. Starting from any arbitrary configuration, SSLE converges to a terminal configuration, where all processes know the ID of the leader, this latter being the process of minimum ID. Moreover, as in most of the solutions from the literature, a distributed spanning tree rooted at the leader is defined in the terminal configuration. This algorithm is written in the locally shared memory model. It assumes the distributed unfair daemon, the most general scheduling hypothesis of the model. Our algorithm requires no global knowledge on the network (such as an upper bound on the diameter or the number of processes, for example). We showed that its stabilization time is in $\Theta(n^3)$ steps in the worst case, where n is the number of processes. Its memory requirement is asymptotically optimal, *i.e.*, $\Theta(\log n)$ bits per processes. Its round complexity is of the same order of magnitude — *i.e.*, $\Theta(n)$ rounds — as the best existing algorithm designed with similar settings. To the best of our knowledge, this was the first self-stabilizing leader election algorithm for arbitrary identified networks that is proven to achieve a stabilization time polynomial in steps. By contrast, we show that the previous best existing algorithm designed with similar settings stabilizes in a non polynomial number of steps in the worst case.

We have also implemented SSLE in a high-level simulator to empirically evaluate its average performances. Experimental results tend to show that its worst case in terms of rounds ($\Theta(3n + D)$ rounds) is rare.

5.2.2. Dynamic Distributed Systems

The first key challenge in understanding highly dynamic networks consists in developing appropriate models that are as close as possible to the phenomena that one wishes to capture. This requires the use of a formalism sufficiently expressive to formulate complex temporal properties. Recently, a vast collection of concepts, formalisms, and models has been unified in a framework called Time-Varying Graphs (TVG) [2], which are represented as time-ordered sequences of graphs defined over a fixed set of nodes. A hierarchy of classes over TVG has been described, mainly depending on properties related to connectivity and recurrence of dynamic. Such an hierarchy is an interesting tool for study computability issues. As an example, if one is able to prove an impossibility result in a class of the hierarchy with strong properties, then this impossibility result also holds in any class of the hierarchy with (strictly) weaker properties. In this context, we provide a generic framework to prove impossibility results in this model [45]. This framework helps to formally prove classical arguments about convergence of sequence of time-varying graphs used to build counter-examples. We apply this generic framework to the study of covering problems (such as minimal dominating set and maximal matching) in the context of time-varying graphs. We obtain a characterization of the weakest topology assumption that makes

[2] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, Time-varying graphs and dynamic networks, International Journal of Parallel, Emergent and Distributed Systems 27(5):387-408, 2012

these problems computable. We also propose a general time complexity measure since time-varying graph model lacks so far of such a definition.

5.2.3. Swarm of Mobile Robots

Swarm of autonomous mobile sensor devices (or, robots) recently emerged as an attractive issue in the study of dynamic distributed systems permits to assess the intrinsic difficulties of many fundamentals tasks, such as exploring or gathering in a discrete space. We consider autonomous robots that are endowed with visibility sensors (but that are otherwise unable to communicate) and motion actuators. Those robots must collaborate to solve a collective task, namely *exclusive perpetual exploration*, despite being limited with respect to input from the environment, asymmetry, memory, etc. The area to be explored is modeled as a graph and the exclusive perpetual exploration task requires every possible vertex to be visited infinitely often by every robot, with the additional constraint that no two robots may be present at the same node at the same time or may concurrently traverse the same edge of the graph.

In [28], we presented and implemented a generic method for obtaining all possible protocols for a swarm of mobile robots operating in a particular discrete space, namely an anonymous rings. Our method permits to discover new protocols that solve the problem, and to assess specific optimization criteria (such as individual coverage, visits frequency, etc.) that are met by those protocols. To our best knowledge, this was the first attempt to mechanize the discovery and fine-grained property testing of distributed mobile robot protocols.

5.3. Management of distributed data

Participants: Pierpaolo Cincilla, Raluca Diaconu, Jonathan Lejeune, Mesaac Makpangou, Olivier Marin, Sébastien Monnet, Karine Pires, Dastagiri Reddy Malikireddy, Masoud Saeida Ardekani, Pierre Sens, Marc Shapiro, Véronique Simon, Julien Sopena, Vinh Tao Thanh, Serdar Tasiran, 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, located where, 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.
- How to apply our approaches towards addressing the above issues onto a challenging use case: achieving true scalability for online games.

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.

We have observed 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/scattered among the nodes, many source nodes may be available to heal the system, and thus, the system losses less pieces of data. In order to study data durability on a long term, we have designed a model, and implemented a discrete event based simulator that can simulate a 100 node system over years within several hours. Our model, SPLAD [49] (for scattering and placing data replicas to enhance long-term durability), allows us to vary the data scattering degree by tuning a selection range width. We are also studying the impact of the policy used while choosing a storing node within the selection range (e.g., randomly, the least loaded, or smarter policies like the power of two choices). This policy has an important impact on both the storage load distribution among nodes and the number of lost pieces of data.

5.3.2. Achieving scalability for online games

Massively Multiplayer Online Games (MMOGs) such as *World of Warcraft* constitute a great use case for the management of distributed data on a large scale. Commercial support systems for MMOGs rely almost exclusively on traditional client/server architectures that are centralized. These architectures do not scale properly, both in terms of the number of players and of the number objects used to model virtual universes that grow ever more complex. Most MMOGs avoid this problem by limiting the scale of the universe: the virtual environment is partitioned into several parallel and totally disconnected worlds, such as the *Realms* in *World of Warcraft*. Each partition, handled in a centralized way, limits the number of players it can host; avatars created on different partitions will never meet in the game.

From a systems point of view, achieving true scalability raises many challenging issues for MMOGs. For instance the system must be very reactive: if the update latency on a player node is too high, the game becomes unplayable. Since these games are meant to operate on a large scale, they induce a trade-off between availability and consistency of data. The consistency aspect is critical because MMOGs incur a high degree of cheating.

Designing and implementing a scalable service for Multiplayer Online Games requires an extensive knowledge of the habits, behaviors and expectations of the players. The first part of our work on MMOGs aimed at gathering and analyzing traces of real games offers to gain insight on these matters. We collected public data from a *League of Legends* server (information over more than 56 million game sessions): the resulting database is freely available online, and an ensuing publication [34] details the analysis and conclusions we draw from this data regarding the expected requirements for a scalable MMOG service.

We steered a second part of our work on MMOGs in 2014 towards designing a peer to peer refereeing system that remains highly efficient, even on a large scale, both in terms of performance and in terms of cheat prevention. Simulations show that such a system scales easily to more than 30,000 nodes while leaving less than 0.013% occurrences of cheating undetected on a mean total of 24,819,649 refereeing queries. This work got published in the Multimedia Systems Journal [21].

Finally, we also worked on the design of a scalable architecture for online games. The goal is to balance the load among nodes to allow the simulation of a whole, contiguous, virtual space.

5.3.3. Management of dynamic big data

Managing and processing Dynamic Big Data, where multiple sources produce new data continuously, is very complex. Static cluster- or grid-based solutions are prone to induce bottleneck problems, and are therefore ill-suited in this context. Our objective in this domain is to design and implement a Reliable Large Scale Distributed Framework for the Management and Processing of Dynamic Big Data. In 2014, we focused our research on data placement and on gathering traces from target applications in order to assess our future solutions.

With respect to placement, we worked on a scheme to store and access massive streams of data efficiently. We designed a solution that extends distributed prefix tree indexing structures for this purpose. Our new maintenance protocol anticipates every data insertion on provisional child nodes and thus significantly reduces overhead and improves query response time. This work has lead to the publication of an Inria research report (RR- 8637) [46].

With respect to application traces, we targeted sport tracker applications. Designing and implementing a big data service for sport tracker applications requires an extensive knowledge of both data distribution and input load. Gathering and analysing traces from a real world sports tracker service provides insight on these matters, but such services are very protective of their data due to competition as well as privacy issues. We avoided these issues by gathering public data from a popular sports tracker server called EndoMondo. The resulting database is freely available online, and allowed an in-depth analysis from a dynamic big data perspective. This study has lead to the publication of an Inria research report (RR- 8636) [47] .

5.3.4. 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 studied the popularity distribution and evolution of live video streams [31], [36].

5.3.5. Keyword-based Indexing and Search Substract for Structured P2P Information System

Number of large scale information systems rely on a DHT-based storage infrastructure. To help users to find suitable information, one attractive solution is to maintain an index that maps keywords to suitable data. Maintaining and exploiting an index distributed towards a DHT is confronted to the performance issue. Mainly, the computation of the intersection of postings related to provided keywords could generate too large traffic over the network; also one is confronted to some unbalanced on peers’ load due to the fact that certain world are too popular!

In 2014, we propose *FreeCore*, a DHT-based distributed indexing substract that can be used to build efficient keyword-based search facilities for large scale information systems. A *FreeCore* index, considers keyword sets, then summarizes each set with a Bloom Filter. To limit the probability of false positive, we anticipate that one will use large size filters together enough hash functions. Thanks to this representation, we transform the searching problem, to the one of bitmaps matching as each query is also coded by a Bloom Filter. To distribute resulting summaries towards peers, *FreeCore* considers each summary as a sequence of binary keywords. Each binary keyword is assigned a peer and all summaries containing this binary keyword are stored at its assigned peer. Finally, to reduce the traffic overhead as well as the the size of local indices, *FreeCore* fragments each filter such as to factorize sequence of bits that occur more than once. In [40], we report the performances of the initial implementation of *FreeCore*. Thought a number of improvements were not included within this initial evaluation, *FreeCore* offers better performances than existing state of the art. Current work focusses on developping applications that exploit *FreeCore*.

5.3.6. Large-Scale File 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, the explicit consistency approach (Section 5.3.11) appears promising.

This work is part of a CIFRE agreeement with Scality (see Section 6.2.1).

5.3.7. 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.

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.

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.

An interesting side-effect of this research is an apples-to-apples comparison of many strong-consistency protocols. This work was published at LADIS 2014 [41] and at Middleware 2014 [33].

This research is supported in part by ConcoRDanT ANR project (Section 7.1.7) and by the FP7 grant SyncFree (Section 7.2.1.1).

5.3.8. 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. Both our simulations and the experimental results obtained with our prototype 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.

We have studied Gargamel's behavior while running over multiple geographically distant sites. One instance of Gargamel runs on each site, synchronizations among the different sites occur off the critical path [39]. Our experiments with the Amazon platform show that or solution can be used to support failures of whole sites.

5.3.9. 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).

CRDTs are the main topic of the ConcoRDanT ANR project (Section 7.1.7) and the FP7 grant SyncFree (Section 7.2.1.1). After developing the SwiftCloud extreme-scale CRDT platform (see Section 4.3), we are currently developing a flexible cloud database called Antidote (see Section 4.4).

5.3.10. Lower bounds and optimality of CRDTs

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, published at Principles of Programming Languages (POPL) 2014 [25]. Geographically distributed systems often rely on replicated eventually consistent data stores to achieve availability and performance. To resolve conflicting updates at different replicas, researchers and practitioners have proposed specialized consistency protocols, called replicated data types, that implement objects such as registers, counters, sets or lists. Reasoning about replicated data types has however not been on

par with comparable work on abstract data types and concurrent data types, lacking specifications, correctness proofs, and optimality results. To fill in this gap, we propose a framework for specifying replicated data types using relations over events and verifying their implementations using replication-aware simulations. We apply it to seven existing implementations of 4 data types with nontrivial conflictresolution strategies and optimizations (last-writer-wins register, counter, multi-value register and observed-remove set). We also present a novel technique for obtaining lower bounds on the worst-case space overhead of data type implementations and use it to prove optimality of four implementations. Finally, we show how to specify consistency of replicated stores with multiple objects axiomatically, in analogy to prior work on weak memory models. Overall, our work provides foundational reasoning tools to support research on replicated eventually consistent stores.

5.3.11. Explicit Consistency: Strengthening Eventual Consistency to support application invariants

The designers of the replication protocols for geo-replicated storage systems have to choose between either supporting low latency, eventually consistent operations, or supporting strong consistency for ensuring application correctness. We propose an alternative consistency model, *explicit consistency*, that strengthens eventual consistency with a guarantee to preserve specific invariants defined by the applications. Given these application-specific invariants, a system that supports explicit consistency must identify which operations are unsafe under concurrent execution, and help programmers to select either violation-avoidance or invariant-repair techniques. We show how to achieve the former while allowing most of operations to complete locally, by relying on a reservation system that moves replica coordination off the critical path of operation execution. The latter, in turn, allow operations to execute without restriction, and restore invariants by applying a repair operation to the database state. We designed and evaluated Indigo, a middleware that provides Explicit Consistency on top of a causally-consistent data store. Indigo guarantees strong application invariants while providing latency similar to an eventually consistent system.

This work was presented at W-PSDS 2014 [24] and LADIS 2014 [38]. It was selected for presentation at EuroSys 2015 [23]. This research is supported in part by the FP7 grant SyncFree (Section 7.2.1.1).

5.4. Memory management for big data

Participants: Antoine Blin, Lokesh Gidra, Sébastien Monnet, Marc Shapiro, Julien Sopena [correspondent], Gaël Thomas.

5.4.1. Garbage collection for big data on large-memory NUMA machines

On contemporary cache-coherent Non-Uniform Memory Access (ccNUMA) architectures, applications with a large memory footprint suffer from the cost of the garbage collector (GC), because, as the GC scans the reference graph, it makes many remote memory accesses, saturating the interconnect between memory nodes. We address this problem with NumaGiC, a GC with a mostly-distributed design. In order to maximise memory access locality during collection, a GC thread avoids accessing a different memory node, instead notifying a remote GC thread with a message; nonetheless, NumaGiC avoids the drawbacks of a pure distributed design, which tends to decrease parallelism. We compared NumaGiC with Parallel Scavenge and NAPS on two different ccNUMA architectures running on the Hotspot Java Virtual Machine of OpenJDK 7. On Spark and Neo4j, two industry-strength analytics applications, with heap sizes ranging from 160 GB to 350 GB, and on SPECjbb2013 and SPECjbb2005, NumaGiC improves overall performance by up to 45% over NAPS (up to 94% over Parallel Scavenge), and increases the performance of the collector itself by up to 3.6× over NAPS (up to 5.4× over Parallel Scavenge).

This research is accepted for presentation at the ASPLOS 2015 conference [29].

5.4.2. File cache pooling

Some applications, like online sales servers, intensively use disk I/Os. Their performance is tightly coupled with I/Os efficiency. To speed up I/Os, operating systems use free memory to offer caching mechanisms. Several I/O intensive applications may require a large cache to perform well. However, nowadays resources are virtualized. In clouds, for instance, virtual machines (VMs) offer both isolation and flexibility.This is the foundation of cloud elasticity, but it induces fragmentation of the physical resources, including memory. This fragmentation reduces the amount of available memory a VM can use for caching I/Os. We propose Puma [35] (for Pooling Unused Memory in Virtual Machines) which allows I/O intensive applications running on top of VMs to benefit of large caches.

This is realized by providing a remote caching mechanism that provides the ability for any VM to extend its cache using the memory of other VMs located either in the same or in a different host. Puma is a kernel level remote caching mechanism that is: (*i*) block device, file system and hypervisor agnostic; and (*ii*) efficient both locally and remotely. It can increase applications performance up to 3 times without impacting potential activity peaks.

6. Bilateral Contracts and Grants with Industry

6.1. Bilateral Contracts with Industry

- Orange Lab, 30,000 euros for 1 PhD Students (CIFRE), Ralucca Diaconu
- Renault, 60,000 over 3 years (2013 - 2016) for a CIFRE. In the context of a Cifre cooperation with Renault, we are supervising with Whipser 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 industrial PhD: CRDTs for Large-Scale Storage Systems, with Scality SA

We have started a joint CIFRE (industrial PhD) research with the French start-up company Scality, as described above (under “Large-Scale File Systems”).

The objective of this research is to design new algorithms for file and block storage systems, considering both the issues of scaling the file naming tree to a very large size, and the issue of conflicting updates to files or to the name tree, in the case of high latency or disconnected work.

6.2.2. EMR CREDIT, with Thales.

Franck Petit and Swan Dubois participate to the creation of the EMR (Equipe Mixte de Recherche) *CREDIT*, (Compréhension, Représentation et Exploitation Des Interactions Temporelles) between LIP6/UPMC and Thales.

Nowadays, networks are the field of temporal interactions that occur in many settings networks, including security issues. The amount and the speed of such interactions increases everyday. Until recently, the dynamics of these objects was little studied due to the lack of appropriate tools and methods. However, it becomes crucial to understand the dynamics of these interactions. Typically, how can we detect failures or attacks in network traffic, fraud in financial transactions, bugs or attacks traces of software execution. More generally, we seek to identify patterns in the dynamics of interactions. Recently, several different approaches have been proposed to study such interactions. For instance, by merging all interactions taking place over a period (e.g. one day) in a graph that are studied thereafter (evolving graphs). Another approach was to built meta-objects by duplicating entities at each unit of time of their activity, and by connecting them together.

The goal of the EMR is to join both teams of LIP6 and Thales on these issues. More specifically, we hope to make significant progress on security issues such as anomaly detection. This requires the use of a formalism sufficiently expressive to formulate complex temporal properties. Recently, a vast collection of concepts, formalisms, and models has been unified in a framework called Time-Varying Graphs. We want to pursuit that way. In the short run, the challenges facing us are: (1) refine the model to capture some interaction patterns, (2) design of algorithms to separate sequences of interactions, (3) Identify classes of entities playing a particular role in the dynamics, such as bridges between communities, or sources and sinks.

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. Labex SMART - (2012–2019)

Members: ISIR (UPMC/CNRS), LIP6 (UPMC/CNRS), LIB (UPMC/INSERM), LJLL (UPMC/CNRS), LTCI (Institut Mines-Télécom/CNRS), CHArt-LUTIN (Univ. Paris 8/EPHE), L2E (UPMC), STMS (IRCAM/CNRS).

Funding: Sorbonne Universités, ANR.

Description: The SMART Labex project aims globally to enhancing the quality of life in our digital societies by building the foundational bases for facilitating the inclusion of intelligent artifacts in our daily life for service and assistance. The project addresses underlying scientific questions raised by the development of Human-centered digital systems and artifacts in a comprehensive way. The research program is organized along five axes and Regal is responsible of the axe "Autonomic Distributed Environments for Mobility."

The project involves a PhD grant of 100 000 euros over 2,5 years.

7.1.2. InfraJVM - (2012–2015)

Members: LIP6 (Regal), Ecole des Mines de Nantes (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, who left the team in 2014. Infra-JVM brings a grant of 202 000 euros from the ANR to UPMC over three years.

7.1.3. 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.

7.1.4. 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 large 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.

7.1.5. 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.

7.1.6. 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.

7.1.7. ConcoRDanT (2010–2014)

Members: Inria Regal, project leader; LORIA, Université de Nantes, Universdade 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.

7.1.8. 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.

7.2. European Initiatives

7.2.1. FP7 & H2020 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 & H2020

Program: COST Action IC1001

Project acronym: Euro-TM

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

Duration: 2011–2015

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

Ecole Polytechnique Fédérale de Lausanne, Distributed Programming Laboratory (Switzerland)

Caracterization of the weakest failure detector for eventual consistency

7.3. International Initiatives

7.3.1. Inria Associate Teams

7.3.1.1. ARMADA

Title: hARnessing MAssive DAta flows

International Partner (Institution - Laboratory - Researcher):

Universidad Tecnica Federico Santa Maria (CHILI)

Duration: 2014 - 2016

See also: http://web.inria-armada.org

The ARMADA project aims at designing and implementing a reliable framework for the management and processing of massive dynamic dataflows. The project is two-pronged: fault-tolerant middleware support for processing massive continuous input, and a redundant storage service for mutable data on a massive scale.

7.3.2. Inria International Partners

7.3.2.1. Declared Inria International Partners

7.3.2.1.1. PHC MAIMONIDE

Title: Application Dependent Intrusion (Byzantine) Detection in Dynamic Cloud Systems

International Partner (Institution - Laboratory - Researcher):

Technion, Haifa (Israel)

Duration: 2014 – 2015

The goal of this project is to study the ability to detect intrusions, and more broadly Byzantine failures, in standard cloud services. The goal is to provide a formal model and a corresponding formal definition of Byzantine failure detection in dynamic cloud environments, and provide formally provable implementations of these detectors. We also intend to study how to combine such Byzantine failure detectors in standard open source cloud building blocks, such as ZooKeeper, Hadoop, and Cassandra, and harden them in order to make them resilient to such attacks.

7.3.3. Participation In other International Programs

Luciana Arantes and Olivier Marin participated to the STIC-AmSud project RESPOND, which ended with a workshop in Punta Arenas, Chile, from November 17th to November 21st, 2014

7.4. International Research Visitors

7.4.1. Visits of International Scientists

Serdar Tasiran

Date: 07/2014 – 09/2014

Institution: Koç University (Turkey)

Anubis Graciela de Moraes Rossetto

Date: 03/2014 – 05/2014

Institution: Federal University of Rio Grande do Sul Porto Alegre (Brazil)

Vivien Quéma

Date: 01/2014 – 08/2014

Institution: LIG (FRANCE)

7.4.1.1. Internships

Dastagiri Reddy Malikireddy

Date: May–Aug 2014

Institution: IIT Kharagpur, India.

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific event organisation

8.1.1.1. General chair, scientific chair

- Pierre Sens was the general co-chair for SBAC-PAD 2014, 26th International Symposium on Computer Architecture and High Performance Computing.
- Marc Shapiro was Program Co-Chair of the OPODIS 2014 conference.

8.1.1.2. Member of the organizing committee

Sébastien Monnet was the workshop chair for SBAC-PAD 2014, 26th International Symposium on Computer Architecture and High Performance Computing. Luciana Arantes was local organizer of SBAC-PAD 2014

8.1.2. Scientific events selection

8.1.2.1. Chair of conference program committee

- Luciana Arantes was PC chair of the XV Workshop on Test and Fault Tolerance, 2014, Brazil.
- Marc Shapiro was Program Co-Chair of the OPODIS 2014 conference.

8.1.2.2. Member of the conference program committee

- Swan Dubois was member of the programm committee of the 16èmes Rencontres Francophones pour les Aspects Algorithmiques des Télécommunications (AlgoTel'14).
- Sébastien Monnet was member of the programm committee of the 1st International Symposium on Cloud Computing (CCA 2014).
- Franck Petit was member of the program committee of the Conférence d'informatique en Parallélisme, Architecture et Système (ComPas) 2014.
- Pierre Sens was member of the program committee of 3rd IEEE/SAE International Conference on Connected Vehicles and Expo (ICCVE 2014), 3rd IEEE Symposium on Network Cloud Computing and Applications (NCCA'2014), 28th IEEE International Parallel and Distributed Processing Symposium (IPDPS'2014).
- Luciana Arantes was member of the program committee The 13th IEEE International Symposium on Network Computing and Applications (NCA 2014).
- Marc Shapiro was member of the PC of the Annual ACM/IFIP/USENIX Middleware Conference (Middleware 2014), and of the Workshop on Adaptive Resource Management and Scheduling for Cloud Computing (AMRS-CC) 2014.

8.1.2.3. Reviewer

Swan Dubois peer-reviewed papers for STACS'14, ICDCS'14, SRDS'14, SSS'14, and WWW'14.

Franck Petit peer-reviewed papers for ICDCN'15, IPDPS'15 and SSS'14.

Marc Shapiro was a reviewer for the European Research Council Consolidator Grant (panel PE6) 2014.

8.1.3. Journal

8.1.3.1. Member of the editorial board

Franck Petit is a member of the editorial board of Journal of Discrete Mathematics and The Scientific World Journal. Pierre Sens is associated editor of International Journal of High Performance Computing and Networking (IJHPCN).

8.1.3.2. Reviewer

Swan Dubois peer-reviewed papers for Information Processing Letters (IPL) and Scientific World Journal (SWJ). Sébastien Monnet peer-reviewed papers for TCS, JPDC and Transaction on Computers journal. Franck Petit peer-reviewed papers for TCS, Distributed Computing, International Journal of Foundations of Computer Science. Luciana Arantes reviewed papers for JPDC and FGCS.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Master: Luciana Arantes, Swan Dubois, Oliver Marin, Sébastien Monnet, Franck Petit, Pierre Sens, Advanced distributed algorithms, M2, UPMC Sorbonne Universités, France

Master: Luciana Arantes, Sébastien Monnet, Pierre Sens, Julien Sopena, Operating systems kernel, M1, UPMC Sorbonne Universités, France

Master: Luciana Arantes, Olivier Marin, System distributed Programming, M1, UPMC Sorbonne Universités, France

Master: Luciana Arantes, Olivier Marin, Franck Petit, Distributed Algorithms, M1, UPMC Sorbonne Universités, France

Licence: Pierre Sens, Luciana Arantes, Julien Sopena, Principles of operating systems, L3, UPMC Sorbonne Universités, France

Licence: Swan Dubois, Sébastien Monnet, Introduction to operating systems, L2, UPMC Sorbonne Universités, France

Licence : Swan Dubois, Data structures in C, 80h, L2, UPMC Sorbonne Universités, France *

Licence : Swan Dubois, Data structures in C, L2, UPMC Sorbonne Universités, France

Ingénieur 4ème année : Marc Shapiro, Introduction aux systèmes d'exploitation, 22 h, M1, Polytech UPMC Sorbonne Universités, France.

8.2.2. Supervision

Completed PhD: Pierpaolo Cincilla: "Gargamel: boosting DBMS performance by parallelising write transactions," UPMC, 09/15/2014. Advisors: Marc Shapiro, Sébastien Monnet.

PhD: Jonathan Lejeune, "Algorithmique distribuée d'exclusion mutuelle : vers une gestion efficace des ressources," UPMC, 09/19/2014, Julien Sopena, Luciana Arantes, Pierre Sens.

Completed PhD: Masoud Saeida Ardekani, "Ensuring Consistency in Partially Replicated Data Stores," UPMC, 09/16/2014. Advisors: Marc Shapiro, Pierre Sutra.

PhD: Yoann Péron, "Development of an adaptive recommendation system", UPMC/Makazi, Franck Petit, Patrick Gallinari, Matthias Oehler (Makazi).

PhD: Mohamed Hamza Kaaouachi, "Autonomic Distributed Environments for Mobility", UPMC/Chart-LUTIN (Labex SMART), Franck Petit, Swan Dubois, and François Jouen (Chart).

PhD in progress: Tao Thanh Vinh, UPMC, CIFRE, since Feb. 2014. Advisors: Marc Shapiro, Vianney Rancurel (Scality).

PhD in progress: Alejandro Z. Tomsic, UPMC, funded by SyncFree, since Feb. 2014. Advisor: Marc Shapiro.

PhD in progress: Mahsa Najafzadeh, UPMC, funded by Inria competitive grant (Cordi-S), since Nov. 2012. Advisor: Marc Shapiro.

PhD in progress: Lokesh Gidra, UPMC, since Feb. 2011. Advisors: Gaël Thomas, Julien Sopena, Marc Shapiro.

PhD in progress: Marek Zawirski, UPMC, funded by Inria Google Research Grant. Since Oct. 2010. Advisor: Marc Shapiro.

8.2.3. Juries

Pierre Sens was the reviewer of:

- Jacques Jorda. HDR IRIT, Toulouse
- A. Sinah. PhD Irisa (Advisor : M. Banatre)
- D. Dib. PhD Irisa (Advisor : C. Morin)
- O. Shahmirzadi. PhD EPFL, Suisse (Advisor : A. Schiper)
- M. Servajean. PhD LIRMM (Advisor : E. Pacitti)
- T-T. Vu. PhD LIFL (Advisor : N. Meleb)

Franck Petit was the reviewer of:

- Greicy Marques-Costa, TIMA/ARIS, Grenoble (Advisor: R. Velazco),
- Ahmed Mouhamadou Wade, LaBRI Bordeaux (Advisor: R. Klasing)

Marc Shapiro was a member of the thesis committee of:

- Ghassan Almaless, UPMC Sorbonne Universités (advisor: Franck Weisburt).
- Jiaqing Du, École Polytechnique Fédérale de Lausanne (advisor: Willy Zwaenepeol).

8.3. Popularization

The team animated a stand on distributed algorithms at the 2014 edition of ”Fête de la Science”

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] 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], https://hal.inria.fr/hal-00868012

[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] 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

[11] J.-P. LOZI, F. DAVID, G. THOMAS, J. L. 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

[12] 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

[13] N. PALIX, G. THOMAS, S. SAHA, C. CALVÈS, J. L. 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

[14] 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

[15] P. CINCILLA. *Gargamel: boosting DBMS performance by parallelising write transactions*, UPMC, September 2014, https://hal.archives-ouvertes.fr/tel-01108975

[16] J. LEJEUNE. *Distributed mutual exclusion algorithmic : toward an efficient resource management*, Institut d'Optique Graduate School, September 2014, https://tel.archives-ouvertes.fr/tel-01077962

[17] M. S. A. SAEIDA ARDEKANI. *Ensuring consistency in partially replicated data stores*, Université Pierre et Marie Curie - Paris VI, September 2014, https://tel.archives-ouvertes.fr/tel-01086358

Articles in International Peer-Reviewed Journals

[18] V. BALEGAS, S. DUARTE, C. FERREIRA, R. RODRIGUES, M. NAJAFZADEH, M. SHAPIRO, N. PREGUIÇA. *Towards Fast Invariant Preservation in Geo-replicated Systems* , in "ACM SIGOPS, Operating Systems Review (ACM OSR)", January 2015, vol. 49, n^{o} 1, 5 p. [*DOI :* 10.1145/2723872.2723889], https://hal.inria.fr/hal-01111206

[19] A. L. S. GRADVOHL, H. SENGER, L. ARANTES, P. SENS. *Comparing Distributed Online Stream Processing Systems Considering Fault Tolerance Issues.* , in "Journal of Emerging Technologies in Web Intellingence", May 2014, vol. 6, n^{o} 2, 5 p. , https://hal.inria.fr/hal-01104700

[20] B. KEMME, A. SCHIPER, R. GANESAN, M. SHAPIRO. *Dagstuhl Seminar Review: Consistency in Distributed Systems*, in "ACM SIGACT News", March 2014, vol. 45, n^{o} 1, 22 p. [*DOI :* 10.1145/2596583.2596601], https://hal.inria.fr/hal-01109111

[21] M. VÉRON, O. MARIN, S. MONNET, Z. GUESSOUM. *Towards a scalable refereeing system for online gaming*, in "Springer Multimedia Systems Journal", October 2014, vol. 20, n^{o} 5, pp. 579-593 [*DOI :* 10.1007/s00530-014-0358-0], http://hal.upmc.fr/hal-01104664

International Conferences with Proceedings

[22] L. ARANTES, L. A. RODRIGUES, E. DUARTE. *An Autonomic Implementation of Reliable Broadcast Based on Dynamic Spanning Trees*, in "The Tenth European Dependable Computing Conference", New Castle, United Kingdom, 2014, https://hal.inria.fr/hal-01105183

[23] V. BALEGAS, M. NAJAFZADEH, S. DUARTE, C. FERREIRA, M. SHAPIRO, R. RODRIGUES, N. PREGUIÇA. *Putting the Consistency back into Eventual Consistency*, in "European Conference on Computer Systems (EuroSys)", Bordeaux, France, European Conference on Computer Systems (EuroSys), ACM, April 2015, https://hal.inria.fr/hal-01109121

[24] V. BALEGAS, N. PREGUIÇA, S. DUARTE, C. FERREIRA, R. RODRIGUES, M. NAJAFZADEH, M. SHAPIRO. *The Case for Fast and Invariant-Preserving Geo-Replication*, in "SRDSW 2014 - 33rd International Symposium on Reliable Distributed Systems Workshops", Nara, Japan, IEEE, October 2014, 5 p. [*DOI :* 10.1109/SRDSW.2014.30], https://hal.inria.fr/hal-01109107

[25] 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. , https://hal.inria.fr/hal-00934311

[26] R. CORTÉS, X. BONNAIRE, O. MARIN, P. SENS. *FreeSplit: A Write-Ahead Protocol to Improve Latency in Distributed Prefix Tree Indexing Structures*, in "29th IEEE International Conference on Advanced Information Networking and Applications (AINA-2015)", Gwangju, South Korea, March 2015, http://hal.upmc.fr/hal-01095702

[27] A. K. DATTA, A. LAMANI, L. LARMORE, F. PETIT. *Explorer un anneau avec des robots amnésiques et myopes*, in "ALGOTEL 2014 – 16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications", Le Bois-Plage-en-Ré, France, June 2014, pp. 1-4, https://hal.archives-ouvertes.fr/hal-00985611

[28] B. FRANCOIS, X. DÉFAGO, F. PETIT, M. GRADINARIU POTOP-BUTUCARU, S. TIXEUIL. *Discovering and Assessing Fine-Grained Metrics in Robot Networks Protocols*, in "Workshop on Self-organization in Swarm of Robots: from Molecular Robots to Mobile Agents (WSSR 2014)", Nara, Japan, October 2014, https://hal.inria.fr/hal-01108586

[29] L. GIDRA, G. THOMAS, J. SOPENA, M. SHAPIRO, N. D. NGUYEN. *NumaGiC: a garbage collector for big data on big NUMA machines*, in "20th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)", Istanbul, Turkey, ACM, March 2015, https://hal.inria.fr/hal-01109275

[30] O. MARIN, F. CORIAT, A. FLADENMULLER, L. ARANTES, E. ROSAS, N. HIDALGO. *Towards distributed geolocation for large scale disaster management*, in "WSDP - Chilean Workshop on Distributed and Parallel Systems", Talca, Chile, November 2014, http://hal.upmc.fr/hal-01104657

[31] K. PIRES, S. MONNET, P. SENS. *POPS: a popularity-aware live streaming service*, in "IEEE International Conference on Parallel and Distributed Systems (ICPADS 2014)", Hsinchu, Taiwan, 2014, https://hal.inria.fr/hal-01105050

[32] M. SAEIDA ARDEKANI, T. B. DOUGLAS. *A Self-Configurable Geo-Replicated Cloud Storage System*, in "11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)", Broomfield, CO, United States, 2014, https://hal.inria.fr/hal-01102803

[33] M. SAEIDA ARDEKANI, P. SUTRA, M. SHAPIRO. *G-DUR: A Middleware for Assembling, Analyzing, and Improving Transactional Protocols*, in "Middleware", Bordeaux, France, IEEE, December 2014, 12 p. [*DOI :* 10.1145/2663165.2663336], https://hal.inria.fr/hal-01109114

[34] 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.upmc.fr/hal-00940774

National Conferences with Proceedings

[35] M. LORRILLERE, J. SOPENA, S. MONNET, P. SENS. *PUMA: Un cache distant pour mutualiser la mémoire inutilisée des machines virtuelles*, in "ComPAS'2014 : Conférence d'informatique en Parallélisme, Architecture et Système", Neuchâtel, Switzerland, P. FELBER, L. PHILIPPE, E. RIVIERE, A. TISSERAND (editors), April 2014, pp. 1-12, https://hal.archives-ouvertes.fr/hal-00983984

[36] K. PIRES, S. MONNET, P. SENS. *POPS : service de diffusion de flux vidéos live prenant en compte la popularité*, in "ComPAS 2014", Neuchâtel, Switzerland, 2014, https://hal.inria.fr/hal-01105063

[37] M. VÉRON, O. MARIN, S. MONNET. *Matchmaking dans les jeux mutlijoueurs en ligne : e´tudier les traces utilisateurs pour ame´liorer l'expe´rience de jeu*, in "Conférence d'informatique en Parallélisme, Architecture et Système", Neuchâtel, Switzerland, April 2014, http://hal.upmc.fr/hal-01104668

Conferences without Proceedings

[38] V. BALEGAS, M. NAJAFZADEH, S. DUARTE, C. FERREIRA, M. SHAPIRO, R. RODRIGUES, N. PREGUIÇA. *Putting the Consistency Back Into Eventual Consistency*, in "Large-Scale Distributed Systems and Middleware (LADIS) 2014", Cambridge, United Kingdom, Large-Scale Distributed Systems and Middleware (LADIS) 2014, October 2014, https://hal.inria.fr/hal-01109719

[39] P. CINCILLA, S. MONNET, M. SHAPIRO. *Multi-site Gargamel: Optimistic synchronization for reliable geo-replicated databases*, in "International Workshop on Middleware for Dependable Systems and Networks", Bordeaux, France, 2014, https://hal.inria.fr/hal-01105053

[40] M. MAKPANGOU, B. NGOM, S. NDIAYE. *FreeCore : Un substrat d'indexation des filtres de Bloom fragmentés pour la recherche par mots clés*, in "ComPAS'2014", Neuchâtel, Switzerland, April 2014, https://hal.archives-ouvertes.fr/hal-01049544

[41] M. SHAPIRO, M. SAEIDA ARDEKANI, P. SUTRA. *Exploring the spectrum of strongly-consistent transactional protocols*, in "Workshop on Large-Scale Distributed Systems and Middleware (LADIS)", Cambridge, United Kingdom, Workshop on Large-Scale Distributed Systems and Middleware (LADIS), October 2014, https://hal.inria.fr/hal-01109740

Scientific Books (or Scientific Book chapters)

[42] *Principles of Distributed Systems, 18th Int. Conf. on (OPODIS 2014)*, Lecture Notes in Computer Science, Springer-VerlagCortina d'Ampezzo, Italy, December 2014, vol. 8878 [*DOI :* 10.1007/978-3-319-14472-6], https://hal.inria.fr/hal-01109118

Research Reports

[43] K. ALTISEN, A. COURNIER, S. DEVISMES, A. DURAND, F. PETIT. *Self-Stabilizing Leader Election in Polynomial Steps*, VERIMAG, May 2014, https://hal.archives-ouvertes.fr/hal-00980798

[44] K. ATTOUCHI, G. THOMAS, A. BOTTARO, J. L. LAWALL, G. MULLER. *Incinerator - Eliminating Stale References in Dynamic OSGi Applications*, Inria, February 2014, n° RR-8485, 22 p. , https://hal.inria.fr/hal-00952327

[45] N. BRAUD-SANTONI, S. DUBOIS, M.-H. KAAOUACHI, F. PETIT. *The Next 700 Impossibility Results in Time-Varying Graphs*, UPMC Sorbonne Universités/CNRS/Inria - EPI REGAL, December 2014, https://hal.inria.fr/hal-01097109

[46] R. CORTÉS, X. BONNAIRE, O. MARIN, P. SENS. *FreeSplit: A Write-Ahead Protocol to Improve Latency in Distributed Prefix Tree Indexing Structures*, Inria Paris, November 2014, n° RR-8637, https://hal.inria.fr/hal-01092251

[47] R. CORTÉS, X. BONNAIRE, O. MARIN, P. SENS. *Sport Trackers and Big Data: Studying user traces to identify opportunities and challenges*, Inria Paris, November 2014, n° RR-8636, https://hal.inria.fr/hal-01092242

[48] S. DUBOIS, M.-H. KAAOUACHI, F. PETIT. *Enabling Minimal Dominating Set in Highly Dynamic Distributed Systems*, UPMC Sorbonne Universités/CNRS/Inria - EPI REGAL, January 2015, https://hal.inria.fr/hal-01111610

[49] V. SIMON, S. MONNET, M. FEUILLET, P. ROBERT, P. SENS. *SPLAD: scattering and placing data replicas to enhance long-term durability*, inria, May 2014, n° RR-8533, 23 p. , https://hal.inria.fr/hal-00988374

Scientific Popularization

[50] 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*, April 2014, ComPAS 2014 : conférence en parallélisme, architecture et systèmes, https://hal.inria.fr/hal-00991709

Other Publications

[51] S. DEVISMES, A. LAMANI, F. PETIT, S. TIXEUIL. *Optimal Torus Exploration by Oblivious Mobile Robots*, January 2014, https://hal.inria.fr/hal-00926573

[52] M. SEGURA, V. RANCUREL, V. TO, M. SHAPIRO. *Scality's experience with a geo-distributed file system*, Middleware: Posters and Demos '14; Proceedings of the Posters & Demos Session, ACM, December 2014, 1 p. , Middleware 2014 [*DOI :* 10.1145/2678508.2678524], https://hal.inria.fr/hal-01111208

References in notes

[53] V. BALEGAS, M. NAJAFZADEH, S. DUARTE, C. FERREIRA, M. SHAPIRO, R. RODRIGUES, N. PREGUIÇA. *Putting the Consistency back into Eventual Consistency*, in "European Conference on Computer Systems (EuroSys)", Bordeaux, France, European Conference on Computer Systems (EuroSys), ACM, April 2015, https://hal.inria.fr/hal-01109121