INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team ADEPT

Algorithms for Dynamic Dependable Systems

Rennes - Bretagne-Atlantique

THEME COM

Activity Report

2008

Table of contents

1. Team **1**

2. Overall Objectives **1**

2.1. Introduction 1

2.2. Highlight of the year 2

3. Scientific Foundations **2**

3.1. Introduction 2

3.2. Models for Dynamic, Scalable and Dependable Systems 3

3.3. Design and Uses of Group Communication Services 4

3.3.1. Dependability within Small Distributed Systems 4

3.3.2. Group Communication Services 4

3.3.3. Security of a Web Access in the event of New Attacks 5

3.3.4. Task Allocation in a Grid in the event of Benign Faults 5

3.4. Reputation and Self-organization in Large Scale Systems 6

3.4.1. Reputation Mechanisms Robust against Attacks and Threats 6

3.4.2. Hierarchical Specification of Self-organization in Dynamic Systems 7

4. Application Domains **7**

4.1. Space Domain Applications 7

4.2. Telecommunication Applications 8

5. Software **8**

6. New Results **9**

6.1. Agreement in Distributed Systems 9

6.1.1. Towards automatic proofs of Impossibility Results in Wait-Free Systems 9

6.1.2. Group Communication Services to Secure a Web Access 10

6.1.3. Group Communication Services to Allocate Tasks in a Grid 10

6.2. Reputation and Self-organization in Dynamic Large Scale Systems 11

6.2.1. Reputation Mechanisms 11

6.2.2. Specification of Self-organization 12

7. Other Grants and Activities **12**

7.1. National Project 12

7.1.1. ACI Daddi (2004-2008) 12

7.1.2. RIAM Project (ANR): Solipsis (2006-2008) 13

7.1.3. DGE Project: P2Pim@ges (2007-2009) 13

7.2. International Cooperations 13

8. Dissemination **14**

8.1. Teaching Activities 14

8.2. Presentations of Research Works 14

8.3. Integration within the Scientific Community 14

9. Bibliography **15**

ADEPT *is a joint project-team with CNRS and the University of Rennes I.*

1. Team

Research Scientist

Michel Hurfin [Team Leader, Research Scientist, INRIA, HdR]
Emmanuelle Anceaume [Research Scientist, CNRS]

Faculty Member

Frédéric Tronel [Faculty Member, Univ. Rennes 1, until september 2008]
Jean-Pierre Le Narzul [External Collaborator, Faculty Member, Institut Télécom / Telecom Bretagne]
Aina Ravoaja [PhD Student, Univ. Rennes 1, ATER ENS, until july 2008]
Frédéric Majorczyk [Faculty Member, ATER Univ. Rennes 1, since september 2008]

Technical Staff

Romaric Ludinard [Technical Staff, INRIA]

PhD Student

Izabela Moise [PhD Student, Univ. Rennes 1, since october 2008]
Heverson Ribeiro [PhD Student, Univ. Rennes 1]

Visiting Scientist

Carlos Maziero [Associate Professor, Pontifical Catholic University of Parana State - Brazil, until february 2008]

Administrative Assistant

Lydie Mabil [Administrative Assistant, INRIA, until febuary 2008]
Laétitia Rihet [Administrative Assistant, INRIA, until february 2008]
Sara Freoul [Administrative Assistant, INRIA, from february to april 2008]
Elodie Besnard [Administrative Assistant, INRIA, since May 2008]

2. Overall Objectives

2.1. Introduction

The field of information technologies in general, and distributed computing in particular, is continuously evolving and maturing at a very high pace. The characteristics of networks and connected entities have progressed so much that these improvements have induced radical changes in the very nature of applications that can be built upon these distributed systems. Many applications are now executed on large scale systems using a huge number of computer resources spread all over the world. While computer-based systems are becoming more and more open and complex (number of interacting nodes, dynamicity, heterogeneity of the hardware and software components, mixing of various standards, use of unreliable components, presence of malicious nodes, ...), the main attributes of dependability (namely reliability, availability, integrity, and confidentiality) are also more and more difficult to guarantee. Although the occurrence of accidental and intentional failures tend to be more frequent and severe, most applications require to inhibit (or at least to limit) their possible consequences.

In the context of distributed applications and systems, the ADEPT team is focusing on dependability and consistency issues. Our goal is to propose models and algorithms that allow to cope with dynamic changes of the system's composition whatever their causes (failure, connection/disconnection, mobility, ...).

The design of dependable mechanisms mainly depends on the types of faults that might occur during the computation. Benign faults (crash, omission, ...) are distinguished from the arbitrary faults (Byzantine faults). In the former case, processes behave according to their specification but after some time they may omit some (or all) computation steps. In the latter case, processes involved in the computation may arbitrarily deviate from their specification. Such faults can be the consequence of malicious intents of individuals.

When a low level of dynamicity (also called churn) is assumed or when the system size is rather small, a process involved in a distributed computation may know and observe all the other participants. In that case, if all the participants can share a common knowledge of the group of interacting processes, various fundamental problems (related to observation and synchronization) can be solved easily. Assuming that variations do not occur very often, adaptive algorithms can be proposed to detect a modification of the whole execution context and react globally to this modification (reconfiguration, execution of another code, ...). In particular, to cope with the dynamic evolution of a distributed system, the Group paradigm (and the associated concept of membership service) allows to efficiently address dependability issues. Solutions to agreement problems (such as the consensus problem) can be used as basic building blocks for designing solutions to higher level protocols that are in charge of maintaining global properties at the group level despite the occurrence of faults within the group. Due to the increasing adversity of the system (asynchrony and failures), the design of efficient solutions that are simple to deploy and easy to adapt remains a difficult issue.

When the system has a very high level of dynamicity, implementing a global observation mechanism that allows to reconfigure the whole system in a single step is no more realistic. Only local observations and progressive adaptations to changes can be performed on cohesive subsets of nodes. Such a radical gap on the scale and dynamicity of systems militates in favor of a paradigm shift for designing solutions to the problems raised by these new systems. Several partial and inconsistent views of the system may coexist (each participant may have its own view). All classical distributed computing problems (for example, dependability issues, communication problems, resource allocation, and data management) require new solutions that address theses challenges in the new settings. In this context we consider mainly reputation mechanisms in P2P systems, and to a lesser extent failures in sensor networks.

2.2. Highlight of the year

The first version of our set of communication services called PROMETEUS has been released. The design and the development of these agreement services have been completed in the context of the Daddi project.

3. Scientific Foundations

3.1. Introduction

Our scientific contributions aim to reach a deeper understanding of some fundamental problems that arise in dynamic distributed systems prone to accidental/intentional failures. We consider mainly problems corresponding to middleware services that need to be correctly and continuously provided to the upper-layer entities despite the occurrence of faults.

During the study of a particular problem, we aim to design, for a particular execution environment (characterized by a set of assumptions on the computation model, the failure model, the dynamicity, the scalability, ...), efficient algorithmic solutions that are optimal and generic if possible. If no solution exists, we aim at exhibiting impossibility results. To validate and to promote the use of these algorithmic solutions, we conduct in parallel experimental evaluations by developing flexible and adaptive middleware services that integrate our know-how and experience in distributed computing. This prototyping activity leads us to consider technical and operational problems as well as methodological issues. The feed-back that we get helps us to define new directions in our research activity.

The aim of the ADEPT project is first to propose models for dynamic, scalable and dependable systems, then to identify, specify and design a set of generic elementary services needed to build dependable distributed applications for these systems. More precisely, our contribution focuses on the following themes:

- **Models for dynamic, scalable and dependable systems.** The new complexities faced by the research community in distributed computing (i.e. large scale, dynamicity, dependability) need adequate formal models. These models should include new abstractions for the communication and frameworks for the system execution.

- **Group communication** Economic activities and human lives are now heavily dependent on distributed systems and applications. When computing resources and stored data can be affected by the occurrence of failures dependability becomes a crucial issue. We aim to consider both accidental and intentional faults and to design algorithms and methods to detect or to mask such faults which are sometimes transient (another dynamic aspect). An important part of our activity is dedicated to the study of agreement problems and to their use in group communication services. Replication technics and consequently any service that allows to managed consistency within small group of replicas are of upmost importance to build critical components. In particular, we investigate the use of group communication services to secure a web server and to allocate task in a Grid.
- **Large scale distributed systems.** We consider different types of large scale systems and study the main dependability issues that are associated. In peer-to-peer systems, we address the reputation problem: the proposed mechanisms aim at evaluating the trust level of each entity in the system.

3.2. Models for Dynamic, Scalable and Dependable Systems

Keywords: *automata model*, *geometric model*, *thermodynamic model*.

We investigate the use of new models (geometrical and stochastic models) for automatic proving of impossibility results and lower bounds in the field of asynchronous distributed systems where processes are prone to crash.

Defining models for distributed computing is a matter of compromise. One needs to define properties that are sufficiently precise to capture the expected behavior of systems under study, while at the same time maintaining consistency and tractability of the model. There is no point in defining extremely detailed rules if one cannot manage inherent complexity induced by a too great level of details.

Till a recent past, two main models are used for studying distributed systems and proving theorems about them. The first and oldest one is an extension of automata for distributed computing. It is widely used in the community [33], [27], [22], [21]. The second one, which is more recent, is based on rather involved mathematical developments from the field of algebraic topology. This second model is qualified as a *geometrical model*. In addition to these two major models, the *thermodynamic* models can also be of interest when one considers large scale dynamic systems.

Based on the first model, Fischer, Lynch, and Paterson have proved in 1985 [27] that the consensus is impossible to solve in an asynchronous distributed system, if at least one process may crash. In 1998, using the same model, Biran, Moran and Zaks [18] have been able to precisely determine the conditions under which a decision problem has a solution in a totally asynchronous system where at most one process can fail, but they were not able to extend this result to a larger number of failures. Indeed, by its very nature, a model based on graphs and automata seems to be not well adapted to determine the frontier between possible and impossible problems with respect to fault-tolerance, when more than one process may crash in a system.

A geometric model is more suited to address such problems. Rather than a planar representation of systems, this model provides objects that span over high dimensional spaces. To study these complex geometrical objects that are used to represent the specification of a problem and the possible computation steps, mathematical tools originating from algebraic topology are commonly used. This field of mathematics offers powerful techniques that are able to tackle the combinatorial explosion that occurs when using the graph theoretical approach. This has been brilliantly demonstrated by Herlihy and Shavit [31], who have been able to precisely characterize the set of decision tasks that can be solved in wait-free systems. Thus, thanks to this new model, these authors have been able to extend previous results to the case where an unbounded number of processes may fail. Using also a geometric model, Borowsky and Gafni have latter introduced a new communication primitive called *immediate snapshot* [19], whose semantics is specifically designed so that computability in this new model can be easily established (by simple geometrical considerations). The necessary and sufficient set of conditions established by Borowsky and Gafni for this model of computation is exactly the same as the one established by Herlihy and Shavit. Then, Borowsky and Gafni have exhibited two distributed algorithms [20] that respectively implement the immediate snapshot communication primitive in the shared memory model, and conversely. This precisely shows that both models of computation are equivalent.

The two traditional models that have been previously illustrated do not seem to completely fulfill the requirements of dynamicity imposed by large scale systems. To overcome these limitations, models inspired by some technics of statistical physics may allow to study large-scale system as a whole without looking at all possible local interactions. This approach is directly inspired by the way physicists are looking at complex thermodynamic systems. While they do not consider every single interactions between microscopic particles that compose these kind of systems, they can still give a good description of its macroscopic behavior in terms of macroscopic quantities such as its temperature, its free energy, enthalpy and so on.

3.3. Design and Uses of Group Communication Services

Keywords: *agreement problem*, *dependability*, *group communication*, *intrusion detection*.

3.3.1. Dependability within Small Distributed Systems

We target small distributed systems that contain less than a few tens of processors. In such a context, we aim at specifying and designing methods to cope with the different attributes of dependability when either accidental faults or intentional faults may occur during operation. Design and implementation faults are out of the scope of our research activities: by assumption, any software is supposed to be correct with respect to its specification. To solve these dependability issues, we promote the use of group communication services that are themself based on a set of agreement protocols.

Many distributed protocols are designed and proved to be correct under the assumption that the protocol is executed by a group of processes whose composition is known by each of the participants. In that case, both the size of the system and the identities of the participants can be defined as constants in the code. When failures may occur, additional quantities (such as the maximal number of processes that may fail and the number of processes that effectively failed) can also be considered: they will be referred either in the algorithmic solutions, in the assumptions used to characterize the environment, or in the correctness proofs. Agreement problems such as the consensus problem [27], [22] are typical examples of fundamental problems whose specifications and solutions rely on such a preliminary agreement. The fact that all processes have to observe and share a common view of the whole system does not imply that the system is static. The group concept [35] allows processes to join or leave a group: a membership service [44] ensures that all the processes share a *consistent* view of the group composition and allows to synchronize the activities of the processes with regard to the successive evolutions of the group composition (view installation, view synchrony, ...). Obviously, the size of the system and the evolution speed of the group composition are two main factors that inherently may limit the use of this approach, especially in large scale dynamic networks. However, many distributed applications involve only a limited number of cooperating processes (*e.g.* small systems, sets of replicas) or can be structured into several sub-systems (*e.g.* hierarchies, clusters, communities of interests). Therefore, designing distributed applications based on the group concept remains a very attractive approach. In what follows, the description of this general activity is subdivided into three sections devoted respectively to (1) the study of group communication in the presence of benign faults, (2) the detection of arbitrary behaviors due to external attacks, and (3) the management of Grids.

3.3.2. Group Communication Services

Providing group communication services within a system is essential [35]. Several important services, such as total order broadcast, group membership, or non-blocking atomic commitment are very useful to observe and synchronize activities within a group. For example, a reliable broadcast primitive guarantees that any message sent to a set of entities is delivered either by all of them or by none of them. An atomic broadcast primitive requires that the order in which messages are delivered to the recipients is the same for all of them. These two particular communication primitives facilitate the task of an application designer since they guarantee strong properties regarding the delivery of the messages to the recipients and the order in which these messages are delivered [28]. For instance, to increase the overall reliability of the system, critical data and functionalities have to be replicated on a group of nodes. Ensuring consistency within such a set of copies becomes trivial if an atomic broadcast service is available.

Many group services can be classified as agreement problems (membership, total order broadcast, consensus, leader election, atomic validation, ...). As these services are used very often, efficiency is a key issue when designing solutions to such agreement problems. Our first goal is to have an even better understanding of these problems while considering various levels of adversity (various computational models ranging from the purely synchronous one to the purely asynchronous one, various failure models, ...). We aim to design an homogeneous library of reliable, flexible, modular, and adaptive group services. To reach this goal, we propose to build all these services on top of a generic and adaptive solution to the consensus problem, that can be customized to cope with the characteristics of the environment, as well as the properties of the reliable distributed abstractions that have to be ensured. From an algorithmic point of view, several design choices (definition of tunable consensus protocol parameters, consensus algorithms with multiple round participations, continual execution of consensus instances, use of clock synchronization algorithms to fix the round duration, ...) lead to obtain an original software whose performance differs from that obtained by other group communication projects (*Ensemble* - Cornell University and the Hebrew University the Ensemble project, *Appia* - University of Lisbon [42], *Samoa* - EPFL [43], ...). From a software engineering point of view, the use of a componentware approach helps to implement the group concept in a modular way. Code tangling is a major concern when designing group communication services. Even if a consensus-based solution allows for a clean separation between agreement related code and protocol speci?c code, many concerns that crosscuts the various protocol codes remain. Thus we have to identify all the hidden synchronisation constraints between the modules used to implement group services and we aim to propose a componentization that takes into account these constraints. Conducting a performance evaluation of our proposal is also a part of our future activities.

3.3.3. Security of a Web Access in the event of New Attacks

Intentional faults are produced by malicious attackers who try to take advantage of residual vulnerabilities that always exist in a complex system. When considering intentional faults, our aim is not to propose preventive measures (access controls, encryption, firewalls,...). Assuming that an intrusion can succeed, we want to be able to detect it, to confine damage and to clean and recover corrupted entities from errors.

Intrusion detection is traditionally based on *explicit methods* in which a reference model is built and used to detect attacks. In the case of a Web server that delivers dynamic contents, we take a radically different approach called *implicit intrusion detection*. We show that the use of diversified COTS servers allows to detect intrusions. To secure a web access to a set of data, we assume that data are replicated and accessible through different systems that may have residual vulnerabilities but hopefully not necessarily the same ones. Consequently, an attack can succeed on a particular copy but not on all the copies. By checking the values returned by the different copies to the malicious attacker we can identify differences and detect anomalies. Of course, the difficult part is to provide replication and detection mechanisms that are safe and will not become an even more simple target for the attacker. Our aim is to study how group services can be used and adapted to achieve this objective. This approach can detect even previously unknown attacks. Similar studies (leading however to different algorithmic solutions) have been conducted by the LAAS (Delta-4 [34], [25]and DIT architectures [38]) and by the university of Texas at Austin [45].

3.3.4. Task Allocation in a Grid in the event of Benign Faults

A complete transparency and a quick response time are always expected by Grid users even when failures occur. To fulfill both requirements, adaptive control mechanisms have to be proposed, first to cope efficiently with the dynamic changes of the computing capacity of the Grid (even if these changes are unpredictable) and second, to distribute the tasks among the resources in an efficient way (dynamic load balancing).

In a Grid that federates resources provided by different institutions, we address two major issues that both require a continuous adaptation to the changing computing environment, namely the *Resource Allocation* issue and the *Dependability* issue. We propose to solve both problems in an homogeneous way, using a slightly modified group concept. Our goal is not to design a complete range of services for Grid systems. We focus only on two specific issues: resource allocation and dependability. Moreover we restrict the scope of our research to time-consuming applications that can be decomposed into a huge number of independent tasks. As all the tasks generated during the execution of such an application are independent, they can be allocated independently on the different resources of a Grid. The above assumptions are not unrealistic and are usually satisfied by genomic applications (See the description of the software Paradis).

With respect to these research topics (resource allocation and dependability), our contributions aim to promote two major complementary ideas. First, we suggest that the architecture of a Grid follows a hierarchical structure. Second we claim that interactions within the grid can be reduced to either a classical master-slave scheme or to a sequence of unanimous decisions depending on the level in the hierarchy of the interacting entities. This strategy allows us to benefit from the existence of synchronous networks where upper temporal bounds (on message transfer delays and also on the time required to execute a computation step) exist and can be known. On the contrary, more complex agreement protocols are used to share a consistent view of the global state of the Grid between unreliable entities linked through an asynchronous networks.

3.4. Reputation and Self-organization in Large Scale Systems

Keywords: *grid computing*, *reputation*, *self-stabilization*.

3.4.1. Reputation Mechanisms Robust against Attacks and Threats

There is an increasing number of highly distributed applications that rely on reputation mechanisms. These mechanisms encourage relationship among trustworthy entities, while they discourage them in presence of unstrustworthy entities. For instance, eBay, one of the largest online-auction sysem, relies on a rating system to find traders and allows partners to rate each other after completion of an auction. Similarly, Amazon, Slashdot, ePinions, Yahoo! auctions, just to cite a few of them, rely on a reputation mechanism to foster a trust relationship among entities that do not know each other *a priori*, and may interact once and only once.

Specifically, similarly to real world reputation, a reputation mechanism expresses a collective opinion about some target entity by collecting and aggregating feedback about the past behavior of that target entity. The derived reputation score is used to help entities to decide whether an interaction with that entity is conceivable or not. By encouraging trust or distrust, reputation helps in finding new resources by using trusted entities as sources of the search. It is also a powerful tool to incentive correct behavior. A well behaving entity maintains its good reputation score so that entities are interested in interacting with it. On the other hand, reputation can be used as a punishment tool. By lowering reputation score of misbehaving entities, establishment of relationships with other entities is made harder.

In the context of open and large scale systems, such as P2P and ad-hocs systems, we envision to use reputation as a building block for the deployment of security policies. Those policies will be dynamically chosen according to the level of hostility perceived by each entity. However, to be considered as a valuable tool for trust assessment, a reputation mechanism has to be itself robust against adversity. In other words, reputation must have the ability to self-heal or at least to self-protect against undesirable behavior not to jeopardize users security. Attacks in open systems are numerous and can be magnified through collusion. Just to name a few, reputation mechanism have to be able to face

- whitewashing (badly scored entites leave and rejoin the system to renew their reputation score),
- masquerading (badly scored entities pretend to be another entity to acquire its good reputation score),
- bad mouthing (collusion to discredit the reputation of a service provider to lately benefit from it),
- ballot stuffing (collusion to advertise the quality of service of a service provider more than its real value to increase its reputation to push users to be involved in fraudulent transactions),
- sybil attack (generation of numerous fake entities to manipulate the reputation score),
- transaction repudation (an entity can deny the existence of a transaction)
- ...

Increasing the robustness of reputation mechanisms encompasses robustness both at the reputation mechanism itself as previously described, but also at the underlying network level. Specifically appropriate mechanisms should prevent message corruption, rerouting, and denial of service during the feedback collect phase.

In this context, we envision to contribute at the different phases of the reputation mechanism construction. Regarding feedback aggregation, we propose to extend existing works (e.g., [16], [24]) by enlarging the behavioral assumptions of interacting entities (e.g., variation of the effort exerted by providing entities according to the entities with which they interact, according to their welfare, or their level of hostility), by minimizing the number of relevant feedback needed to build a fair enough score estimation so that reputation could quickly react to highly dynamic environments. An interesting approach would be to combine credibility-based reputation function with endogenous techniques, well adapted for massive churn [17]. Regarding feedback availability, a classical solution amounts in replicating feedback at different entities hence guaranting that despite disconnections and malicious behavior, feedback information remain available within the system. However this type of solution relies on entities propensity to fully and honestly cooperate. Such assumptions are ideal ones, and cannot be enforced without relying on incentive mechanisms. Finally, it has been shown that peer-to-peer overlay networks can only survive severe (Byzantine) attacks if malicious peers are not able to predict what is going to be the topology of the network for a given sequence of join and leave operations. Induced churn, by which peers are required to rejoin (leave and, immediately after, join again) the system seems to be an appealing solution for the construction of Byzantine-resilient overlays.

3.4.2. Hierarchical Specification of Self-organization in Dynamic Systems

Self-organization is an evolutionary process that appears in many disciplines. Physics, biology, chemistry, mathematics, economics, just to cite a few of them, show many examples of self-organizing systems. Crystallisation, percolation, chemical reactions, proteins folding, flocking, cellular automata, market economy are among the well-admitted self-organizing systems. In all these disciplines, self-organization is described as a process from which properties emerge at a global level of the system. These properties are solely due to local interactions among components of the system, that is with no explicit control from outside the system). Influence of the environment is present but not intrusive, in the sense that it does not disturb the internal organization process. In the newly emerging fields of distributed systems (p2p, ad-hoc networks, sensor networks, cooperative robotics), self-organization becomes one of the most desired properties. The major feature of all recent scalable distributed systems is their extreme dynamism in terms of structure, content, and load. In p2p networks, nodes continuously join and leave the system. In large scale sensor, ad-hoc or robot networks, the energy fluctuation of batteries and the inherent mobility of nodes induce a dynamic aspect of the system. In all these systems there is no central entity in charge of their organization and control, and there is an equal capability, and responsibility entrusted to each of them to own data [32]. To cope with such characteristics, these systems must be able to spontaneously organize toward desirable global properties. In peer-to-peer systems, self-organization is handled through protocols for node arrival and departure, as provided by distributed hash tables based-overlay (e.g., CAN, Chord, Pastry [26], [36], [41], [37]) , or random graph-based ones (e.g. Gnutella, GIA [23]).

4. Application Domains

4.1. Space Domain Applications

Keywords: *distributed computing*, *failure model*, *fault tolerance*.

To cope with more and more complex requirements, this sector of activity shows a growing interest in distributed computing. More precisely, the adequacy between the properties ensured by their applications (that are getting increasingly stronger) and the assumptions about their systems (that are getting inexorably weaker) becomes questionable. In particular, regarding **fault tolerance**, a large number of entities (software and hardware entities) of the embedded computer-based system interact with each other. To make interaction robust, a broad range of failures (from benign failures up to malicious failures) have to be tolerated. Regarding **flexibility and adaptability**, the new generation of distributed services has to be adaptive. To achieve this goal, algorithmic solutions have to benefit from the recent advances in software engineering (componentware approach) and a provable methodology to specify, design and prove the distributed algorithms is needed.

4.2. Telecommunication Applications

Keywords: *P2P*, *distributed computing*, *reputation*.

Telecommunications domain is very interested in peer-to-peer computing. "Nowadays, people are not just satisfied with "can hear a person from another side of the earth", instead, the demands of clearer voice in real-time are increasing globally. Just like the TV network, there are already cables in place, and it's not very likely for companies to change all the cables. Many of them turn to use the internet, more specifically P2P networks. For instance, Skype, one of the most widely used internet phone applications is using P2P (peer-to-peer) technology" [excerpt from Wikipedia]. By relying on P2P paradigm, telecommunication industry is enlarging its panel of innovating applications ranging from video on demand to massively-shared and user-generated unbounded digital universe. A prerequisite for these applications to meet quality of service requirements of their users is the effective and honest participation of these very same users. In absence of any large centralized enforcement institution in charge of controlling users behavior, the only viable alternative for encouraging trustworthy behavior is to rely on informal social mechanisms collecting, and aggregating information about user behaviors, a.k.a., reputation mechanisms.

5. Software

5.1. PROMETEUS: a Group Communication Service

Keywords: *fault tolerance*, *group communication*, *middleware*.

Participants: Michel Hurfin, Jean-Pierre Le Narzul, Romaric Ludinard, Frédéric Tronel.

The PROMETEUS project, part of the Inria Gforge, is a software environment for reliable programming developed by the Adept team. The basic elements of PROMETEUS are Eva, a component-based framework and, on top of Eva, Adam that is a library of agreement components for use by applications.

PARADIS is a middleware for a Grid dedicated to genomic applications. Paradis makes use of the ADAM library to reliably allocate resources to tasks to be executed on the Grid.

In the context of the DADDI project, we have also developed a software system that enhance the integrity and availability of an Intrusion Detection Architecture targeted to a Web Server that delivers dynamic contents.

- Eva

  EVA is an implementation of a component model that aims at supporting the development of distributed abstractions and high-level communication protocols. EVA implements a publish/subscribe communication environment to structure components composing high level protocols. In the EVA model, protocols are regarded as a number of cooperating components that communicate via an event channel. Communication is achieved via the production of events (output data) by supplier components, and the consumption of these events (input data) by consumer components. A supplier component uses the service of an event channel to route the events it produces to any consumer component that has registered with the event channel its interest in consuming that particular type of event. The event channel decouples suppliers from consumers yielding an interesting flexibility. Synchronous interactions between components is also supported in EVA. Special attention has been devoted to optimize the implementation. For example, potential sources of overheads (in the management or transmission of events) have been limited or eliminated in the design and implementation of EVA.

- Adam

ADAM is a library of agreement components, based on the component model implemented by eva. The central element of the ADAM library is gac (generic agreement component). It implements a generic and adaptive fault-tolerant consensus algorithm that can be customized to cope with the characteristics of the environment. Moreover, thanks to a set of versatile methods, its behavior can be tuned to fit the exact needs of a specific agreement problem. A range of fundamental ADAM components are implemented as specializations of this gac component. The ADAM library currently includes the most important components for reliable distributed programming: Group Membership, Atomic Broadcast, Leader Election, etc.

- Paradis

  PARADIS is a middleware for a Grid dedicated to genomic applications. Genomic applications are time-consuming applications that can usually be split into a huge number of independent tasks. Paradis is used to reliably allocate resources to these tasks.

  In PARADIS, we consider that the network which is globally asynchronous, is composed of synchronous subnetworks called domains (in practice, these domains correspond to LANs). To improve the fault tolerance and the efficiency of computations on the Grid, we try to benefit as much as possible from the synchronous properties of communications within a domain and to avoid as much as we can the communications within domains. To avoid a flood of the Grid, only one node per domain is allowed to communicate with the other domains. This node is called the proxy. In order to provide an easy access to the Grid from anywhere, the applications can be launched through web portals.

  The implementation of PARADIS relies on the ADAM library. Agreement components are used by proxies to share a common view of the evolution of the Grid. Decisions are used to solve, despite failures, the group membership problem and the resource allocation problem.

- Extensions to cope with Security Aspects

  In the context of the DADDi project, we develop a software system that enhance the integrity and availability of an Intrusion Detection Architecture targeted to a Web Server that delivers dynamic contents. Like PARADIS, this system relies on the ADAM library and more specifically on the leader election, membership management and atomic broadcast components. Thanks to these components, our system provides to the intrusion detection architecture, the two following properties: (1) availability through the replication of the IDS (2) integrity of data through the replication of the SQL backend.

  Our system has been deployed on a intrusion detection platform that is based on a set of diversified Web servers running on top of three different operating systems (Windows, Linux, Mac OS X).

6. New Results

6.1. Agreement in Distributed Systems

Keywords: *geometric model*, *grid computing*, *group communication*.

Participants: Michel Hurfin, Jean-Pierre Le Narzul, Izabela Moise, Frédéric Tronel.

6.1.1. Towards automatic proofs of Impossibility Results in Wait-Free Systems

Our studies are intended to analyze the mathematical modeling of wait-free systems. In this computation model, processes synchronize and interact by the mean of a totally asynchronous distributed memory, and all of them but one can fail (we only consider fail-stop failures). This computational model is called *wait-free*, because no process should ever wait for a message coming from any other process, because it may have crashed. Waiting for a process to make some progress in a computation is clearly a potential deadlock. There has been an intensive research work in this domain, and many deep results have been obtained. For example, Herlihy and Shavit have defined a precise characterization of the distrdynamicallyibuted tasks that can be solved in the wait-free computation model [31]. A restriction of the Herlihy-Shavit result relies in the fact that it mixes tools borrowed from the algebraic topology domain with more classical tools such as relations. In particular, the specification of a distributed problem is given as a relation between inputs and outputs. Unfortunately, these two worlds do not cooperate smoothly. More recently, Havlicek has reconsidered the theorem of Herlihy and Shavit to express the whole result in purely algebraic terms, by proposing a way to express the specification of a distributed problem in terms of geometrical objects [30], [29]. Using this new way to state the Herlihy-Shavit result, Havlicek has proposed a set of necessary conditions for a distributed problem to have a solution in a wait-free systems.

Our research activity focuses on two points. On the one hand, we propose to determine computable characterizations of tasks that make them unsolvable in the wait-free model. On the other hand, for tasks that are solvable, we have design and develop an algorithm that, given a formal specification of such a task, is able to synthesize a wait-free protocol that solves it. We have extended the result of Havlicek by showing that all the algebraic objects involved in the Havlicek theorem can be automatically computed (chromatic complexes, homology group, etc). We have been able to show that it is possible for a computer program to derive impossibility results when the necessary conditions exhibited by Havlicek are not satisfied by a given problem. Using this approach, we have been able to reestablished well-known impossibility results of the domain.

6.1.2. Group Communication Services to Secure a Web Access

Intrusion detection is one of the numerous techniques that help improving the overall security of a system. It is traditionally based on explicit methods that be can classified as follows :

- scenario-based techniques : one needs to explicitly provide scenarios of previously known attacks. All kinds of probes are placed in the system to protect and the flow of information returned by these probes is analyzed to find possible traces of attacks. Of course, this method is extremely sensible to the quality of scenarios that are provided. Furthermore, it cannot detect previously unknown attacks.
- behavioral techniques : this method requires to provide a reference model which precisely describes the legal states of a system. The system is observed by an external observer which raises alarms when the system deviates from the set of legal states. Of course, providing a good reference that do not raise to many false alarms, and that can detect real attacks is challenging.

In the context of the Daddi project, we take a radically different approach called implicit intrusion detection. We show that the use of diversified COTS servers allows to detect intrusions. This approach can detect even previously unknown attacks. Based on a architecture proposed by our partners in the DADDi project, we propose an extension that brings the availability and integrity properties to the system. Replication techniques implemented on top of agreement services are used to avoid any single point of failure.

We focus on a particular case study, namely a Web server that delivers dynamic content. This technology traditionally implements the storage of this content in a database backend that receives read/write operations issued by the Web server. An interesting property of this Web server architecture resides in the fact that the whole internal state of the COTS servers is located in the database backend and that any change to the internal state is carried out by the means of SQL queries. We take advantage of this property in order to ensure integrity of the data by introducing proxies located between the Web servers and the database whose goal is to compare the SQL queries submitted by the diverse Web servers to the database. As unexpected SQL queries issued by a corrupted Web server would threaten data integrity, we use a majority voting algorithm to compare queries submitted to the database; this comparison allows us to detect and mask any attempt to data integrity. Performance evaluations have been conducted and described in [15].

6.1.3. Group Communication Services to Allocate Tasks in a Grid

A Grid is a distributed system involving heterogeneous resources located in different geographical domains that are potentially managed by different organizations (companies, laboratories, universities, ...) or individuals. The major purpose of a Grid is to federate multiple powerful distributed resources (computers but also data storage facilities) within a single virtual entity which can be accessed transparently and efficiently by external users. In our work, we consider a Grid composed of resources provided by various institutions. These potential contributors are identified preliminarily and correspond to well-established institutions that agree to share their resources and to trust each other. Yet each institution keeps its independence and freedom. The decision to include or to exclude some (or even all) local resources from the Grid can be taken at any time by the local administrator without any coordination with the others.

In [12], we specifically address the resource allocation and dependability issues. We define services that allow a Grid user to continuously take full advantage of the computing power offered by the Grid in a simple and completely transparent manner. Whatever the circumstances, a complete transparency and a quick response time are always expected by the customers. To fulfill these two requirements, adaptive control mechanisms have to be proposed, on one hand to cope efficiently with the dynamic changes of the computing capacity of the Grid (even if these changes are unpredictable) and, on the other hand to distribute the tasks among the resources in an efficient way (dynamic load balancing). This leads us to address two major issues that both require a continuous adaptation to the changing computing environment, namely the *Resource allocation* issue and the *dependability* issue. We propose to solve both problems in an homogeneous way using a slightly modified group concept [35]. More precisely, all distant interactions between domains corresponding to distinct organizations are managed by a small group of registered processors (exactly one per domain). Each member of this group acts as a *master* for its own domain and interacts with the other members of the group to build consistent observations (1) of current workloads in each domain and (2) of the current composition of the group. In that sense, we argue that, in a distributed system prone to failures, an agreement service is a key concept to transform several local views into a single global one without opting for a centralized control approach and thus without having a single point of failure. An agreement service allows all the domains to acquire the same set of accurate data describing the current state of the Grid. Based on this unanimous observation, each domain can locally determine the right adaptation to react to the observed changes.

6.2. Reputation and Self-organization in Dynamic Large Scale Systems

Keywords: *P2P*, *reputation*, *self-organization*.

Participants: Emmanuelle Anceaume, Jean-Pierre Le Narzul, Aina Ravoaja, Heverson Ribeiro.

6.2.1. Reputation Mechanisms

In [10], we present PeerCube, a DHT-based system aiming at avoiding high churn from impacting the performance of the system and at the same time at preventing malicious behaviour (coordinated or not) from subverting the system. As many existing DHT-based overlays, PeerCube is based on a hypercubic topology. PeerCube peers self-organise into clusters whose interconnections form the hypercubic topology. Peers within each cluster are classified into two categories, core members and spares, such that only the former ones are actively involved in PeerCube operations. Thus only a fraction of churn affects the overall topology of the hypercube. Defences against eclipse attacks are based on the observation that malicious peers can more easily draw a successful adversarial strategy from a deterministic algorithm than from a randomised one. We show that regardless of the adversarial strategy colluders employ, the randomised insertion algorithm we propose guarantees that the expected number of colluders in each routing table is minimal. Furthermore, by keeping the number of core members per cluster small and constant, it allows to rely on the powerful consensus building block to guarantee consistency of the routing tables despite Byzantine peers. Finally, PeerCube takes advantage of independent and optimal length paths offered by the hypercubic topology to decrease exponentially the probability of encountering a faulty peer with the number of independent paths [40].

To summarise, PeerCube brings together research achievements in both "classical” distributed systems and open large scale systems (Byzantine consensus, clustering, distributed hash tables) so that it efficiently deals with collusion and churn. To the best of our knowledge this work is the first one capable of tolerating collusion by requiring for each `lookup`, `put`, `join` and `leave` operation $\mathcal{O}(logN)$ latency and only $\mathcal{O}(logN)$ messages.

6.2.2. Specification of Self-organization

The main focus of this work is to propose a formal specification of the self-organization notion which, for the best of our knowledge, has never been formalized in the area of scalable and dynamic systems, in spite of an overwhelming use of the term. Reducing the specification of self-organisation to the behavior of the system during its non-dynamic periods is clearly not satisfying essentially because these periods may be very short, and rare. On the other hand, defining self-organization as a simple convergence process towards a stable predefined set of admissible configurations is inadequate for the following two reasons. First, it may be impossible to clearly characterize the set of admissible configurations since, in dynamic systems, a configuration should include the state of some key parameters that have a strong influence on the dynamicity of the system. These parameters can seldom be quantified a priori (e.g., the status of batteries in sensor networks, or the data stored within p2p systems). Second, due to the dynamic behavior of nodes, it may happen that no execution of the system converges to one of the predefined admissible configurations.

Hence our attempt to specify self-organisation according to the very principles that govern dynamic systems, namely high interaction, dynamics , and heterogeneity. These tenets share as a common seed the locality principle, i.e., the fact that both interactions and knowledge are limited in range. The first contribution of this paper is a formalisation of this idea, leading first to the notion of *local self-organisation*. Intuitively, a locally self-organizing system should reduce locally the entropy of the system. For example, a locally self-organized p2p system forces components to be adjacent to components that improve, or at least maintain, some property or evaluation criterion.

The second goal of this work is the proposition of a hierarchical specification of self-organisation. The purpose of this hierarchy is to classify dynamic systems according to their capability to produce emerging properties with respect to the churn of the system. Basically, we propose three classes of self-organization. The first one characterizes dynamic systems that converge toward sought global properties only during stable periods of time (these properties can be lost due to instability). The second one depicts dynamic systems capable of infinitely often increasing the convergence towards global properties (despite some form of instability). Finally, the last one describes dynamic systems capable of continuously increasing that convergence. We show that complex emergent properties can be described as a combinaison of local and independent ones. Results of this work will soon be submitted for publication to the Journal of Parallel and Distributed Computing.

7. Other Grants and Activities

7.1. National Project

7.1.1. ACI Daddi (2004-2008)

Participants: Michel Hurfin, Jean-Pierre Le Narzul, Frédéric Tronel.

Each day, the databases maintaining information about system vulnerabilities are growing with new cases. In addition to the classical prevention security tools, intrusion detection systems (IDS) are nowadays widely used by security administrators to detect attack occurrences against their systems. Anomaly detection is often viewed as the only approach to detect new forms of attack. The main principle of this approach consists in building a reference model of the behavior for a given entity (user, machine, service, or application) in order to compare it with the current observed behavior. If the observed behavior diverges from the model, an alert is raised to report the anomaly. Rather than defining an explicit model, we suggest to consider an implicit one. Design diversity will be used to identify dynamically the reference model. In our approach, any request is forwarded to different modules implementing the same functionality but through diverse designs. Any difference between results that are returned can be interpreted as a possible corruption of one or several modules. The task of the ADEPT project is to provide secure group communication mechanisms that allow to managed the group of modules. In 2008, a first version of our set of communication services called PROMETEUS has been released. Performance evaluations have been conducted and described in [15].

7.1.2. RIAM Project (ANR): Solipsis (2006-2008)

Participants: Emmanuelle Anceaume, Aina Ravoaja, Heverson Ribeiro.

Solipsis is a R&D project (ANR-RIAM) leaded by Orange lab, and supported by ANR and Media and Networks cluster of Brittany. It was launched in january 2007 and is based on prior works stated in 1998. Five partners are involved: IRISA, Archivideo, Artefacto, University of Rennes 2 , and Orange Labs.

Solipsis is an open source system for a massively multi-participant shared 3D virtual world. Briefly, the challenge of this project is to offer the opportunity to the users to imply themselves in the collective creation of a public virtual space. Nowadays, virtual worlds are the property of companies or association which manage centralized servers. Even the most open systems are not freed from a central and responsible authority. By considering an open system based on a peer-to-peer architecture, Solipsis hopes for a rich ecosystem to develop. This architectural choice guarantees that the virtual world can be populated by an unlimited number of users, without cost of deployment.

Lack of scalability is a key issue for virtual-environment technology, and more generally for any large-scale online experience because it prevents the emergence of a truly massive virtual-world infrastructure (Metaverse). The Solipsis project tackles this issue through the use of peer-to-peer technology, and makes it possible to build and manage a world-scale Metaverse in a truly distributed manner. Following a peer-to-peer scheme, entities collaborate to build up a common set of virtual worlds. The Solipsis architecture as well as the communication protocol used to share data between peers is presented in [11]. The protocol is based on Raynet, an n-dimensional Voronoi-based overlay network. By not relying on dedicated servers that control consistency and legitimacy of peers behaviors and interactions, we have to face several challenges such as Eclipse attacks that aim at populating honest peers neighborhood by malicious peers, Sybil attacks whose goal is to out vote honest peers in collaborative tasks by the creation of an unbounded number of fake identities. Facing such attacks is worsened by the fact that peers interactions involves huge amount of data (3-D objects) thus making each single maintenance operation critical.

7.1.3. DGE Project: P2Pim@ges (2007-2009)

Participants: Emmanuelle Anceaume, Jean-Pierre Le Narzul, Romaric Ludinard.

The P2Pim@ge project is supported by the Direction Generale des Entreprises. This project aims at studying, prototyping and testing legal advanced streaming technology on peer-to-peer systems. Different applications will be addressed such as video on demand, immediate or differed download, access to scare content, etc. Partners of the project are Thomson R&D, Thomson Broadcast & Multimedia, Mitsubishi Electric ITE/TCL, Devoteam, France Telecom, ENST Bretagne, Marsouin, IRISA, IPdiva, and TMG.

In such large-scale dynamic systems, users may have a strategic behavior that is neither obedient nor malicious, but just rational. Tracking such behavior is complex since it requires taking into account a large set of features: large population, asymmetry of interest, collusion, "zero-cost identity", high turnover, and rationality. Techniques from the security domain (e.g. intrusion detection), and new fault tolerant distributed algorithms inspired from social theories will be investigated to deal with these undesirable behaviors.

7.2. International Cooperations

7.2.1. Brazil (Federal University of Bahia and Federal University of Campina Grande)

Participants: Emmanuelle Anceaume, Michel Hurfin, Jean-Pierre Le Narzul, Frederic Tronel.

A cooperation project with the Federal University of Bahia, the Federal University of Paraiba, and several French laboratories (EPI ADEPT, EPI GRAND LARGE, EPI REGAL and ENST Bretagne) is supported by Capes/Cofecub (projet 497/05) during a period of four years (2005-2008). Michel Hurfin is the French coordinator of this project which focuses on distributed computing and Grid computing. In July 2008, Fabiola Greve (Federal University of Bahia) and Francisco Brasileiro (Federal University of Campina Grande) have visited the French Laboratories. Pierre Sens (EPI REGAL) and Sébastien Tixeuil (EPI GRAND LARGE) have visited the Federal University of Bahia respectively in september and october 2008. In 2008, this cooperation has lead to two joint publications presented at SASO 2008 [10] and OPODIS 2008 [39].

8. Dissemination

8.1. Teaching Activities

- Some members of the ADEPT research team belong to the University of Rennes I or to Telecom Bretagne (a telecommunication engineering school). Therefore, an important part of their time is devoted to teaching to engineers and master students.
- Jean-Pierre Le Narzul has the responsibility for organizing several teaching units at Telecom Bretagne (RSM Department). He gives lectures on both distributed computing and object-oriented language. He is also involved in the setting of programs for continuous training.
- Till september 2008, Frédéric Tronel has the responsibility of managing the master in computer science devoted to security aspects (University of Rennes I, Ifsic).
- Emmanuelle Anceaume was a member of the specialist commission (University of Rennes I, section 27). She participates to the master research (modules BIB and META).
- Michel Hurfin gave lectures on fault tolerance and distributed computing to students of two engineering schools: Telecom Bretagne (Rennes, 6 hours) and Supelec (Rennes, 8 hours).

8.2. Presentations of Research Works

- Since january 2000, Emmanuelle Anceaume co-organizes with Bruno Tuffin the seminars entitled "Networks and Systems" that are held in our institute.

8.3. Integration within the Scientific Community

- Emmanuelle Anceaume :
  - served as a program committee member of the 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), March 2008, Okinawa, Japan.
  - served as a program committee member of the 2nd International Symposium on Service, Security and its Data management for Ubiquitous Computing (SSDU 2008), May 2008, Kunming, China.
  - served as a program committee member of the 4th International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2008), July 2008, Sorrento, Italy.
  - served as a program committee member of the 2nd International Symposium on Security and Multimodality in Pervavise Environments (SMPE 2008), July 2008, Dublin, Ireland.
  - served as a program committee member of the 10th International Conference on High Performance Computing and Communications (HPCC 2008), September 2008, Dalian, China.
  - served as a program committee member of the International Symposium on Trusted Computing (TrustCom 2008), November 2008, Hunan, China.
  - acted as a reviewer for the journal IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans.
  - acted as a reviewer during the selection process of the INRIA Associated Team projects.
- Michel Hurfin :
  - served as a program committee member of the "5th International Conference on Autonomic and Trusted Computing" (ATC-08), June 2008, Oslo, Norway.

- served as a program committee member of the "2008 Workshop on Advanced Computing for Critical Systems and Emergency Preparedness and Response" (WCEMP 2008), July 2008, São Paulo, Brazil.
- served as a lecture committee member of the "9th African Conference on Research in Computer Science and Applied Mathematics" (CARI'2008), October 2008, Rabat, Maroc.
- served as a program committee member of the "Latin America Grid workshop" (LAGrid 2008), October 2008, Campo Grande, Brazil.
- acts as a program committee member of the "conférence sur la sécurité des architectures réseaux et des systèmes d'information" (SARSSI 2009), June 2009, Luchon, France.
- acted as a reviewer for the Vienna Science and Technology Fund (WWTF), Vienna, Austria.
- acted as a reviewer for several conferences and journals.
- acts as the french coordinator of a cooperation project between two brazilian federal universities (UFBA et UFCG) and 4 french research teams (EPI ADEPT, EPI GRAND LARGE, EPI REGAL, Telecom Bretagne).

9. Bibliography

Major publications by the team in recent years

[1] E. ANCEAUME, A. DATTA, M. GRADINARIU, G. SIMON. *Publish/Subscribe Scheme for Mobile Networks*, in "Proc. of the 2nd ACM International Workshop on Principles of Mobile Computing (POMC), Toulouse, France", October 2002, p. 74–81.

[2] E. ANCEAUME, M. HURFIN, P. RAÏPIN PARVÉDY. *An Efficient Solution to the k-set Agreement Problem*, in "Proc. of the 4th European Dependable Computing Conference (EDCC), Toulouse, France", LNCS 2485, Springer Verlag, October 2002, p. 62–78.

[3] E. ANCEAUME, C. DELPORTE-GALLET, H. FAUCONNIER, M. HURFIN, G. LE LANN. *Designing Modular Services in the Scattered Byzantine Failure Model*, in "Proc. of the 3rd International Symposium on Parallel and Distributed Computing (ISPDC), Cork, Ireland", July 2004, p. 262–269.

[4] F. BRASILEIRO, F. GREVE, M. HURFIN, J.-P. LE NARZUL, F. TRONEL. *Eva: an Event-Based Framework for Developing Specialised Communication Protocols*, in "Proc. of the 1st IEEE International Symposium on Network Computing and Applications (NCA), Cambridge, MA", February 2002.

[5] J.-M. HELARY, M. HURFIN, A. MOSTÉFAOUI, M. RAYNAL, F. TRONEL. *Computing Global Functions in Asynchronous Distributed Systems with Process Crashes*, in "Proc. of the 20th International Conference on Distributed Computing Systems (ICDCS)", Best paper award, April 2000, p. 584–591.

[6] M. HURFIN, A. MOSTÉFAOUI, M. RAYNAL. *A Versatile Family of Consensus Protocols Based on Chandra-Toueg's Unreliable Failure Detectors*, in "IEEE Transactions on Computers", vol. 51, n^o 4, April 2002, p. 395–408.

[7] M. HURFIN, M. RAYNAL. *A simple and Fast Asynchronous Consensus Protocol Based on a Weak Failure Detector*, in "Distributed Computing", vol. 4, n^o 12, 1999, p. 209–223.

[8] Y. Wang, E. Anceaume, F. Brasileiro, F. Greve, M. Hurfin. *Solving the Group Priority Inversion Problem in a Timed Asynchronous System*, in "IEEE Transactions on Computers. Special Issue on Asynchronous Real-Time Disttributed Systems", vol. 51, n^{o} 8, August 2002, p. 900–915.

Year Publications

Doctoral Dissertations and Habilitation Theses

[9] A. Ravoaja. *Mécanismes et architectures P2P robustes et incitatifs pour la réputation*, Thèse de doctorat, Université de Rennes I (école doctorale Matisse), Dec 2008.

International Peer-Reviewed Conference/Proceedings

[10] E. Anceaume, R. Ludinard, A. Ravoaja, F. Brasileiro. *PeerCube: An Hypercube-based P2P overlay robust against collusion and churn*, in "Proc. of the 2nd International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008), Venice, Italy", IEEE, Oct 2008, p. 15–24, http://hal.inria.fr/inria-00258933/fr/.

[11] D. Frey, J. Royan, R. Piegay, A.-M. Kermarrec, E. Anceaume, F. Le Fessant. *Solipsis: A Decentralized Architecture for Virtual Environments*, in "Proc. of the 1st International Workshop on Massively Multiuser Virtual Environments (MMVE 2008), Reno, Nevada", IEEE, Mar 2008, p. 29–33, http://hal.inria.fr/inria-00337057/fr/.

[12] J.-P. Le Narzul, M. Hurfin. *Design and Performance Evaluation of a Resource Allocation System Based on Agreement Services*, in "Proc. of the International Workshop on Grid Computing Applications Development (GridCAD / SYNASC 2008), Timisoara, Roumania", IEEE, Sept 2008.

[13] D. Moise, I. Moise, F. Pop, V. Cristea. *Resource CoAllocation for Scheduling Tasks with Dependencies, in Grid*, in "Proc of the 2nd International Workshop on High Performance Grid Middleware (HiPerGRID 2008), Bucharest, Romania", IEEE, Nov 2008.

[14] I. Moise, D. Moise, F. Pop, V. Cristea. *Advance Reservation of Resources for Task Execution in Grid Environments*, in "Proc of the 2nd International Workshop on High Performance Grid Middleware (HiPerGRID 2008), Bucharest, Romania", IEEE, Nov 2008.

Other Publications

[15] M. Hurfin, R. Ludinard, F. Tronel, J.-P. Le Narzul. *Services de communication de groupe - Evaluations à partir de la trace commune*, Projet ANR Daddi - Annexe aux délivrables 4.1 et 4.2, Jul 2008.

References in notes

[16] E. Anceaume, A. Ravoaja. *Incentive-based Robust Reputation Mechanism for P2P Services*, in "Proc. of the 10th International Conference On Principles Of Distributed Systems (OPODIS 2006), Bordeaux, France", December 2006.

[17] B. Awerbuch, B. Patt-Shamir, D. Peleg, Mark R. Tuttle. *Collaboration of untrusting peers with changing interests*, in "ACM Conference on Electronic Commerce", 2004.

[18] O. BIRAN, S. MORAN, S. ZAKS. *A combinatorial Characterization of the Distributed Tasks which are Solvable in the Presence of One Faulty Processor*, in "Proc. of the 7th Principles of Distributed Computing", august 1998.

[19] E. BOROWSKY, E. GAFNI. *Generalized FLP Impossibility Results for t-Resilient Asynchronous Computations*, in "Proc. 25th ACM Symposium on Theory of Computation, California (USA)", 1993, p. 91-100.

[20] E. BOROWSKY, E. GAFNI. *A Simple Algorithmically Reasoned Characterization of Wait-Free Computations (Extended Abstract)*, in "Symposium on Principles of Distributed Computing", 1997, p. 189-198.

[21] TUSHAR DEEPAK. CHANDRA, V. HADZILACOS, S. TOUEG. *The Weakest Failure Detector for Solving Consensus*, in "Proceedings of the 11th Annual ACM Symposium on Principles of Distributed Computing (PODC'92), Vancouver, BC, Canada", M. HERLIHY (editor), ACM Press, 1992, p. 147–158, http://citeseer.ist.psu.edu/chandra96weakest.html.

[22] TUSHAR DEEPAK. CHANDRA, S. TOUEG. *Unreliable failure detectors for reliable distributed systems*, in "Journal of the ACM", vol. 43, n^{o} 2, 1996, p. 225–267, http://citeseer.ist.psu.edu/chandra96unreliable.html.

[23] Y. CHAWATHE, S. RATNASAMY, L. BRESLAU, N. LANHAM, S. SHENKER. *Making Gnutella-like P2P systems scalable*, in "Proc. of the annual conference of the Special Interest Group on Data Communication (SIGCOMM)", ACM, 2003.

[24] Z. DESPOTOVIC, K. ABERER. *P2P reputation management: Probabilistic estimation vs social networks*, in "Computer Networks", vol. 50, n^{o} 4, 2006, p. 485–500.

[25] Y. DESWARTE, L. BLAIN, JEAN-CHARLES. FABRE. *Intrusion Tolerance in Distributed Computing Systems*, in "Proceedings of the IEEE Symposium on Research in Security and Privacy", May 1991, p. 110-122, ftp://ftp.laas.fr/pub/Publications/1990/90373.ps.

[26] P. DRUSCHEL, A. ROWSTRON. *PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility*, in "HotOS VIII", May 2001.

[27] M. J. FISCHER, N. A. LYNCH, M. S. PATERSON. *Impossibility of distributed consensus with one faulty process*, in "J. ACM", vol. 32, n^{o} 2, 1985, p. 374–382.

[28] V. HADZILACOS, S. TOUEG. *Distributed Systems*, in "Fault-tolerant broadcasts and related problems", S. MULLENDER (editor), ACM Press, 1996.

[29] J. HAVLICEK. *Computable obstructions to wait-free computability*, in "Distrib. Comput.", vol. 13, n^{o} 2, 2000, p. 59–83.

[30] J. HAVLICEK. *Computable Obstructions to Wait-free Computability*, in "IEEE Symposium on Foundations of Computer Science", 1997, p. 80-89.

[31] M. HERLIHY, N. SHAVIT. *The topological structure of asynchronous computability*, in "Journal of the ACM", vol. 46, n^{o} 6, 1999, p. 858–923.

[32] G. KAN. *Harnessing the benefits of a disruptive technology*, O'Reilley & Associates, March 2001.

[33] N. A. LYNCH. *Some Perspective on PODC*, in "Distributed Computing", vol. 16, 2003, p. 71–74.

[34] D. POWELL, G. BONN, D. SEATON, P. VERISSIMO, F. WAESELYNCK. *The Delta-4 Approach to Dependability in Open Distributed Computing Systems*, in "Proceedings of Twenty-Fifth International Symposium on Fault-Tolerant Computing", IEEE, 27-30 june 1995, 56.

[35] D. POWELL. *Group Communication*, in "Communications of the ACM", vol. 39, n^{o} 4, 1996, p. 50–53.

[36] S. RATNASAMY. *A Scalable Content-Addressable Network*, Ph. D. Thesis, University of California at Berkeley, 2002.

[37] A. ROWSTRON, P. DRUSCHEL. *Storage Management and Caching in PAST, a Large-Scale, Persistent Peer-to-Peer Storage Utility*, in "Proc. of the 17th ACM Symposium on Operating Systems Principles (SOSP)", 2001, p. 188–201.

[38] A. SAIDANE, Y. DESWARTE, V. NICOMETTE. *An Intrusion Tolerant Architecture for Dynamic Content Internet Servers*, in "Proceedings of the 2003 ACM Workshop on Survivable and Self-Regenerative Systems (SSRS-03), Fairfax, VA", P. LIU, P. PAL (editors), ACM Press, October 2003, p. 110-114.

[39] P. SENS, L. ARANTES, M. BOUILLAGUET, V. SIMON, F. GREVE. *An Unreliable Failure Detector for Unknown and Mobile Networks*, in "In Proc of 12th International Conference On Principles Of DIstributed Systems (OPODIS 2008), Luxor, Egypt", December 2008.

[40] M. SRIVATSA, L. LIU. *Vulnerabilities and Security Threats in Structured Peer-to-Peer Systems: A quantitiative Analysis*, in "Proc. of the 20th Annual Computer Security Applications Conference (ACSAC)", 2004.

[41] I. STOICA, R. MORRIS, D. KARGER, M. FRANS. KAASHOEK, H. BALAKRISHNAN. *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*, in "In Proc. SIGCOMM", ACM, 2001.

[42] THE APPIA PROJECT. *APPIA Communication Framework*, http://appia.di.fc.ul.pt.

[43] THE SAMOA PROJECT. *Samoa Project*, http://lsrwww.epfl.ch/page13488.html.

[44] R. VITENBERG, I. KEIDAR, GREGORY V. CHOCKLER, D. DOLEV. *Group Communication Specifications: A Comprehensive Study.*, in "ACM Computing Surveys", vol. 33, n^{o} 4, September 2001.

[45] J. YIN, JEAN-PHILIPPE. MARTIN, A. VENKATARAMANI, L. ALVISI, M. DAHLIN. *Separating Agreement from Execution for Byzantine Fault Tolerant Services*, in "Proceedings of the 19th ACM Symp. on Operating Systems Principles (SOSP-2003)", 2003.