RESEARCH CENTRE

Paris

IN PARTNERSHIP WITH:

Sorbonne Université

2020 ACTIVITY REPORT

Project-Team DELYS

DistributEd aLgorithms and sYStems

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

DOMAIN

Networks, Systems and Services, Distributed Computing

THEME

Distributed Systems and middleware

Contents

Project-Team DELYS

Creation of the Team: 2018 January 01, updated into Project-Team: 2019 January 01

Keywords

Computer sciences and digital sciences

A1.1.1. – Multicore, Manycore
A1.1.9. – Fault tolerant systems
A1.1.13. – Virtualization
A1.2.5. – Internet of things
A1.3.2. – Mobile distributed systems
A1.3.3. – Blockchain
A1.3.4. – Peer to peer
A1.3.5. – Cloud
A1.3.6. – Fog, Edge
A1.5.2. – Communicating systems
A2.6. – Infrastructure software
A2.6.1. – Operating systems
A2.6.2. – Middleware
A2.6.3. – Virtual machines
A2.6.4. – Ressource management
A3.1.3. – Distributed data
A3.1.8. – Big data (production, storage, transfer)
A7.1.1. – Distributed algorithms

Other research topics and application domains

B6.4. – Internet of things

1 Team members, visitors, external collaborators

Research Scientists

- Mesaac Makpangou [Inria, Researcher, HDR]
- Marc Shapiro [Inria, Emeritus, HDR]

Faculty Members

- Pierre Sens [Team leader, Université Pierre et Marie Curie, Professor, HDR]
- Luciana Bezerra Arantes [Sorbonne Université, Associate Professor]
- Philippe Darche [Université René Descartes, Associate Professor]
- Swan Dubois [Sorbonne Université, Associate Professor]
- Colette Johnen [Université de Bordeaux, Professor, from Sep 2020, HDR]
- Jonathan Lejeune [Sorbonne Université, Associate Professor]
- Franck Petit [Sorbonne Université, Professor, HDR]
- Julien Sopena [Sorbonne Université, Associate Professor]

Post-Doctoral Fellow

- Sara Hamouda [Inria, until Jul 2020]

PhD Students

- Jose Jurandir Alves Esteves [Orange Labs]
- Arnaud Favier [Inria]
- Saalik Hatia [Sorbonne Université]
- Célia Mahamdi [Sorbonne Université]
- Benoît Martin [Sorbonne Université]
- Sreeja Nair [Sorbonne Université]
- Laurent Prosperi [Inria]
- Jonathan Sid-Otmane [Orange Labs, CIFRE]
- Ilyas Toumlilt [Sorbonne Université, from Sep 2020]
- Dimitrios Vasilas [Sorbonne Univeristé, from Nov 2020]
- Daniel Wilhelm [Sorbonne Université]

Technical Staff

- Yannick Li [Inria, Engineer, from Oct 2020]
- Ilyas Toumlilt [Inria, Engineer, until Aug 2020]

Administrative Assistants

- Christine Anocq [Inria]
- Nelly Maloisel [Inria]

External Collaborator

- Sébastien Monnet [Université Savoie Mont-Blanc]

2 Overall objectives

The research of the Delys team addresses the theory and practice of distributed systems, including multicore computers, clusters, networks, peer-to-peer systems, cloud, fog end edge computing systems, and other communicating entities such as swarms of robots. It addresses the challenges of correctly communicating, sharing information, and computing in such large-scale, highly dynamic computer systems. This includes addressing the core problems of communication, consensus and fault detection, scalability, replication and consistency of shared data, information sharing in collaborative groups, dynamic content distribution, and multi- and many-core concurrent algorithms.

Delys is a joint research team between LIP6 (Sorbonne University/CNRS) and Inria Paris.

3 Research program

3.1 Research rationale

DELYS addresses both theoretical and practical issues of *Computer Systems*, leveraging our dual expertise in theoretical and experimental research. Our approach is a "virtuous cycle," triggered by issues with real systems, of algorithm design which we prove correct and evaluate theoretically, and then implement and test experimentally feeding back to theory. The major challenges addressed by DELYS are the sharing of information and guaranteeing correct execution of highly-dynamic computer systems. Our research covers a large spectrum of distributed computer systems: multicore computers, mobile networks, cloud computing systems, and dynamic communicating entities. This holistic approach enables handling related problems at different levels. Among such problems we can highlight consensus, fault detection, scalability, search of information, resource allocation, replication and consistency of shared data, dynamic content distribution, and concurrent and parallel algorithms.

Two main evolutions in the Computer Systems area strongly influence our research project:

(1) Modern computer systems are **increasingly distributed**, **dynamic** and composed of multiple devices **geographically spread over heterogeneous platforms**, spanning multiple management domains. Years of research in the field are now coming to fruition, and are being used by millions of users of web systems, peer-to-peer systems, gaming and social applications, cloud computing, and now fog computing. These new uses bring new challenges, such as *adaptation to dynamically-changing conditions*, where knowledge of the system state can only be partial and incomplete.

(2) **Heterogeneous architectures and virtualisation are everywhere**. The parallelism offered by distributed clusters and *multicore* architectures is opening highly parallel computing to new application areas. To be successful, however, many issues need to be addressed. Challenges include obtaining a consistent view of shared resources, such as memory, and optimally distributing computations among heterogeneous architectures. These issues arise at a more fine-grained level than before, leading to the need for different solutions down to OS level itself.

The scientific challenges of the distributed computing systems are subject to many important features which include scalability, fault tolerance, dynamics, emergent behaviour, heterogeneity, and virtualisation at many levels. Algorithms designed for traditional distributed systems, such as resource allocation, data storage and placement, and concurrent access to shared data, need to be redefined or revisited in order to work properly under the constraints of these new environments. Sometimes, classical "*static*" problems, (*e.g.*, Leader Election, Spanning Tree Construction, ...) even need to be redefined to consider the unstable nature of the distributed system. In particular, DELYS will focus on a number of key challenges:

Consistency in geo-scale systems. Distributed systems need to scale to large geographies and large numbers of attached devices, while executing in an untamed, unstable environment. This poses difficult scientific challenges, which are all the more pressing as the cloud moves more and more towards the edge, IoT and mobile computing. A key issue is how to share data effectively and consistently across the whole spectrum. Delys has made several key contributions, including CRDTs, the Transactional Causal Consistency Plus model, the AntidoteDB geo-distributed database, and its edge extension EdgeAnt.

Rethinking distributed algorithms. From a theoretical point of view the key question is how to adapt the fundamental building blocks to new architectures. More specifically, how to rethink the classical algorithms to take into account the dynamics of advanced modern systems. Since a recent past, there have been several papers that propose models for dynamic systems: there is practically a different model for each setting and currently there is no unification of models. Furthermore, models often suffer of lack of realism. One of the key challenge is to identify which assumptions make sense in new distributed systems. DELYS's objectives are then (1) to identify under which realistic assumptions a given fundamental problem such as mutual exclusion, consensus or leader election can be solved and (2) to design efficient algorithms under these assumptions.

Resource management in heterogeneous systems. The key question is how to manage resources on large and heterogeneous configurations. Managing resources in such systems requires fully decentralized solutions, and to rethink the way various platforms can collaborate and interoperate with each other. In this context, data management is a key component. The fundamental issue we address is how to efficiently and reliably share information in highly distributed environments.

Adaptation of runtimes. One of the main challenge of the OS community is how to adapt runtime supports to new architectures. With the increasingly widespread use of multicore architectures and virtualised environments, internal runtime protocols need to be revisited. Especially, memory management is crucial in OS and virtualisation technologies have highly impact on it. On one hand, the isolation property of virtualisation has severe side effects on the efficiency of memory allocation since it needs to be constantly balanced between hosted OSs. On the other hand, by hiding the physical machine to OSs, virtualisation prevents them to efficiently place their data in memory on different cores. Our research will thus focus on providing solutions to efficiently share memory between OSs without jeopardizing isolation properties.

4 Application domains

We target highly distributed infrastructures composed of multiple devices geographically spread over heterogeneous platforms including cloud, fog computing and IoT.

At OS level, we study multicore architectures and virtualized environments based on VM hypervisors and containers. Our research focuses on providing solutions to efficiently share memory between virtualized environments.

5 Highlights of the year

5.1 Awards

Francis Laniel received the Best Paper Award at IEEE NCA 2020 for "MemOpLight: Leveraging application feedback to improve container memory consolidation" [29]

Sreeja Nair was awarded the "Séphora Berrebi Scholarship for Women in Advanced Mathematics & Computer Science" (2020, 3d edition).

6 New results

6.1 Distributed Algorithms for Dynamic Networks and Fault Tolerance

Participants Luciana Bezerra Arantes, Swan Dubois, Arnaud Favier, Colette Johnen, Jonathan Lejeune, Célia Mahamdi, Mesaac Makpangou, Franck Petit, Pierre Sens, 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.

The fact that computing units may leave, join, or move may result of an intentional behavior or not. In the latter case, the system may be subject to disruptions due to component faults that can be permanent, transient, exogenous, evil-minded, etc. It is therefore crucial to come up with solutions tolerating some types of faults.

In 2020, we obtained the following results.

Leader election Eventual leader election is an essential service for many reliable applications that require coordination actions on top of asynchronous fail-prone distributed systems. In [26] we proposed an new algorithm that eventually elects a leader for each connected component of a dynamic network where nodes can move or fail by crash. A node only communicates with nodes in its transmission range and locally keeps a global view, denoted topological knowledge, of the communication graph of the network and its dynamic evolution. Every change in the topology or in nodes membership is detected by one or more nodes and propagated over the network, updating thus the topological knowledge of the nodes. As the choice of the leader has an impact on the performance of applications that use an eventual leader election service, our algorithm, thanks to nodes topological knowledge, exploits the closeness centrality as the criterion for electing a leader. Experiments were conducted on top of PeerSim simulator, comparing our algorithm to a representative flooding algorithm. Performance results show that our algorithm outperforms the flooding one when considering leader choice stability, number of messages, and average distance to the leader.

Self-Stabilizing Leader election Essentially, self-stabilizing algorithms tolerate *transient failures*, since by definition such failures last a finite time (as opposed to crash failures, for example) and their frequency is low (as opposed to intermittent failures).

We initiate research on self-stabilization in highly dynamic identified message passing systems where the dynamics is modeled using TVGs to obtain solutions tolerating both transient faults and high dynamics in [16]. We reformulate the definition of self-stabilization to accommodate Time-Vary Graphs (TVGs).

We investigate the self-stabilizing leader election problem. This problem is fundamental in distributed computing since it allows to synchronize and self-organize a network. In [17], we have studied this problem in three classes of TVGs: (i) the $\mathcal{TC}^{\mathcal{B}}(\Delta)$ class of TVGs with temporal diameter bounded by Δ, (ii) the $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ class of TVGs with temporal diameter almost bounded by Δ and (iii) the class of TVGs with recurrent temporal connectivity, $\mathcal{TC}^{\mathcal{R}}$. We show that in spite of the identities, in the $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ class and in the $\mathcal{TC}^{\mathcal{R}}$ class, any self-stabilizing election algorithm requires the exact knowledge of the number of processes. Then, we propose three election algorithms. The first, for the $\mathcal{TC}^{\mathcal{B}}(\Delta)$ class, stabilizes at most 3Δ rounds. In the classes $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ and $\mathcal{TC}^{\mathcal{R}}$, the stabilization time of a self-stabilizing election algorithm cannot be limited. However, we show that our two solutions are speculative, i.e. they have good performances in favorable cases; indeed, they stabilize in $O(\Delta)$ rounds when restricted to the $\mathcal{TC}^{\mathcal{B}}(\Delta)$ class.

From Gathering to Leader Election A team of mobile agents, starting from different nodes of an unknown network, possibly at different times, have to meet at the same node and declare that they have all met. Agents have different labels which are positive integers, and move in synchronous rounds along

links of the network. The above task is known as gathering and was traditionally considered under the assumption that when some agents are at the same node then they can talk, i.e., exchange currently available information. In [23], we ask the question of whether this ability of talking is needed for gathering. The answer turns out to be no.

Our main contribution are two deterministic algorithms that always accomplish gathering in a much weaker model. We only assume that at any time an agent knows how many agents are at the node that it currently occupies but agents do not see the labels of other co-located agents and cannot exchange any information with them. They also do not see other nodes than the current one. Our first algorithm works under the assumption that agents know *a priori* some upper bound N on the size of the network, and it works in time polynomial in N and in the length ℓ of the smallest label. Our second algorithm does not assume any *a priori* knowledge about the network but its complexity is exponential in the size ofthe network and in the labels of agents. Its purpose is to show feasibility of gathering under this harsher scenario.

As a by-product of our techniques we obtain, in the same weak model, the solution of the fundamental problem of leader election among agents: One agent is elected aleader and all agents learn its identity. As an application of our result we also solve, in the same model, the well-known *gossiping* problem: if each agent has a message at the beginning, we show how to make all messages known to all agents, even without any *a priori* knowledge about the network. If agents know an upper bound N on the size of the network then our gossiping algorithm works in time polynomial in N, in the length of the smallest label ℓ and in the length of the largest message.

Robustness In [10], we investigate a special case of hereditary property in graphs, referred to as *robustness*. A property (or structure) is called robust in a graph G if it is inherited by all the connected spanning subgraphs of G. We motivate this definition using two different settings of dynamic networks. The first corresponds to networks of low dynamicity, where some links may be permanently removed so long as the network remains connected. The second corresponds to highly-dynamic networks, where communication links appear and disappear arbitrarily often, subject only to the requirement that the entities are temporally connected in a recurrent fashion (*i.e.* they can always reach each other through temporal paths). Each context induces a different interpretation of the notion of robustness.

We start by motivating the definition and discussing the two interpretations, after what we consider the notion independently from its interpretation, taking as our focus the robustness of *maximal independent sets* (MIS). A graph may or may not admit a robust MIS. We characterize the set of graphs in which *all* MISs are robust. Then, we turn our attention to the graphs that *admit* a robust MIS. This class has a more complex structure; we give a partial characterization in terms of elementary graph properties, then a complete characterization by means of a (polynomial time) decision algorithm that accepts if and only if a robust MIS exists. This algorithm can be adapted to construct such a solution if one exists.

Treasure Hunt with Angular Hints In [9], we consider a mobile agent equipped with a compass and a measure of length has to find an inert treasure in the Euclidean plane. Both the agent and the treasure are modeled as points. In the beginning, the agent is at a distance at most $D > 0$ from the treasure, but knows neither the distance nor any bound on it. Finding the treasure means getting at distance at most 1 from it. The agent makes a series of moves. Each of them consists in moving straight in a chosen direction at a chosen distance. In the beginning and after each move the agent gets a hint consisting of a positive angle smaller than 2π whose vertex is at the current position of the agent and within which the treasure is contained. We investigate the problem of how these hints permit the agent to lower the cost of finding the treasure, using a deterministic algorithm, where the cost is the worst-case total length of the agent's trajectory. It is well known that without any hint the optimal (worst case) cost is $\Theta(D^2)$. We show that if all angles given as hints are at most π, then the cost can be lowered to $O(D)$, which is optimal. If all angles are at most β, where $\beta < 2\pi$ is a constant unknown to the agent, then the cost is at most $O(D^{2-\epsilon})$, for some $\epsilon > 0$. For both these positive results we present deterministic algorithms achieving the above costs. Finally, if angles given as hints can be arbitrary, smaller than 2π, then we show that cost $\Theta(D^2)$ cannot be beaten.

Grid Exploration by Asynchronous Oblivious Robots In [13], we deal with a team of autonomous robots that are endowed with motion actuators and visibility sensors. Those robots are weak and evolve in a discrete environment. By weak, we mean that they are anonymous, uniform, unable to explicitly communicate, and oblivious.

We propose optimal (*w.r.t.* the number of robots) deterministic solutions for the *terminating exploration* of an anonymous grid-shaped network by a team of asynchronous oblivious robots.

We first consider the semi-synchronous model. We show that it is impossible to explore a grid of at least 3 nodes with less than 3 robots. Next, we show that it is impossible to explore a $(2,2)$-Grid with less than 4 robots, and a $(3,3)$-Grid with less than 5 robots, respectively. The two first results hold for both deterministic and probabilistic settings, while the latter holds only in the deterministic case.

We then consider the asynchronous model. This latter being strictly weakest than the semi-synchronous model, all the aforementioned impossibility results still hold in that context. We then propose deterministic algorithms to exhibit the optimal number of robots allowing to explore of a given grid. Our results show that except in two particular cases, 3 robots are necessary and sufficient to deterministically explore a grid of at least 3 nodes. The optimal number of robots for the two remaining cases is: 4 for the $(2,2)$-Grid and 5 for the $(3,3)$-Grid, respectively.

Anonymous Rendezvous in the Plane Two mobile agents represented by points freely moving in the plane and starting at two different positions, have to meet. The meeting, called *rendezvous*, occurs when agents are at distance at most r of each other and never move after this time, where r is a positive real unknown to them, called the *visibility radius*. Agents are anonymous and execute the same deterministic algorithm. Each agent has a set of private *attributes*, some or all of which can differ between agents. These attributes are: the initial position of the agent, its system of coordinates (orientation and chirality), the rate of its clock, its speed when it moves, and the time of its wake-up. If all attributes (except the initial positions) are identical and agents start at distance larger than r then they can never meet, as the distance between them can never change. However, differences between attributes make it sometimes possible to break the symmetry and accomplish rendezvous. Such instances of the rendezvous problem (formalized as lists of attributes), are called *feasible*.

Our contribution in [24] is three-fold. We first give an exact characterization of feasible instances. Thus it is natural to ask whether there exists a single algorithm that guarantees rendezvous for all these instances. We give a strong negative answer to this question: we show two sets S_1 and S_2 of feasible instances such that none of them admits a single rendezvous algorithm valid for all instances of the set. On the other hand, we construct a single algorithm that guarantees rendezvous for all feasible instances outside of sets S_1 and S_2. We observe that these exception sets S_1 and S_2 are geometrically very small, compared to the set of all feasible instances: they are included in low-dimension subspaces of the latter. Thus, our rendezvous algorithm handling all feasible instances other than these small sets of exceptions can be justly called *almost universal*.

Minimum Spanning Tree approximation We explore the impact of approximation on time-polynomial distributed algorithms. In particular we show in [22] that approximation can help reduce the space used for self-stabilization. In the classic state model, where the nodes of a network communicate by reading the states of their neighbors, an important measure of efficiency is the space: the number of bits used at each node to encode the state. In this model, a classic requirement is that the algorithm has to be silent, that is, after stabilization the states should not change anymore. We design a silent self-stabilizing algorithm for the problem of minimum spanning tree, that has a trade-off between the quality of the solution and the space needed to compute it.

Collaborative decision-making in heterogeneous and dynamic environment New distributed system models such as Fog computing are based on computing resources decentralization. However, this complicates the orchestration of distributed resources due to its large-scale, unreliable and highly dynamic nature preventing any effcicient construction of a global and consistent view. Thus, we need to define new decentralized solutions where different autonomous subsytems, having a local view of their own resources, are able to make collaborative decisions in a reasonable time while limiting the communication cost. In this axis, we currently work about a new model of collaborative decicion-making based

on consensus protocols (such as Paxos). In our model, we consider several concurrent sets of nodes on a common dynamic infrastructure where each set runs an instance of a consensus protocol to decide the value of a shared and replicated variable. A given node can belong to several consensus set. This implies that several decisions can be taken asynchronously in the system by several subsets of nodes and decisions conflicts may occur. In case of decision conflict, we need to revoke some decisions in order to garantee the invariants of nodes, which consequently modify the initial definition of the consensus problem. Our works are focused on 1) the execution optimisation of a high number of concurrent consensus and 2) the problem of decision revocability.

This work has been submitted for publication.

6.2 Distributed systems and Large-scale data distribution

Participants Guillaume Fraysse, Jose Jurandir Alves Esteves, Pierre Sens, Marc Shapiro, Julien Sopena.

Resource management in large networks Network Operators expect to accurately satisfy a wide range of user's needs by providing fully customized services relying on Network Slicing. The efficiency of Network Slicing depends on an optimized management of network resources and Quality of Service (QoS). We focus on Network Slice placement optimization problem.

In [19] we propose a Proof-of-Concept (PoC) illustrated by an Interactive Gaming time-sensitive use case. In [21], We focus on Virtual Network Functions (VNF) Placement and Chaining problem. In contrary to most studies related to VNF placement, we deal with the most complete and complex Network Slice topologies and we pay special attention to the geographic location of Network Slice Users. We propose a data model adapted to Integer Linear Programming. Extensive numerical experiments assess the relevance of taking into account the user location constraints. We also propose in [18] an online heuristic algorithm for the problem of network slice placement optimization. The solution is adapted to support placement on large scale networks and integrates Edge-specific and URLLC constraints. We rely on an approach called the Power of Two Choices to build the heuristic. The evaluation results show the good performance of the heuristic that solves the problem in few seconds under a large scale scenario. The heuristic also improves the acceptance ratio of network slice placement requests when compared against a deterministic online Integer Linear Programming (ILP) solution.

In [27], we study slicing in the context of 5G networks for allowing multiple users to share a common infrastructure. The chaining of Network Function (NFs) introduces constraints on the order in which NFs are allocated. We first model the allocation of resources for Chains of NFs in 5G Slices. Then we introduce a distributed mutual exclusion algorithm to address the problem of the allocation of resources. We show with selected metrics that choosing an order of allocation of the resources that differs from the order in which resources are used can give better performances. We then show experimental results where we improve the usage rate of resources by more than 20% compared to the baseline algorithm in some cases. The experiments run on our own simulator based on SimGrid.

Task scheduling in cloud environments Cloud platforms usually offer several types of Virtual Machines (VMs) with different guarantees in terms of availability and volatility, provisioning the same resource through multiple pricing models. For instance, in the Amazon EC2 cloud, the user pays per use for on-demand VMs while spot VMs are instances available at lower prices. However, a spot VM can be terminated or hibernated by EC2 at any moment. In [14], we proposed the Hibernation-Aware Dynamic Scheduler (HADS) that schedules Bag-of-Tasks (BoT) applications with deadline constraints in both hibernation prone spots VMs and on-demand VMs. HADS aims at minimizing the monetary costs of executing BoT applications on Clouds ensuring that their deadlines are respected even in the presence of multiple hibernations. Results collected from experiments on Amazon EC2 VMs using synthetic applications and a NAS benchmark application show the effectiveness of HADS in terms of monetary costs when compared to on-demand VM only solutions.

6.3 Leveraging Formal Approaches and Verification in Distributed System Design

Participants Saalik Hatia, Sreeja Nair, Laurent Prosperi, Pierre Sens, Marc Shapiro.

Proving the safety of highly-available distributed objects To provide high availability in distributed systems, object replicas allow concurrent updates. Although replicas eventually converge, they may diverge temporarily, for instance when the network fails. This makes it difficult for the developer to reason about the object's properties , and in particular, to prove invariants over its state. For the sub-class of state-based distributed systems, we propose a proof methodology for establishing that a given object maintains a given invariant, taking into account any concurrency control. Our approach allows reasoning about individual operations separately. We demonstrate that our rules are sound, and we illustrate their use with some representative examples. We automate the rule using Boogie, an SMT-based tool.

This work was published at the 29th European Symposium on Programming (ESOP), April 2020, Dublin, Ireland [31].

A coordination-free, convergent, and safe replicated tree The tree is a basic data structure present in many applications. We consider the case where the tree is replicated across a distributed system, for instance in a distributed file system. To improve performance and availability, it is desirable to support concurrent updates to different replicas without coordination. Such concurrent updates converge if the effects commute. However, in a naïve implementation, concurrent moves might violate the tree invariant. To avoid this issue, previous approaches would either eschew atomic moves, require preventative cross-replica coordination, or totally order move operations after-the-fact, requiring roll-back and compensation operations.

In this work, we study a novel replicated tree data structure that supports coordination-free concurrent atomic moves, and provably maintains the tree invariant. Our analysis identifies cases where concurrent moves are inherently safe, and we devise a coordination-free, rollback-free algorithm for the remaining cases. The trade-off is that in some cases a move operation "loses" (i.e., is interpreted as skip).

We present a detailed analysis of the concurrency issues with trees, justifying our replicated tree data structure. We provide mechanized proof that the data structure is convergent and maintains the tree invariant. Finally, we compare the response time and availability of our design against the literature.

This work has been submitted for publication.

Specification of a Transactionally and Causally-Consistent (TCC) database Large-scale application are typically built on top of geo-distributed databases running on multiple datacenters (DCs) situated around the globe. Network failures are unavoidable, but in most internet services, availability is not negotiable; in this context, the CAP theorem proves that it is impossible to provide both availability and strong consistency at the same time. Sacrificing strong consistency, exposes developers to complex anomalies that are complex to build against. AntidoteDB is a database designed for geo-replication. As it aims to provide high availability with the strongest possible consistency model, it guarantees Transactional Causal Consistency (TCC) and supports CRDTs. TCC means that: (1) if one update happens before another, they will be observed in the same order (causal consistency), and (2) updates in the same transaction are observed all-or-nothing. In AntidoteDB, the database is persisted as a journal of operations. In the current implementation, the journal grows without bound. The main objective of this work is to specify a mechanism for pruning the journal safely, by storing recent checkpoints. This will enable faster reads and crash recovery.

Work in cooperation with Annette Bieniusa (Uni Kaiserslautern), Carla Ferreira (Universidade NOVA de Lisboa) and Gustao Petri (ARM, Cambridge, UK).

An environment for composable distributed computing Modern applications are highly distributed and data-intensive. Programming a distributed system is challenging because of asynchrony, failures and trade-offs. In addition, application requirements vary with the use-case and throughout the development cycle. Moreover, existing tools come with restricted expressiveness or limited runtime customizability.

Our work aims to address this by improving reuse while maintaining fine-grain control and enhancing dependability. We argue that an environment for composable distributed computing will facilitate the process of developing distributed systems. We use high-level composable specification, verification tools and a distributed runtime.

This work was presented at the EuroSys Doctoral Workshop by Benoît Martin and Laurent Prosperi [37].

6.4 Resource management in system software

Participants Jonathan Lejeune, Marc Shapiro, Julien Sopena, Francis Laniel.

MemOpLight: Leveraging applicative feedback to improve container memory consolidation The container mechanism supports consolidating several servers on the same machine, thus amortizing cost. To ensure performance isolation between containers, Linux relies on memory limits. However these limits are static, but application needs are dynamic; this results in poor performance. To solve this issue, MemOpLight reallocates memory to containers based on dynamic applicative feedback. MemOpLight rebalances physical memory allocation, in favor of under-performing ones, with the aim of improving overall performance. Our research explores the issues, addresses the design of MemOpLight, and validates it experimentally. Our approach increases total satisfaction by 13% compared to the default.

It is standard practice in Infrastructure as a Service to *consolidate* several logical servers on the same physical machine, thus amortizing cost. However, the execution of one logical server should not disturb the others: the logical servers should remain *isolated* from one another.

To ensure both consolidation and isolation, a recent approach is "containers," a group of processes with sharing and isolation properties. To ensure *memory performance isolation*, *i.e.*, guaranteeing to each container enough memory for it to perform well, the administrator limits the total amount of physical memory that a container may use at the expense of others. In previous work, we showed that these limits impede memory consolidation. Furthermore, the metrics available to the kernel to evaluate its policies (*e.g.*, frequency of page faults, I/O requests, use of CPU cycles, *etc.*), are not directly relevant to performance as experienced from the application perspective, which is better characterized by, for instance, response time or throughput measured at application level.

To solve these problems, we propose a new approach, called the Memory Optimization Light (MemOpLight). It is based on application-level feedback from containers. Our mechanism aims to rebalance memory allocation in favor of unsatisfied containers, while not penalizing the satisfied ones. By doing so, we guarantee application satisfaction, while consolidating memory; this also improves overall resource consumption.

Our main contributions are the following:

- An experimental demonstration of the limitations of the existing Linux mechanisms.
- The design of a simple feedback mechanism from application to the kernel.
- An algorithm for adapting container memory allocation.
- And implementation in Linux and experimental confirmation.

These results published at NCA 2020 [29] obtained the best paper paper award.

Leveraging High-Frequency Cores in the OS Scheduler In modern server CPUs, individual cores can run at different frequencies, which allows for fine-grained control of the performance/energy tradeoff. Adjusting the frequency, however, incurs a high latency. We find that this can lead to a problem of frequency inversion, whereby the Linux scheduler places a newly active thread on an idle core that takes dozens to hundreds of milliseconds to reach a high frequency, just before another core already running at a high frequency becomes idle. In [28], we first illustrate the significant performance overhead of repeated frequency inversion through a case study of scheduler behavior during the compilation of the Linux

kernel on an 80-core Intel R Xeon-based machine. Following this, we propose two strategies to reduce the likelihood of frequency inversion in the Linux scheduler. When benchmarked over 60 diverse applications on the Intel R Xeon, the better performing strategy, Smove, improves performance by more than 5% (at most 56% with no energy overhead) for 23 applications, and worsens performance by more than 5% (at most 8%) for only 3 applications. On a 4-core AMD Ryzen we obtain performance improvements up to 56%.

7 Bilateral contracts and grants with industry

7.1 Bilateral contracts with industry

DELYS has a CIFRE contract with Scality SA:

- Dimitrios Vasilas is advised by Marc Shapiro and Brad King. He works on secondary indexing in large-scale storage systems under weak consistency.

DELYS has three contracts with Orange within the I/O Lab joint laboratory:

- Guillaume Fraysse is advised by Jonathan Lejeune, Julien Sopena, and Pierre Sens. He works on distributed resources allocation in virtual network environments.
- Jonathan Sid-Otmane is advised by Marc Shapiro. He studies the applications of distributed databases to the needs of the telco industry in the context of 5G.
- José Alves Esteves Jurandir is advised by Pierre Sens. He works on network slice placement stategies.

7.2 Startup support from Inria

Marc Shapiro received support from Inria Startup Studio to incubate start-up concordant.io, developing CRDT-based solutions for geo-scale and edge distribution of data. ISS supports two software engineers for 12 months.

8 Partnerships and cooperations

8.1 National initiatives

8.1.1 ANR

AdeCoDS (2019–2023)

Title: Programming, verifying, and synthesizing Adequately-Consistent Distributed Systems (AdeCoDS).

Members: Université de Paris (project leader), Sorbonne-Université LIP6, ARM, Orange.

Funding: The total funding of AdeCoDS from ANR is 523 471 euros, of which 162 500 euros for Delys.

Objectives The goal of the project is to provide a framework for programming distributed systems that are both correct and efficient (available and performant). The idea is to offer to developers a programming framework where it is possible, for a given application, (1) to build implementations that are correct under specific assumptions on the consistency level guaranteed by the infrastructure (e.g., databases and libraries of data structures), and (2) to discover in a systematic way the different trade-offs between the consistency level guaranteed by the infrastructure and the type and the amount of synchronization they need to use in their implementation in order ensure its correctness. For that, the project will develop a methodology based on combining (1) automated verification and synthesis methods, (2) language-based methods for correct programming, and (3) techniques for efficient system design.

ESTATE - (2016–2021)

Members: LIP6 (DELYS, project leader), LaBRI (Univ. de Bordeaux); Verimag (Univ. de Grenoble).

Funding: ESTATE is funded by ANR (PRC) for a total of about 544 000 euros, of which 233 376 euros for DELYS.

Objectives: The core of ESTATE consists in laying the foundations of a new algorithmic framework for enabling Autonomic Computing in distributed and highly dynamic systems and networks. We plan to design a model that includes the minimal algorithmic basis allowing the emergence of dynamic distributed systems with self-* capabilities, *e.g.*, self-organization, self-healing, self-configuration, self-management, self-optimization, self-adaptiveness, or self-repair. In order to do this, we consider three main research streams:

(i) building the theoretical foundations of autonomic computing in dynamic systems, (ii) enhancing the safety in some cases by establishing the minimum requirements in terms of amount or type of dynamics to allow some strong safety guarantees, (iii) providing additional formal guarantees by proposing a general framework based on the Coq proof assistant to (semi-)automatically construct certified proofs.

The coordinator of ESTATE is Franck Petit.

RainbowFS - (2016–2021)

Members: LIP6 (DELYS, project leader), Scality SA, CNRS-LIG, Télécom Sud-Paris, Université Savoie-Mont-Blanc.

Funding: is funded by ANR (PRC) for a total of 919 534 euros, of which 359 554 euros for DELYS.

Objectives: RainbowFS proposes a "just-right" approach to storage and consistency, for developing distributed, cloud-scale applications. Existing approaches shoehorn the application design to some predefined consistency model, but no single model is appropriate for all uses. Instead, we propose tools to co-design the application and its consistency protocol. Our approach reconciles the conflicting requirements of availability and performance vs. safety: common-case operations are designed to be asynchronous; synchronisation is used only when strictly necessary to satisfy the application's integrity invariants. Furthermore, we deconstruct classical consistency models into orthogonal primitives that the developer can compose efficiently, and provide a number of tools for quick, efficient and correct cloud-scale deployment and execution. Using this methodology, we will develop an entreprise-grade, highly-scalable file system, exploring the rainbow of possible semantics, and we demonstrate it in a massive experiment.

The coordinator of RainbowFS is Marc Shapiro.

SeMaFoR - (2021–2024)

Members: LS2N-IMT Atlantique (project leader), LIP6 (DELYS), AlterWay.

Funding: is funded by ANR (PRCE) for a total of 506 787 euros, of which 157 896 euros for DELYS.

Objectives: The goal is to propose an autonomic Fog system designed in a generic way. To this end, we will address several open challenges: 1) Provide an Architecture Description Language (ADL) for modeling Fog systems and their specific features such as the locality concept, QoS constraints applied on resources and their dependencies, the dynamicity of considered workloads, etc. This ADL should be generic and customizable to address any possible kind of Fog system. 2) Support collaborative decision-making between a fleet of small autonomic controllers distributed over the Fog. Tackling the convergence of local decisions to obtain a shared and consistent decision among these autonomic controllers requires new distributed agreement protocols based on distributed consensus algorithms. 3) Support the automatic generation and coordination of reconfiguration plans between the autonomic controllers. Even if each controller gets a new local target configuration to apply from the consensus, the execution plan of the overall reconfiguration needs to

be generated and coordinated to minimize the disruption time and avoid failures. 4) Design and implement a fully open source framework usable in a standalone way or integrated with standard solutions (e.g., Kubernetes). The project targets in particular the future generation of Fog architects, DevOps engineers. We plan to evaluate the solution both on simulated Fog infrastructures as well as real infrastructures.

The local coordinator of SeMaFor in Delys is Jonathan Lejeune.

9 Dissemination

9.1 Promoting scientific activities

9.1.1 Scientific events: organisation

Marc Shapiro organised a series of seminars on the *Loi de programmation pluriannuelle de la recherche* (French bill organising the next 10 years of publicly-funded research) open to all scholars in Informatics. Speakers: Sebastian Stride (SIRIS Academic), Antoine Petit (head of CNRS), Patrick Lemaire (leader of the assembly of French learned societies), Christine Musselin (instituts d'Études Politiques, Paris), Pierre Ouzoulias (CNRS, senator, member of OPECST).

9.1.2 Scientific events: selection

Member of the conference program committees

- Luciana Arantes, The 20th IEEE International Symposium on Network Computing and Applications (NCA 2020), The 32nd IEEE International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2020), The 9th Workshop on Parallel Programming Models - Special Edition on IoT, Edge/Fog computing: Machine Learning and Security (MPP2020)
- Swan Dubois, 22th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2020)
- Pierre Sens, 22th IEEE Cluster conference (CLUSTER 2020), 30th International Symposium on Software Reliability Engineering (ISSRE 2020), 20th IEEE International Symposium on Network Computing and Applications (NCA 2020).
- Marc Shapiro, the European Conference on Computer Systems (EuroSys 2021).

9.1.3 Journal

Member of the editorial boards

- Pierre Sens, International Journal of High Performance Computing and Networking (IJHPCN)

Reviewer - reviewing activities Transaction on Computers (P. Sens), Journal of Parallel and Distributed Computing (L. Arantes, P. Sens), Theoretical Computer Science (S. Dubois), Transactions on Parallel and Distributed Computing (M. Shapiro, ×2).

9.1.4 Invited talks

- Pierre Sens, invited keynote speaker, "Fault Tolerance in Dynamic Distributed Systems", The 19th IEEE International Symposium on Network Computing and Applications (NCA 2020).
- Marc Shapiro, invited keynote speaker to Conférence francophone d'informatique en Parallélisme, Architecture et Système (Compas). Planned in Lyon, June 2020; canceled due to Covid.

9.1.5 Leadership within the scientific community

Marc Shapiro, Vice President for Research of *Société informatique de France* (SIF), the French learned society in Informatics.

9.1.6 Research administration

- Colette Johnen, since 2020: Member of section 27 of Conseil national des Universités.
- Pierre Sens, since 2016: Member of Section 6 of the national committee for scientific research CoNRS.
- Franck Petit, Pierre Sens, since 2012: Member of the Executive Committee of Labex SMART, Co-Chairs of Track 4, Autonomic Distributed Environments for Mobility.

9.2 Teaching - Supervision - Juries

9.2.1 Teaching

- Julien Sopena is Member of "Directoire des formations et de l'insertion professionnelle" of Sorbonne Université, France
- Master: Julien Sopena is responsible of Computer Science Master's degree in Distributed systems and applications (in French, SAR), Sorbonne Universités, France
- Master: Luciana Arantes, Swan Dubois, Jonathan Lejeune, Franck Petit, Pierre Sens, Julien Sopena, Advanced distributed algorithms, M2, Sorbonne Université, France
- Master: Jonathan Lejeune, Designing Large-Scale Distributed Applications, M2, Sorbonne Université, France
- Master: Maxime Lorrillere, Julien Sopena, Linux Kernel Programming, M1, Sorbonne Université, France
- Master: Luciana Arantes, Swan Dubois, Jonathan Lejeune, Pierre Sens, Julien Sopena, Operating systems kernel, M1, Sorbonne Université, France
- Master: Luciana Arantes, Swan Dubois, Franck Petit, Distributed Algorithms, M1, Sorbonne Université, France
- Master: Franck Petit, Autonomic Networks, M2, Sorbonne Université, France
- Master: Franck Petit, Distributed Algorithms for Networks, M1, Sorbonne Université, France
- Master: Jonathan Lejeune, Julien Sopena, Client-server distributed systems, M1, Sorbonne Université, France.
- Master: Luciana Arantes, Pierre Sens, Franck Petit. Cloud Computing, M1, EIT Digital Master, France.
- Master: Julien Sopena, Marc Shapiro, Ilyas Toumlilt, Francis Laniel. Kernels and virtual machines (*Noyaux et machines virtuelles*, NMV), M2, Sorbonne Université, France.
- Licence: Pierre Sens, Luciana Arantes, Julien Sopena, Principles of operating systems, L3, Sorbonne Université, France
- Licence: Swan Dubois, Initiation to operating systems, L3, Sorbonne Université, France
- Licence: Swan Dubois, Multi-threaded Programming, L3, Sorbonne Université, France
- Licence: Jonathan Lejeune, Oriented-Object Programming, L3, Sorbonne Université, France
- Licence: Franck Petit, Advanced C Programming, L2, Sorbonne Université, France
- Licence: Swan Dubois, Jonathan Lejeune, Julien Sopena, Introduction to operating systems, L2, Sorbonne Université, France
- Licence: Mesaac Makpangou, C Programming Language, 27 h, L2, Sorbonne Université, France

- Ingénieur 4ème année : Marc Shapiro, Introduction aux systèmes d'exploitation, 26 h, M1, Polytech Sorbonne Université, France.
- Licence : Philippe Darche (coordinator), Architecture of Internet of Things (IoT), 2 × 32h, L3, Institut Universitaire Technologique (IUT) Paris Descartes, France.
- Engineering School: Philippe Darche (coordinator), Solid-State Memories, 4th year, ESIEE, France.
- DUT: Philippe Darche (coordinator), Introduction to Computer Systems - Data representation, 60h, Institut Universitaire Technologique (IUT) Paris Descartes, France.
- DUT: Philippe Darche (coordinator), Computer Architecture, 32h, Institut Universitaire Technologique (IUT) Paris Descartes, France.
- DUT: Philippe Darche (coordinator), Computer Systems Programming, 80h, Institut Universitaire Technologique (IUT) Paris Descartes, France.

9.2.2 Supervision

- CIFRE PhD: Guillaume Fraysse, Orange Lab - Inria, "Ubiquitous Resouces for Service Availability", Dec. 2020. Advised by Pierre Sens, Imen Grida Ben Yahia (Orange-Lab) , Jonathan Lejeune, Julien Sopena.
- PhD : Francis Laniel, "Vers une utilisation efficace de la mémoire non volatile pour économiser l'énergie."Sorbonne Univ., Dec. . 2020. Advised by Marc Shapiro, Julien Sopena, Jonathan Lejeune.
- CIFRE PhD in progress: José Alves Esteves, "Adaptation dynamique en environnements répartis contraints", Sorbonne Univ., since Sep. 2019. Advised by Pierre Sens and Amina Boubendir Orange Labs.
- PhD in progress: Arnaud Favier, "Algorithmes de coordination répartis dans des réseaux dynamiques", Sorbonne Univ., since Sep. 2018. Advised by Pierre Sens and Luciana Arantes.
- PhD in progress: Célia Mahamdi, "Prise de décision collaborative dans un système distribué et dynamique", Sorbonne Univ., since Sep. 2020. Advised by Mesaac Makpongou and Jonathan Lejeune.
- PhD in progress: Saalik Hatia, "Efficient management of memory and storage for CRDTs", Sorbonne Univ., since Oct. 2018. Advised by Marc Shapiro.
- PhD in progress: Gabriel Le Bouder, "Autonomic synchronization", Sorbonne Univ., since Sep. 2019. Advised by Franck Petit.
- PhD in progress: Benoît Martin, "Protocol de cohérence hybride: de la cohérence causal à la cohérence forte", Sorbonne Univ., since Sep. 2019. Advised by Mesaac Makpangou and Marc Shapiro.
- PhD in progress: Sreeja Nair, "Just-Right Consistency for massive gee-replicated storage", Sorbonne Univ., since Apr. 2018. Advised by Marc Shapiro.
- PhD in progress: Laurent Prosperi, "Abstractions, langage et runtime pour les systèmes distribués", Sorbonne Univ., since Sep. 2019. Advised by Marc Shapiro.
- CIFRE PhD in progress: Jonathan Sid-Otmane. "Étude des critères de distribution et de l'usage d'une base de données distribuée pour un OS Telco", since Dec. 2017. Advised by Marc Shapiro, with Sofiane Imadali and Frédéric Martelli, Orange Labs.
- PhD in progress: Ilyas Toumlilt, "Bridging the CAP gap, all the way to the edge", Sorbonne Univ., since Sep. 2016. Advised by Marc Shapiro.
- CIFRE PhD in progress: Dimitrios Vasilas, "Indexing in large-scale storage systems", Sorbonne Univ., since Sep. 2016. Advised by Marc Shapiro, with Brad King, Scality.

- PhD in progress: Daniel Wladdimiro, “Adaptation dynamique en environnements répartis contraints”, Sorbonne Univ., since Sep. 2019. Advised by Pierre Sens and Luciana Arantes.
- PhD in progress: Daniel Wilhelm, “Algorithmes de diffusion causale dans les systèmes répartis dynamique”, Sorbonne Univ., since Oct. 2019, Pierre Sens and Luciana Arantes.

9.2.3 Juries

Pierre Sens was the reviewer of:

- Anne-Cécile Orgerie, HDR, IRISA, Univ. Rennes
- Mohand Mezmaz, HDR, CRISTAL, Univ. Lille
- Nikos Parlavantzas, HDR, Inria, Sorbonne Univ.
- Maha Alsayasneh, PhD, LIG, Univ. Grenoble
- Mathieu Bacou, PhD, IRIT, Toulouse

Pierre Sens was Chair of

- Yifan Du, PhD, Inria Paris, Sorbonne Univ.
- Giovania Farina, PhD, LIP6, Sorbonne Univ.
- Francis Laniel, PhD, LIP6, Sorbonne Univ.
- Andrea Petreto, PhD, LIP6, Sorbonne Univ.

Colette Johnen was the reviewer of

- Marie Laveau, PhD, Paris Saclay.

10 Scientific production

10.1 Major publications

[1] V. Balegas, N. Preguiça, R. Rodrigues, S. Duarte, C. Ferreira, M. Najafzadeh and M. Shapiro. ‘Putting Consistency back into Eventual Consistency’. In: *euroconfon # Comp.\Sys.\(EuroSys)*. Bordeaux, France, Apr. 2015, 6:1–6:16. DOI: 10.1145/2741948.2741972. URL: https://doi.org/10.1145/2741948.2741972.

[2] L. Gidra, G. Thomas, J. Sopena, M. Shapiro and N. Nguyen. ‘NumaGiC: a garbage collector for big data on big NUMA machines’. In: *intconfon # Archi.\Support for Prog.\Lang.\and Systems (ASPLOS)*. Istanbul, Turkey: Assoc.\for Computing Machinery, Mar. 2015, pp. 661–673. DOI: 10.1145/2694344.2694361. URL: http://dx.doi.org/10.1145/2694344.2694361.

[3] A. Gotsman, H. Yang, C. Ferreira, M. Najafzadeh and M. Shapiro. ‘’Cause I’m Strong Enough: Reasoning about Consistency Choices in Distributed Systems’. In: *sympon # Principles of Prog.\Lang.\(POPL)*. St.~Petersburg, FL, USA, 2016, pp. 371–384. DOI: 10.1145/2837614.2837625. URL: http://dx.doi.org/10.1145/2837614.2837625.

[4] J. Peeters, N. Ventroux, T. Sassolas and M. Shapiro. *Distributing computing system implementing a non-speculative hardware transactional memory and a method for using same for distributed computing*. Patent awarded US 10 416 925 B2. United States Patent and Trademark Office (USPTO), Sept. 2019.

[5] M. Shapiro. *Living at the edge, safely*. Blog post. LightKone European Project, May 2019.

[6] M. Shapiro, N. Preguiça, C. Baquero and M. Zawirski. ‘Conflict-free Replicated Data Types’. In: *intsympon # Stabilization, Safety, and Security of Dist.\Sys.\(SSS)*. Ed. by X. Défago, F. Petit and V. Villain. Vol. 6976. Lecture Notes in Comp.\Sc. Grenoble, France: Springer-Verlag, Oct. 2011, pp. 386–400. URL: http://lip6.fr/Marc.Shapiro/papers/CRDTs%5C_SSS-2011.pdf.

[7] A. Z. Tomsic. 'Exploring the design space of highly-available distributed transactions'. PhD thesis. Paris, France: Université Pierre et Marie Curie, Apr. 2018.

[8] M. Zawirski, N. Preguiça, S. Duarte, A. Bieniusa, V. Balegas and M. Shapiro. 'Write Fast, Read in the Past: Causal Consistency for Client-side Applications'. In: *intconfon # Middleware (MIDDLEWARE)*. ACM/IFIP/Usenix. Vancouver, BC, Canada, Dec. 2015, pp. 75–87.

10.2 Publications of the year

International journals

[9] S. Bouchard, Y. Dieudonné, A. Pelc and F. Petit. 'Deterministic Treasure Hunt in the Plane with Angular Hints'. In: *Algorithmica* 82.11 (Nov. 2020), pp. 3250–3281. DOI: 10.1007/s00453-020-00724-4. URL: https://hal.inria.fr/hal-03138288.

[10] A. Casteigts, S. Dubois, F. Petit and J. Robson. 'Robustness: A new form of heredity motivated by dynamic networks'. In: *Theoretical Computer Science* 806 (Feb. 2020), pp. 429–445. DOI: 10.1016/j.tcs.2019.08.008. URL: https://hal.archives-ouvertes.fr/hal-02491886.

[11] K. Censor-Hillel and M. Rabie. 'Distributed Reconfiguration of Maximal Independent Sets'. In: *Journal of Computer and System Sciences* 112 (Sept. 2020), pp. 85–96. DOI: 10.1016/j.jcss.2020.03.003. URL: https://hal.sorbonne-universite.fr/hal-02879023.

[12] T. Heimfarth, J. C. Giacomin, E. Pignaton De Freitas, G. F. Araujo and J. P. de Araujo. 'PAX-MAC: A Low Latency Anycast Protocol with Advanced Preamble †'. In: *Sensors* 20.1 (Jan. 2020), pp. 23–25. DOI: 10.3390/s20010250. URL: https://hal.sorbonne-universite.fr/hal-02479153.

[13] P. Raymond, S. Devismes, A. Lamani, F. Petit and S. Tixeuil. 'Terminating Exploration Of A Grid By An Optimal Number Of Asynchronous Oblivious Robots'. In: *The Computer Journal*. The Computer Journal 64.1 (Jan. 2021), pp. 132–154. DOI: 10.1093/comjnl/bxz166. URL: https://hal.archives-ouvertes.fr/hal-02363013.

[14] L. Teylo, L. Arantes, P. Sens and L. M. A. Drummond. 'A dynamic task scheduler tolerant to multiple hibernations in cloud environments'. In: *Cluster Computing* (Sept. 2020). DOI: 10.1007/s10586-020-03175-2. URL: https://hal.inria.fr/hal-03136616.

National journals

[15] P. Darche. 'Evolution of solid-state random access memories - version 2'. In: *Techniques de l'Ingenieur* (10th June 2020). URL: https://hal.archives-ouvertes.fr/hal-03120674.

International peer-reviewed conferences

[16] K. Altisen, S. Devismes, A. Durand, C. Johnen and F. Petit. 'Brief Announcement: Self-stabilizing Systems in Spite of High Dynamics'. In: PODC 2020 - ACM Symposium on Principles of Distributed Computing. 227-229. Salerne / Virtual, Italy, 3rd Aug. 2020. DOI: 10.1145/3382734.3404502. URL: https://hal.archives-ouvertes.fr/hal-02911071.

[17] K. Altisen, S. Devismes, A. Durand, C. Johnen and F. Petit. 'Self-stabilizing Systems in Spite of High Dynamics'. In: 22nd International Conference on Distributed Computing and Networking, ICDCN'21. ICDCN '21: International Conference on Distributed Computing and Networking 2021. Nara, Japan: http://www.icdcn2021.net, Jan. 2021, pp. 156–165. DOI: 10.1145/3427796.3427838. URL: https://hal.archives-ouvertes.fr/hal-02376832.

[18] J. J. Alves Esteves, A. Boubendir, F. Guillemin and P. Sens. 'Heuristic for Edge-enabled Network Slicing Optimization using the "Power of Two Choices"'. In: CNSM 2020 - 16th International Conference on Network and Service Management. Izmir / Virtual, Turkey, 2nd Nov. 2020. URL: https://hal.inria.fr/hal-02981120.

[19] J. J. Alves Esteves, A. Boubendir, F. Guillemin and P. Sens. 'Optimized Network Slicing Proof-of-Concept with Interactive Gaming Use Case'. In: ICIN 2020 - 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops. Paris, France, 24th Feb. 2020, pp. 150–152. DOI: 10.1109/ICIN48450.2020.9059328. URL: https://hal.inria.fr/hal-02981083.

[20] J. J. Alves Esteves, A. Boubendir, F. Guillemin and P. Sens. 'Edge-enabled Optimized Network Slicing in Large Scale Networks'. In: NoF 2020 - 11th International Conference on Network of the Future. Bordeaux / Virtual, France, 12th Oct. 2020. URL: https://hal.inria.fr/hal-02981108.

[21] J. J. Alves Esteves, A. Boubendir, F. Guillemin and P. Sens. 'Location-based Data Model for Optimized Network Slice Placement'. In: NetSoft 2020 - 6th IEEE International Conference on Network Softwarization. Ghent / Virtual, Belgium, 29th June 2020, pp. 404–412. DOI: 10.1109/NetSoft48620.2020.9165427. URL: https://hal.inria.fr/hal-02981095.

[22] L. Blin, S. Dubois and L. Feuilloley. 'Silent MST Approximation for Tiny Memory'. In: SSS 2020 : The 22th International Symposium on Stabilization, Safety, and Security of Distributed Systems. Vol. 12514. Lecture Notes in Computer Science. Austin, TX / Virtual, United States, 18th Nov. 2020, pp. 118–132. DOI: 10.1007/978-3-030-64348-5_10. URL: https://hal.inria.fr/hal-03140584.

[23] S. Bouchard, Y. Dieudonné and A. Pelc. 'Want to Gather? No Need to Chatter!' In: PODC '20 - 39th Symposium on Principles of Distributed Computing. Salerno / Virtual, Italy, 31st July 2020, pp. 253–262. DOI: 10.1145/3382734.3405693. URL: https://hal.inria.fr/hal-03138303.

[24] S. Bouchard, Y. Dieudonné, A. Pelc and F. Petit. 'Almost Universal Anonymous Rendezvous in the Plane'. In: *SPAA '20: Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*. SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures. Virtual Event, United States, July 2020, pp. 117–127. DOI: 10.1145/3350755.3400283. URL: https://hal.inria.fr/hal-03138344.

[25] L. Corrêa, L. Arantes, P. Sens, M. Inostroza-Ponta and M. Dorn. 'A dynamic evolutionary multi-agent system to predict the 3D structure of proteins'. In: WCCI 2020 - IEEE World Congress on Evolutionary Computation - CEC Sessions. Glasgow / Virtual, United Kingdom, 19th July 2020, pp. 1–8. DOI: 10.1109/CEC48606.2020.9185761. URL: https://hal.inria.fr/hal-03132137.

[26] A. Favier, N. Guittonneau, L. Arantes, A. Fladenmuller, J. Lejeune and P. Sens. 'Topology Aware Leader Election Algorithm for Dynamic Networks'. In: PRDC 2020 - 25th IEEE Pacific Rim International Symposium on Dependable Computing. 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC). Perth, Australia: https://prdc.dependability.org/PRDC2020/, 14th Jan. 2021, pp. 1–10. DOI: 10.1109/PRDC50213.2020.00011. URL: https://hal.archives-ouvertes.fr/hal-02954037.

[27] G. Fraysse, J. Lejeune, J. Sopena and P. Sens. 'A resource usage efficient distributed allocation algorithm for 5G Service Function Chains'. In: DAIS 2020 - 20th IFIP WG 6.1 International Conference Distributed Applications and Interoperable Systems. Vol. 12135. Lecture Notes in Computer Science. Valetta, Malta, 15th June 2020, pp. 169–185. DOI: 10.1007/978-3-030-50323-9_11. URL: https://hal.archives-ouvertes.fr/hal-02975998.

[28] R. Gouicem, D. Carver, J.-P. Lozi, J. Sopena, B. Lepers, W. Zwaenepoel, N. Palix, J. Lawall and G. Muller. 'Fewer Cores, More Hertz: Leveraging High-Frequency Cores in the OS Scheduler for Improved Application Performance'. In: 2020 USENIX Annual Technical Conference. Boston / Virtual, United States: https://www.usenix.org/conference/atc20/technical-sessions, 15th July 2020. URL: https://hal.inria.fr/hal-02901169.

[29] ***Best Paper***
F. Laniel, D. Carver, J. Sopena, F. Wajsburt, J. Lejeune and M. Shapiro. 'MemOpLight: Leveraging application feedback to improve container memory consolidation'. In: NCA 2020 - 19th IEEE International Symposium on Network Computing and Applications. Cambridge / Virtual, United States, 24th Nov. 2020, pp. 1–10. DOI: 10.1109/NCA51143.2020.9306717. URL: https://hal.archives-ouvertes.fr/hal-03065629.

[30] B. Lepers, R. Gouicem, D. Carver, J.-P. Lozi, N. Palix, M.-V. Aponte, W. Zwaenepoel, J. Sopena, J. Lawall and G. Muller. 'Provable Multicore Schedulers with Ipanema: Application to Work Conservation'. In: Eurosys 2020 - European Conference on Computer Systems. Heraklion / Virtual, Greece, 27th Apr. 2020. DOI: 10.1145/3342195.3387544. URL: https://hal.inria.fr/hal-02554342.

[31] S. S. Nair, G. Petri and M. Shapiro. 'Proving the safety of highly-available distributed objects'. In: ESOP 2020 - 29th European Symposium on Programming. Dublin, Ireland, 25th Apr. 2020. URL: https://hal.archives-ouvertes.fr/hal-02424317.

[32] J. Sid-Otmane, S. Imadali, F. Martelli and M. Shapiro. 'Data Consistency in the 5G Specification'. In: ICIN 2020 - 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops. Paris, France, 24th Feb. 2020, pp. 110–117. DOI: 10.1109/ICIN48450.2020.9059408. URL: https://hal.archives-ouvertes.fr/hal-02943802.

[33] V. Vallade, L. Le Frioux, S. Baarir, J. Sopena, V. Ganesh and F. Kordon. 'Community and LBD-Based Clause Sharing Policy for Parallel SAT Solving'. In: SAT 2020 - 23rd International Conference on Theory and Applications of Satisfiability Testing. Vol. 12178. Lecture Notes in Computer Science. Alghero / Virtual, Italy, 26th June 2020, pp. 11–27. DOI: 10.1007/978-3-030-51825-7_2. URL: https://hal.inria.fr/hal-02906505.

[34] V. Vallade, L. Le Frioux, S. Baarir, J. Sopena and F. Kordon. 'On the Usefulness of Clause Strengthening in Parallel SAT Solving'. In: NFM 2020 - 12th NASA Formal Methods Symposium. Moffett Field / Virtual, United States, 11th May 2020. URL: https://hal.archives-ouvertes.fr/hal-02545756.

National peer-reviewed Conferences

[35] K. Altisen, S. Devismes, A. Durand, C. Johnen and F. Petit. 'Élection Autostabilisante dans les Réseaux à Haute Dynamicité'. In: ALGOTEL 2020 – 22èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications. Lyon, France, 29th Sept. 2020. URL: https://hal.archives-ouvertes.fr/hal-02791667.

[36] G. Bu, M. Potop-Butucaru and M. Rabie. 'Diffusion dans les réseaux sans fil en utilisant des filtres à mémoire constante'. In: ALGOTEL 2020 – 22èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications. Lyon, France, 29th Sept. 2020. URL: https://hal.archives-ouvertes.fr/hal-02867634.

Conferences without proceedings

[37] B. Martin, L. Prosperi and M. Shapiro. 'An environment for composable distributed computing'. In: EuroDW 2020 - 14th EuroSys Doctoral Workshop. Heraklion / Virtual, Greece, 27th Apr. 2020. URL: https://hal.inria.fr/hal-03146124.

[38] D. Vasilas, M. Shapiro, B. King and S. S. Hamouda. 'Towards application-specific query processing systems'. In: BDA 2020 - 36ème Conférence sur la Gestion de Données – Principes, Technologies et Applications. Paris / Virtual, France, 27th Oct. 2020. URL: https://hal.archives-ouvertes.fr/hal-02943380.

Scientific books

[39] P. Darche. *Microprocessor 1: Prolegomenes - Calculation and Storage Functions - Models of Computation and Computer Architecture*. 2nd Nov. 2020. URL: https://hal.archives-ouvertes.fr/hal-03120694.

[40] P. Darche. *Microprocessor 2. Core Concepts: Communication in a Digital System*. 1st Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03120700.

[41] P. Darche. *Microprocessor 3. Core Concepts: Hardware Aspects*. 1st Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03120707.

[42] P. Darche. *Microprocessor 4. Core Concepts: Software Aspects*. 1st Feb. 2021. URL: https://hal.archives-ouvertes.fr/hal-03120713.

[43] P. Darche. *Microprocessor 5. Software and Hardware Aspects of Development, Debugging and Testing – The Microcomputer*. 1st Feb. 2021. URL: https://hal.archives-ouvertes.fr/hal-03120718.

Doctoral dissertations and habilitation theses

[44] G. Fraysse. 'Distributed resource allocation for virtual networks'. Sorbonne Université, CNRS, LIP6, Paris, France, 18th Dec. 2020. URL: https://tel.archives-ouvertes.fr/tel-03128234.

[45] F. Laniel. 'MemOpLight: toward memory consolidation for containers thanks to application feedbackm'. Ecole Doctorale Informatique, Télécommunications et Electronique, 9th Nov. 2020. URL: https://tel.archives-ouvertes.fr/tel-03144835.

Reports & preprints

[46] A. Amamou, M. Camey, C. Cérin, J. Rivalan and J. Sopena. *Resources management for controlling dynamic loads in clouds environments. The Wolphin project experience*. Université Sorbonne Paris Nord; Sorbonne Université, 17th Feb. 2020. URL: https://hal.archives-ouvertes.fr/hal-02481264.

[47] S. Bouchard, Y. Dieudonné and A. Pelc. *Want to Gather? No Need to Chatter!* Université de Picardie Jules Verne, 2nd Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03035137.

[48] S. Bouchard, Y. Dieudonné, A. Pelc and F. Petit. *Almost Universal Anonymous Rendezvous in the Plane*. Université de Picardie Jules Verne, 2nd Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03035154.

[49] G. Bu, Z. Lotker, M. Potop-Butucaru and M. Rabie. *Lower and upper bounds for deterministic convergecast with labeling schemes*. Sorbonne Université, 29th May 2020. DOI: 10.4230/LIPIcs...23. URL: https://hal.archives-ouvertes.fr/hal-02650472.

[50] G. Bu, M. Potop-Butucaru and M. Rabie. *Wireless Broadcast with short labelling*. 26th Jan. 2020. URL: https://hal.archives-ouvertes.fr/hal-01869563.

[51] S. Hatia and M. Shapiro. *Specification of a Transactionally and Causally-Consistent (TCC) database*. DELYS; LIP6, Sorbonne Université, Inria, Paris, France, 20th July 2020. URL: https://hal.inria.fr/hal-02902474.

[52] C. Johnen and M. Haddad. *Efficient self-stabilizing construction of disjoint MDSs in distance-2 model*. Inria Paris, Sorbonne Université; LaBRI, CNRS UMR 5800; LIRIS UMR CNRS 5205, 11th Feb. 2021. URL: https://hal.archives-ouvertes.fr/hal-03138979.

[53] S. S. Nair, F. Meirim, M. Pereira, C. Ferreira and M. Shapiro. *A coordination-free, convergent, and safe replicated tree*. LIP6, Sorbonne Université, Inria, Paris, France; Universidade nova de Lisboa, 23rd Feb. 2021. URL: https://hal.archives-ouvertes.fr/hal-03150817.

[54] S. S. Nair, G. Petri and M. Shapiro. *Proving the safety of highly-available distributed objects (Extended version)*. LIP6, Sorbonne Université, Inria, Paris, France; Arm Research, Cambridge, UK, 27th Feb. 2020. URL: https://hal.archives-ouvertes.fr/hal-02492599.

[55] D. Wilhelm, L. Arantes and P. Sens. *A scalable causal broadcast that tolerates dynamics of mobile networks*. Sorbonne University , UPMC, 29th May 2020. URL: https://hal-upec-upem.archives-ouvertes.fr/hal-02652082.