INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team sardes

System architecture for reflective distributed computing environments

Grenoble - Rhône-Alpes

Theme : Distributed Systems and Services

Table of contents

1. Team

Research Scientists

Jean-Bernard Stefani [Team Leader, Senior Researcher, Ingénieur Général des Mines]
Eric Rutten [INRIA Junior Researcher, HdR]
Alan Schmitt [INRIA Junior Researcher]
Damien Pous [CNRS Junior Researcher]
Vivien Quéma [CNRS Junior Researcher, HdR]

Faculty Members

Sara Bouchenak [Associate Professor, Université Joseph Fourier, HdR]
Fabienne Boyer [Associate Professor, Université Joseph Fourier, HdR]
Olivier Gruber [Professor, Université Joseph Fourier, HdR]
Noël De Palma [Professor, Université Joseph Fourier, HdR]
Renaud Lachaize [Associate Professor, Université Joseph Fourier]

Technical Staff

Clément Deschamps
Fabien Mottet

PhD Students

Michael Lienhardt [government grant, until Jun. 2010]
Jean Arnaud [government grant, until Oct. 2010]
Soufyane Aboubekr [Inria grant, until Sep. 2010]
Stéphane Fontaine [government grant, until Sep. 2010]
Fabien Gaud [government grant, from Oct. 2007]
Willy Malvaud [Inria grant (contract), from Oct. 2007]
Loris Bouzonnet [industrial grant, from Jan. 2008]
Claudio Mezzina [Inria grant (contract), from Oct. 2008]
Thomas Braibant [government grant, from Oct. 2008]
Ludovic Demontes [government grant, from Oct. 2008]
Sylvain Genevès [government grant, from Oct. 2008]
Alessio Pace [Inria grant, from Jan. 2009]
Quentin Sabah [industrial grant, from Apr. 2010]
Baptiste Lepers [Inria grant (contract), from Oct. 2010]
Xin An [Inria grant (contract), from Oct. 2010]
Xavier Etchevers [Orange Labs, from Oct. 2010]

Post-Doctoral Fellows

Benoît Claudel [Inria grant (contract), until Sep. 2010]
Gwenaël Delaval [Inria grant (contract), until Sep. 2010]
Cinzia Di Giusto [Digiteo grant, from Jan. 2010]
Nicolas Palix [university grant, from Oct. 2010]
Dàmian Serrano [Inria grant (contract), from Nov. 2010]

Administrative Assistant

Diane Courtiol

2. Overall Objectives

2.1. Overall objectives

The overall goal of the SARDES project-team is to develop software engineering and software infrastructure (operating system, virtual machine, middleware) foundations for the construction of provably dependable, self-manageable distributed systems.

To contribute to the above goal, the project-team has three major objectives:

1. To develop component-based software technology, that allows the construction of efficient, dynamically configurable systems, and that relies on a well-defined formal foundation.
2. To develop a "language-based" approach to the construction of configurable, provably dependable operating systems and distributed software infrastructures.
3. To develop algorithms and control techniques required to build scalable, self-manageable distributed systems.

In line with these objectives, the project-team organizes its research along four major areas:

- **Languages and foundations for component systems** Work in this area focuses on language support and semantical foundations for distributed component-based systems, with two main goals: (1) the development of a new generation of reflective software component technology with a formal semantical basis, and extensive language support in the form of architecture description and programming languages for dynamic distributed software architectures; (2) the study of process calculus foundations and coinductive proof techniques for distributed component-based programs.
- **System support for multiscale systems** Work in this area focuses on operating system and middleware services required for the construction of component-based systems at different scales (multicore systems on chip, and peer-to-peer systems), with two main goals: (1) to develop algorithms and operating system functions required for the support of efficient event-based concurrency and component reconfiguration in MPSoCs; (2) to develop algorithms and middleware functions required for the deployment, configuration and operation of applications in realistic peer-to-peer environments, typically exploiting an epidemic approach.
- **Control for adaptive and self-managed systems** Workk in this area focuses on the exploitation and development of discrete and continuous control techniques for the construction of adaptive component-based system. Application domains considered for this theme are, respectively, embedded systems and performance management for application server clusters.
- **Virtual machine for component systems** Work in this area focuses on the development of a component-based virtual machine for embedded systems, with two main goals: (1) to develop an extended instruction set for component support, including support for dynamic configuration, orthogonal component persistence, and isolation; (2) to develop a native implementation of the virtual machine, on resource-constrained hardware.

2.2. Highlights

The papers [31] and [35] have been awarded the Best Paper Award at the 13th International Symposium on Component-Based Software Engineering (CBSE 2010) and the 5th European Conference on Computer systems (Eurosys 2010) conferences, respectively.

3. Scientific Foundations

3.1. Components and semantics

The primary foundations of the software component technology developed by SARDES relate to the component-based software engineering [105], and software architecture [103] fields. Nowadays, it is generally recognized that component-based software engineering and software architecture approaches are crucial to the development, deployment, management and maintenance of large, dependable software systems [52]. Several component models and associated architecture description languages have been devised over the past fifteen years: see e.g. [83] for an analysis of recent component models, and [87], [59] for surveys of architecture description languages.

To natively support configurability and adaptability in systems, SARDES component technology also draws from ideas in reflective languages [78], and reflective middleware [81], [57], [64]. Reflection can be used both to increase the separation of concerns in a system architecture, as pioneered by aspect-oriented programming [79], and to provide systematic means for modifying a system implementation.

The semantical foundations of component-based and reflective systems are not yet firmly established, however. Despite much work on formal foundations for component-based systems [84], [47], several questions remain open. For instance, notions of program equivalence when dealing with dynamically configurable capabilities, are far from being understood. To study the formal foundations of component-based technology, we try to model relevant constructs and capabilities in a process calculus, that is simple enough to formally analyze and reason about. This approach has been used successfully for the analysis of concurrency with the π-calculus [90], or the analysis of object-orientation [48]. Relevant developments for SARDES endeavours include behavioral theory and coinductive proof techniques [99], [97], process calculi with localities [60], [62], [65], and higher-order variants of the π-calculus [98], [72].

3.2. Open programming

Part of the language developments in SARDES concern the challenge of providing programming support for computer systems with continuously running services and applications, that operate at multiple physical and logical locations, that are constantly introduced, deployed, and combined, that interact, fail and evolve all the time. Programming such systems – called *open programming* by the designers of the Alice programming language [95] — is challenging because it requires the combination of several features, notably: (i) *modularity*, i.e. the ability to build systems by combining and composing multiple elements; (ii) *security*, i.e. the ability to deal with unknown and untrusted system elements, and to enforce if necessary their isolation from the rest of the system; (iii) *distribution*, i.e. the ability to build systems out of multiple elements executing separately on multiple interconnected machines, which operate at different speed and under different capacity constraints, and which may fail independently; (iv) *concurrency*, i.e. the ability to deal with multiple concurrent events, and non-sequential tasks; and (v) *dynamicity*, i.e. the ability to introduce new systems, as well as to remove, update and modify existing ones, possibly during their execution.

The rigorous study of programming features relate to the study of programming language constructs and semantics [91], [107], in general. Each of the features mentioned above has been, and continues to be, the subject of active research on its own. Combining them into a practical programming language with a well-defined formal semantics, however, is still an open question. Recent languages that provide relevant background for SARDES' research are:

- For their support of dynamic notions of modules and software components: Acute [101], Alice [95], [96], ArchJava [49], Classages [85], Erlang [51], Oz [107], and Scala [92].
- For their security and failure management features: Acute, E [89], Erlang and Oz [63].
- For their support for concurrent and distributed execution, Acute, Alice, JoCaml [68], E, Erlang, Klaim [56], and Oz.

3.3. Software infrastructure

The SARDES approach to software infrastructure is both architecture-based and language-based: architecture-based for it relies on an explicit component structure for runtime reconfiguration, and language-based for it relies on a high-level type safe programming language as a basis for operating system and middleware construction. Exploiting high-level programming languages for operating system construction [104] has a long history, with systems such as Oberon [108], SPIN [55] or JX [69]. More recent and relevant developments for SARDES are:

- The developments around the Singularity project at Microsoft Research [67], [75], which illustrates the use of language-based software isolation for building a secure operating system kernel.

- The seL4 project [70], [80], which developed a formal verification of a modern operating system microkernel using the Isabelle/HOL theorem prover.
- The development of operating system kernels for multicore hardware architectures such as Corey [58] and Barrelfish [53].
- The development of efficient run-time for event-based programming on multicore systems such as libasync [109], [82].

3.4. System management and control

Management (or *Administration*) is the function that aims at maintaining a system's ability to provide its specified services, with a prescribed quality of service. We approach management as a *control* activity, involving an event-reaction loop: the management system detects events that may alter the ability of the managed system to perform its function, and reacts to these events by trying to restore this ability. The operations performed under system and application administration include observation and monitoring, configuration and deployment, resource management, performance management, and fault management.

Up to now, administration tasks have mainly been performed in an ad-hoc fashion. A great deal of the knowledge needed for administration tasks is not formalized and is part of the administrators' know-how and experience. As the size and complexity of the systems and applications are increasing, the costs related to administration are taking up a major part of the total information processing budgets, and the difficulty of the administration tasks tends to approach the limits of the administrators' skills. For example, an analysis of the causes of failures of Internet services [93] shows that most of the service's downtime may be attributed to management errors (e.g. wrong configuration), and that software failures come second. In the same vein, unexpected variations of the load are difficult to manage, since they require short reaction times, which human administrators are not able to achieve.

The above motivates a new approach, in which a significant part of management-related functions is performed automatically, with minimal human intervention. This is the goal of the so-called *autonomic computing* movement [76]. Several research projects [46] are active in this area. [77], [74] are recent surveys of the main research problems related to autonomic computing. Of particular importance for SARDES' work are the issues associated with configuration, deployment and reconfiguration [66], and techniques for constructing control algorithms in the decision stage of administration feedback loops, including discrete control techniques [61], and continuous ones [71].

Management and control functions built by SARDES require also the development of distributed algorithms [86], [106] at different scales, from algorithms for multiprocessor architectures [73] to algorithms for cloud computing [88] and for dynamic peer-to-peer computing systems [50], [94]. Of particular relevance in the latter contexts are epidemic protocols such as gossip protocols [102] because of their natural resilience to node dynamicity or *churn*, an inherent scalability.

4. Software

4.1. JADE

Participants: Noël De Palma [correspondant], Fabienne Boyer.

JADE is a framework for the construction of autonomic systems using the FRACTAL reflective component model. The controlled system is described in terms of an assembly of components equipped with elementary management capabilities. This description, in turn, is the base of the feedback control loops that implement various self-management functions. Legacy applications are managed by wrapping them into components. Since JADE is itself developed using the FRACTAL component model, management functions can also apply to management subsystems.

JADE is available under a Cecill-C open source licence, as part of the JASMINe OW2 open source project. See http://jasmine.ow2.org.

- ACM: D.2.9 Management
- Keywords: Distributed Application Management
- Software benefit: JADE facilitates the construction of self-managed cluster systems as component-based control loops. JADE features a unique architecture that enables true self-repair in the sense that the management subsystem in a JADE system can manage faults occurring within the management subsystem itself. The JADE framework includes facilities for automatic softare deployment and configuration.
- License: CeCILL-C
- Type of human computer interaction: Administration consoles - Configuration files for wrapping legacy systems
- OS/Middleware: Windows / LinuxJRE >= 1.4
- Programming language: Java
- Documentation: javadoc

4.2. MELY

Participants: Fabien Gaud, Sylvain Genevès, Baptiste Lepers, Renaud Lachaize [correspondant], Fabien Mottet, Vivien Quéma.

MELY is an event-driven programming run-time for multicore systems. MELY relies on an event coloring model introduced by the libasync event programming library developed by MIT to allow programmers to inject support for safe parallel execution of multiple event handlers through code annotations. MELY introduces a novel workstealing algorithm to dynamically balance the execution of event handlers on multiple processing cores.

MELY should be released in 2011 under an LGPL licence.

- ACM: D.1.3 Concurrent Programming
- Keywords: Event-driven programming, multicore systems
- Software benefit: MELY provides an easy to use model for concurrent event-driven programming that includes novel heuristics for scalable execution on multicore systems and features an innovative run-time design that minimizes the inherent costs of the workstealing policy used for load balancing.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Linux
- Programming language: C

4.3. AAC_tactics

Participants: Thomas Braibant, Damien Pous [correspondant].

AAC_tactics is a plugin for the Coq proof-assistant that implements new proof tactics for rewriting modulo associativity and commutativity. It is available at http://sardes.inrialpes.fr/~braibant/aac_tactics and as part of the Coq distribution.

- ACM: D.2.4 Software/Program Verification
- Keywords: Rewriting, rewriting modulo AC, proof tactics, proof assistant
- Software benefit: AAC_tactics provides novel efficient proof tactics for rewriting modulo associativity and commutativity.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.4. ATBR

Participants: Thomas Braibant, Damien Pous [correspondant].

ATBR (Algebraic Tools for Binary Relations) is library for the Coq proof assistant that implements new proof tactics for reasoning with binary relations. Its main tactics implements a decision procedure for inequalities in Kleene algebras. It is available at http://sardes.inrialpes.fr/~braibant/atbr and as part of the Coq distribution contributed modules.

- ACM: D.2.4 Software/Program Verification
- Keywords: Binary relations, Kleene algebras, proof tactics, proof assistant
- Software benefit: ATBR provides new proof tactics for reasoning with binary relations.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.5. MoKa

Participants: Jean Arnaud, Sara Bouchenak [correspondant].

MoKa is a software framework for the modeling and capacity planning of distributed systems. It first provides a set of tools to build analytical models that describe the behavior of distributed computing systems, in terms of performance, availability, cost. The framework allows to include several model algorithms and to compare them regarding their accuracy and their efficiency. Furthermore, MoKa provides a set of tools to build capacity planning methods. A capacity planning method allows to find a distributed system configuration that guarantee given quality-of-service objectives. MoKa is able to include different capacity planning algorithms and to compare them regarding their efficiency and the optimality of their results. It is available at http://sardes.inrialpes.fr/research/moka.

- ACM: D.2.9 Management
- Keywords: System management, capacity planning, performance management
- Software benefit: MoKa provides modeling, capacity planning and performance management facilities for application server clusters. Thanks to its model-based capacity planning, MoKa is able to enforce service level objectives while minimizing the service cost.
- License: LGPL
- Type of human computer interaction: Web interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, AspectJ

4.6. ConSer

Participant: Sara Bouchenak [correspondant].

ConSer is a software framework for admission control in servers systems. It implements a fluid model that exhibits the dynamics and behavior of a server system in terms of service performance and availability. ConSer implements various novel admission control laws for servers that allow to trade-off and meet given service level objectives. ConSer's modeling and control laws algorithms are implemented following a proxy-based approach for more transparency.

- ACM: D.2.9 Management
- Keywords: System management, performance management, admission control
- Software benefit: ConSer supports a novel approach to admission control in application servers, based on a fluid model and the use of control-theoretic techniques.
- License: TBD
- Type of human computer interaction: Web interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java

5. New Results

5.1. Languages and Foundations: Process algebra

Participants: Damien Pous, Alan Schmitt, Jean-Bernard Stefani, Claudio Mezzina, Cinzia di Giusto.

The goal of this work is to study process algebraic foundations for component-based programming. Because of the inherently higher-order character of dynamic configuration operations (modelled e.g. by the passivation construct of the Kell calculus [100]), we are led to study new techniques for proving program equivalence in higher-order calculi, to develop new forms of bisimulation, and to study the expressivity of different constructs in higher-order calculi.

In our ongoing collaboration with the INRIA Focus team lead by Davide Sangiorgi at the University of Bologna, we have continued exploring the expressive power of the higher-order pi-calculus. In particular, we have shown that the biadic variant of the calculus is strictly more expressive than the monadic variant [39].

Early in 2010, Sergueï Lenglet has successfully defended his PhD thesis [16], where he has shown the first characterization of weak contextual equivalence for calculi with passivation and applied it to the Seal and Kell calculi.

In our collaboration with the Plume team at LIP (Daniel Hirschkoff) we have continued our study of π-calculus fragments and of their expressivity. We have proven a new congruence result for the pi-calculus: bisimilarity is a congruence in the sub-calculus that does not include restriction nor sum, and features top-level replications only [37].

Our collaboration with the Focus team in Bologna has been extended with another topic on reversible concurrent models of computation. The notion of reversible computation already has a long history in computer science [54]. Nowadays, it is attracting increasing interest because of its applications in diverse fields, including hardware design, biological modelling, program debugging and testing, and quantum computing. We are interested in investigating whether a reversible programming model can be used as a basis for building dependable, component-based distributed systems. Our initial investigations have focused on extending the higher-order π-calculus (HOπ)) with reversibility features. We have shown how to derive a reversible form for HOπ that preserves its structural congruence and we have shown that, surprisingly enough, the obtained reversible HOπ can be faithfully encoded (up to weak barbed bisimilarity) in HOπ [38].

5.2. Languages and Foundations: Proof tactics

Participants: Damien Pous, Thomas Braibant.

The goal of this work is to develop proof-assistant-based tools for verifying distributed systems and distributed abstract machines. In particular, we aim to support the derivation of fully formal proofs of correctness for abstract machines supporting the component-based languages and programming models we develop.

We have continued the development of proof tactics for the Coq proof-assistant for reasoning with binary relations and Kleene algebras [28], and developed proof tactics for rewriting modulo associativity and commutativity. The resulting Coq library and plugin are now part of the Coq distribution.

These substantial formal developments have given rise to unexpected theoretical questions. In particular they have lead us to consider a new notion of colouring for proof nets of cyclic linear logic and to define an original optimization of the standard proof search algorithm [40].

5.3. System support: Support for efficient multicore programming

Participants: Vivien Quéma, Renaud Lachaize, Fabien Gaud, Baptiste Lepers, Sylvain Genevès, Fabien Mottet.

The goal of this work is to develop efficient run-time support for multicore programming. We have targeted this year two main aspects: efficient support for multicore event-driven systems, and a general study to improve the scalability of servers deployed on multicore hardware. Both topics are at the core of the PhD thesis of Fabien Gaud (defended in December 2010 [15]) and are described below.

Many high-performance communicating systems are designed using the event-driven paradigm. As multicore platforms are now pervasive, it becomes crucial for such systems to take advantage of the available hardware parallelism. *Event-coloring* is a promising approach in this regard: it allows programmers to simply and progressively inject support for the safe, parallel execution of multiple event handlers through the use of annotations. In order to balance the load on available cores, the *event-coloring* runtimes rely on a workstealing algorithm in charge of dynamically balancing the execution of event handlers on the available cores. We have studied the impact of the workstealing algorithm on the overall system performance. We observed that the only existing workstealing algorithm designed for event-coloring runtimes is not always efficient: for instance, it causes a 33% performance degradation on a Web server. We have thus introduced several enhancements to improve the workstealing behavior : (i) a set of heuristics to improve the selection of stolen events, and (ii) an improved runtime design minimizing the inherent costs of workstealing. An evaluation using both microbenchmarks and real applications, a Web server and the Secure File Server (SFS), shows that our system consistently outperforms the state-of-the-art *event-coloring* runtime (Libasync-smp), with and without workstealing. In particular, our new workstealing improves performance by up to +25% compared to MIT's Libasync-smp without workstealing, and by up to +73% compared to the Libasync-smp workstealing algorithm, in the Web server case. This work has been presented at the ICDCS conference in June 2010 [34].

We have also considered the scalability of the popular Apache-PHP web server on NUMA multicore architectures. Our study has revealed a number of performance issues with this software as the number of cores increases. We have developed a detailed methodology and extended some state-of-the-art tools in order to precisely pinpoint the major bottlenecks at various levels of the software stack. We have proposed solutions to remove most of these bottleneck and we are currently investigating the few remaining ones.

5.4. System support: Protocols for resilient systems

Participants: Vivien Quéma, Alessio Pace.

The goal of this work is to develop efficient protocols for building fault resilient systems. We have targeted this year two main aspects: the development of a new abstraction for constructing efficient state machine replication protocols that are tolerant to Byzantine failurs (BFT protocols), and the development of a new gossip protocol that is able to limit the dissemination of spam messages.

Modern BFT protocol implementations are complex pieces of software, encompassing sychronization, networking and crytography, which are notirously difficult to develop, test and prove correct. In collaboration with R. Guerraoui's team at EPFL, we have developed a new abstraction to simplify the development of efficient and correct BFT protocols, where a BFT protocol can be seen as a composition of instances of our abstraction. We have shown the benefits of our abstraction by developing two new BFT protocols: *Azyzzyva* and *Aliph*. *Azyzzyva* is a protocol that mimics *Zyzzyva* (a well-known BFT protocol developed at the U. of Texas at Austin) in best case situations, and that uses PBFT (another well-know BFT protocol developed at MIT) to cover worst case situations. *Aliph* is a new BFT protocol that outperforms previous BFT protocols both in terms of latency and throughput. This work has been presented at the Eurosys 2010 conference, where it won the Best Paper Award [35].

Gossip protocols are an efficient and reliable way to disseminate information. These protocols have nevertheless a drawback: they are unable to limit the dissemination of spam messages. Indeed, messages are redundantly disseminated in the network and it is enough that a small subset of nodes forward spam messages to have them received by a majority of nodes. In collaboration with Sonia Ben Mkhtar from the LIRIS laboratory in Lyon, we have developed *FireSpam*, a gossiping protocol that is able to limit spam dissemination. In order to make it usable in practice, we designed FireSpam in the BAR model. This model takes into account selfish and malicious behaviors. We have shown in simulations that *FireSpam* drastically limits the dissemination of spam messages, while still ensuring reliable dissemination of good messages.

5.5. System support: End-to-end caching

Participants: Sara Bouchenak, Dàmian Serrano.

Cloud Computing is a paradigm for enabling remote, on-demand access to a virtually infinite set of configurable computing resources. This model aims to provide hardware and software services to customers, while minimizing human efforts in terms of service installation, configuration and maintenance, for both cloud provider and cloud customer. A cloud may have the form of an Infrastructure as a Service (SaaS), a Platform as a Service (PaaS) or a Software as a Service (SaaS). Clouds pose significant challenges to the full elasticity of clouds, their scalibility and their dependability in large scale data management and large scale computing resources.

Caching is a means for high performance and scalability of distributed systems. Although caching solutions have been successfully studied for individual systems such as database systems or web servers, if collectively applied, these solutions violate the coherence of cached data.

We precisely studied this issue in e-Caching, a novel end-to-end caching system. The contribution is twofold: guaranteeing the coherence of cached data at multiple locations of a distributed system, while improving the overall performance of the system. In collaboration with Marta Patino and Ricardo Jimenez from Universidad Politecnica de Madrid, we proposed a novel distributed caching protocol, implemented it and evaluated it with real online services. The experiments showed that e-Caching was successfully able to improve service performance by two orders of magnitude. This work has been submitted for publication.

5.6. Control for adaptive systems: Discrete control for adaptive and reconfigurable systems

Participants: Eric Rutten, Noël De Palma, Gwenaël Delaval, Soufyane Aboubekr.

The goal of this work is to apply control techniques based on the behavioral model of reactive automata and the algorithmic techniques of discrete controller synthesis. We adopt the synchronous approach to reactive systems, and use an associated effective controller synthesis tool, Sigali, developed at INRIA Rennes. Both are integrated into a programming language, called BZR, and its compiler, as an extension of the Heptagon language. We thus have a complete tool-supported method from control modeling down to concrete execution, considering different execution models, and targetting either software or hardware. We explore control theory for computer science, as an original alernative to computer science for control (as more usually in embedded systems), and to classical discrete control theory (as more usually applied to manufacturing). We are exploring several target application domains, where we expect to find commonalities in the control problems, and variations in the definitions of configurations, and in the criteria motivating adaptation.

We have obtained this year the following results:

- In cooperation with the VerTecs team (INRIA Rennes), the Pop Art team (INRIA Grenoble) and INSA Lyon we have developed a technique for discrete controller synthesis (DCS), with optimal synthesis on bounded paths, in order to model, design, and optimize fault-tolerant distributed systems, taking into account several criteria[32]. In cooperation with GIPSA-Lab and ENSI Tunis, we have adapted discrete controller synthesis technique to the control of decentralized systems that are composed of several subsystems spread across remote sites [26].
- At the programming language level, we have debveloped a modelling and controller generation language called BZR, which involves DCS in its compilation. BZR is designed and developed in cooperation with the VerTecs team (INRIA Rennes) [30].
- We have developed a technique for designing reconfiguration controllers in the Fractal component-based framework, where discrete control loops automatically enforce safety properties on the interactions between components, concerning, e.g., mutual exclusions, forbidden or imposed sequences. This work was presented at the conference CBSE 2010, where it won the Best Paper Award [31].

- We have integrated BZR with Orccad, a programming environment for real-time control systems, in cooperation with the NeCS and SED teams at INRIA Grenoble [43].
- We have started investigating administration loops in a language virtual machine, and the coordination of such loops, especially in relation with green computing problems [29].
- We have started an investigation on the formal modelling and control of dynamic reconfiguration in FPGA circuits, in cooperation with the DaRT team (INRIA Lille) and the Lab-STICC laboratory in Lorient [36], building upon earlier work related to the MARTE framework [33], [24].

5.7. Control for adaptive systems: Self-tuning for Internet services

Participants: Sara Bouchenak, Jean Arnaud.

This work aims at building SLA (Service Level Agreement) management and self-tuning functions for Web application servers. In particular, we address the issue of providing guarantees on both service performance and service availability, two criteria of service quality that are usually handled separetely, and can be seen as antagonistic (high service availability is usually obtained at the expense of performance, and good performance has usually an impact on availability). We have developed two sets of techniques to achieve self-tuning with guarantees on service level objectives (SLOs) such as maximum response times and maximum abandon rates.

The first set of techniques applies fluid models from control theory to modeling, capacity planning, system control and provisioning of server systems. Our main contributions are:

1. a novel model that reflects the non-linear behavior of server systems,
2. a dynamically and automatically evolving model state based on online monitoring in order to reflect service workload variation,
3. novel control laws that take into account both performance and availability SLOs to dynamically and automatically control the system and provision its resources accordingly.

These techniques were proposed in the form of theoretical model and control laws, implemented in a Java-based prototype called *ConSer*, and successfully evaluated in a real warehouse online service running on a database system [21]. This work is conducted in the context of a collaboration between SARDES and the NeCS INRIA Project-Team. This work is the subject of the PhD thesis of Luc Malrait, who is co-advised by Nicolas Marchand (NeCS) and Sara Bouchenak (SARDES).

The second set of techniques makes use of queuing theory to model, provision and plan the capacity of Internet services deployed on clusters of computers, as is usually the case of e-commerce services. Our main contributions consist of:

1. an extended queuing model that takes into account the distribution and parallelism of cluster-based distributed systems, and allows to predict system performance and availability,
2. a novel approach for dynamically and automatically configuring model state, which reflects workload changes and does not require system administrators to perform offline calibration of the model, a technically tricky phase usually necessary prior to the use of these types models,
3. a novel control algorithm that takes into account both performance and availability SLOs while minimizing system costs; it applies dynamic and automatic configuration and provisioning of cluster-based systems with necessary and sufficient resources that guarantee target service performance and availability.

These techniques were proposed in the form of a theoretical extension of the well-known MVA queuing model, the specification of a capacity planning and provisioning algorithm, the design of an online distributed monitoring mechanism of cluster-based systems, and the implementation of a Java-based software prototype called *MoKa* running in realistic distributed Web applications running on Web servers and database servers. This work is the subject of the Ph.D. thesis of Jean Arnaud, defended in September 2010 [13], and has been presented at the ACM SAC conference [25] and in a book chapter [41].

5.8. Virtual Machine

Participants: Olivier gruber, Fabienne Boyer, Ludovic Demontes, Clément Deschamps.

The goal of this work is to develop a new meta-ciruclar virtual machine for the Java language extended with a notion of component, that provides native support for the dynamic reconfiguration of applications and of the virtual machine, and that constitutes its own operating system (in order to run directly on hardware without requiring indpendent operating system support).

A first prototype of the virtual machine has been developed that includes a novel fault-tolerant algorithm for supporting online consistent reconfigurations. In collaboration with the VASY team at INRIA Grenoble, we have formally specified the reconfiguration algorithm and verified it using the CADP toolset developed by VASY. This work has been submitted to publication.

6. Contracts and Grants with Industry

6.1. Contracts with Industry

- PhD advising grant Loris Bouzonnet, funded by Bull.
- PhD advising grant Quentin Sabah, funded by STMicroelectronics.
- PhD advising grant Xavier Etchevers, funded by Orange Labs.

6.2. Regional Initiatives

6.2.1. Aravis (ANR-Minalogic)

Participants: Vivien Quéma, Renaud Lachaize, Fabien Gaud, Sylvain Genevès, Fabien Mottet, Baptiste Lepers.

The ARAVIS project aims at addressing the challenges raised, both at the hardware and software levels, by the production of highly integrated multiprocessor systems on chip (MPSoCs) designed for demanding applications such as video encoding/decoding and software-defined radio communications. Due to the complexity of the manufacturing process, the latest generations of chips exhibit peculiar features that must be taken into account : (i) massively parrallel processing units, (ii) irregular behavior and aging of the processing units due to unavoidable defects of the manufacturing process. The ARAVIS project strives to provide a hardware and software platform suited to the adaptation requirements raised by the needs of such emerging hardware technologies and applications. The proposed approach encompasses three contributions: (i) a symmetric hardware architecture based on an asynchronous interconnect with integrated voltage/frequency scaling, (ii) a set of regulation algorithms based on control theory to optimize quality of service and energy consumption, (iii) a component-based runtime environment and related software tools to ease the dynamic management of applications and execution resources.

The project partners are STMicroelectronics, CEA-LETI, TIMA and INRIA (Necs and Sardes project teams). The project runs from October 2007 to September 2010.

6.2.2. MIND (ANR-Minalogic)

Participants: Eric Rutten, Jean-Bernard Stefani, Gwenaël Delaval, Soufyane Aboubekr, Cinzia di Giusto.

The MIND project aims to develop an industrial technology for component-based construction of embedded systems, based on the Fractal component model.

This includes the development of programming languages (extended C, ADL, IDL), a chain for compiling software architecture descriptions and generating code, and a graphical IDE integrated to Eclipse. In addition, the project aims to study extensions and refinements to the Fractal model suitable for dealing with non-functional aspects such as real-time and priority constraints, and its integration with the BIP component model developed at the Verimag laboratory.

The project partners include STMicroelectronics, CEA, INRIA (Adam and Sardes prokect teams), Schneider. The project runs from October 2008 to October 2010.

6.3. National Initiatives

6.3.1. ASR Network

The Sardes team is a member of the CNRS research network GDR 725 ASR "Architecture, Système et Réseau". See http://asr.univ-perp.fr/.

6.3.2. PiCoq (ANR project)

Participants: Damien Pous, Alan Schmitt, Jean-Bernard Stefani, Thomas Braibant.

The goal of the PiCoq project is to develop an environment for the formal verification of properties of distributed, component-based programs. The project's approach approach lies at the interface between two research areas: concurrency theory and proof assistants. Achieving this goal relies on three scientific advances, which the project intends to address:

- Finding mathematical frameworks that ease modular reasoning about concurrent and distributed systems: due to their large size and complex interactions, distributed systems cannot be analysed in a global way. They have to be decomposed into modular components, whose individual behaviour can be understood.
- Improving existing proof techniques for distributed/modular systems: while behavioural theories of first-order concurrent languages are well understood, this is not the case for higher-order ones. We also need to generalise well-known modular techniques that have been developed for first-order languages to facilitate formalisation in a proof assistant, where source code redundancies should be avoided.
- Defining core calculi that both reflect concrete practice in distributed component programming and enjoy nice properties w.r.t. behavioural equivalences.

The project partners include INRIA (Sardes), LIP (Plume team), and Université de Savoie. the project runs from November 2010 to October 2014.

6.3.3. Project MyCloud (ANR project)

Participants: Sara Bouchenak, Dàmian Serrano.

The objective of the MyCloud project is to define and implement a novel cloud model: *SLAaaS* (*SLA aware Service*). The SLAaaS model enriches the general paradigm of Cloud Computing, and enables systematic and transparent integration of service levels and SLA to the cloud. SLAaaS is orthogonal to IaaS, PaaS and SaaS clouds and may apply to any of them. The MyCloud project takes into account both the cloud provider and cloud customer points of view. From cloud provider's point of view, MyCloud proposes autonomic SLA management to handle performance, availability, energy and cost issues in the cloud. An innovative approach combines control theory techniques with distributed algorithms and language support in order to build autonomic elastic clouds. Novel models, control laws, distributed algorithms and languages will be proposed for automated provisioning, configuration and deployment of cloud services to meet SLA requirements, while tackling scalability and dynamics issues. On the other hand from cloud customer's point of view, the MyCloud project provides SLA governance. It allows cloud customers to be part of the loop and to ba automatically notified about the state of the cloud, such as SLA violation and cloud energy consumption. The former provides more transparecy about SLA guaranties, and the latter aims to raise customers' awareness about cloud's energy footprint.

The project partners are INRIA (Sardes is the project coordinator), Grenoble; LIP6, Paris; EMN, Nantes; We Are Cloud, Montpellier; Elastic Grid LLC, USA.

The project runs from November 2010 to October 2013.

6.3.4. Famous (ANR project)

Participants: Eric Rutten, Xin An.

The FAMOUS project (FAst Modeling and Design FlOw for Dynamically ReconfigUrable Systems) intends to make reconfigurable hardware systems design easier and faster, by introducing a complete methodology that takes the reconfigurability of the hardware as an essential design concept and proposes the necessary mechanisms to fully exploit those capabilities at runtime. The tool under development in this project is expected to be used by both indstrial designers and academic researchers, especially for modern application system specific design such as smart cameras, image and video processing, etc.

The project partners are INRIA (Sardes in Grenoble and DaRT in Lille), Université de Bretagne Sud, Université de Bourgogne, Sodius.

The project runs from December 2009 to November 2013.

6.4. European Initiatives

6.4.1. Play (FP7 ICT Project)

The Play project aims to develop and validate an elastic and reliable architecture for dynamic and complex, event-driven interaction in large highly distributed and heterogeneous service systems. The project intends to define a federated architecture to address the multiplicity and the heterogeneity of service networks, to address Quality of Service (QoS) requirements, and to allow event-driven service networks to scale to Internet-size.

The project partners are INRIA (Oasis in Sophia and Sardes in Grenoble), Institute of Communication and Computer Systems NTUA (GR), EBM WebSourcing (FR), ARMINES (FR), France Telecom R&D (FR), CIM Grupa DOO (RS).

The project runs from October 2010 to September 2013.

7. Dissemination

7.1. Animation of the scientific community

- S. Bouchenak is an Expert at the European Commission for the evaluation of european projects.
- S. Bouchenak is an officer of ACM SIGOPS France.
- S. Bouchenak and Eric Rutten were Chairs of the ACM EuroSys FeBID 1020 (Feedback Control Implementation and Design in Computing Systems and Networks) workshop.
- S. Bouchenak was a member of the program committee of the CFSE 2011 conference.
- S. Bouchenak set up a bilateral Erasmus agreement between Universidad Politecnica de Madrid and University of Grenoble I.
- F. Boyer was a member of the program committee of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2010).
- F. Boyer was a member of the program committee of the GCM 2010 (Green Computing Middleware) workshop.
- N. De Palma was a member of the program committee of the ICAS 2010 (International Conference on Autonomic and Autonomous Systems) conference.
- N. De Palma was the General Chair of the GCM 2010 (Green Computing Middleware) workshop.
- D. Pous was a member of the program committee of the 21st Journées Françaises des Langages Applicatifs (JFLA).

- V. Quema was the Organizer of the third edition of the Winter School on Hot Topics in Distributed Computing.
- V. Quema was the Publicity chair of ACM EuroSys 2010 conference.
- A. Schmitt is a member of the steering committee of the Journées Françaises des Langages Applicatifs (JFLA).
- J.B. Stefani is a member of the editorial board of the journal *Annals of Telecommunications*.
- J.B. Stefani was a member of the program committee of the ACES-MB 2010 (Model Based Architecting and Construction of Embedded Systems) workshop.
- J.B. Stefani was a member of the program committee of the CBHPC 2010 (Component-Based High-Performance Computing) workshop.
- J.B. Stefani is a member of the Technology Council of STMicroelectronics.

7.2. Teaching

- S. Bouchenak, F. Boyer, N. De Palma, O. Gruber and R. Lachaize have full teaching assignments at U. Joseph Fourier.
- F. Boyer is teaching the Operating Systems and Distributed Systems Master level courses at the Polytech/RICM (U. Joseph Fournier).
- F. Boyer is head of the Systems and Network Master courses M2PGI at U. Joseph Fourier.
- N. De Palma is head of the Distributed Systems and Applications Bachelor level courses at Ensimag (INP Grenoble).
- O. Gruber, R. Lachaize and V. Quema are teaching the Distributed Systems and Advanced Operating Systems course of the U. Grenoble International Master MOSIG.
- D. Pous and A. Schmitt are teaching the Bisimulations and Process Calculi course at the MSTII Doctoral School (U. Grenoble).
- D. Pous is teaching a Bachelor level course on Natural Deduction at U. Joseph Fourier.
- A. Schmitt is teaching a Bachelor level course on Commputational Models at U. Joseph Fourier.

8. Bibliography

Major publications by the team in recent years

[1] É. BRUNETON, T. COUPAYE, M. LECLERCQ, V. QUÉMA, J.-B. STEFANI. *The Fractal Component Model and its Support in Java*, in "Software – Practice and Experience (SP&E)", 2006, vol. 36, n^{o} 11-12.

[2] G. DELAVAL, É. RUTTEN. *Reactive Model-Based Control of Reconfiguration in the Fractal Component-Based Model*, in "13th International Symposium on Component-Based Software Engineering (CBSE)", Lecture Notes in Computer Science, Springer, 2010, vol. 6092.

[3] J. N. FOSTER, M. B. GREENWALD, J. T. MOORE, B. C. PIERCE, A. SCHMITT. *Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem*, in "ACM Trans. Program. Lang. Syst.", 2007, vol. 29, n^{o} 3.

[4] P. GENEVÈS, N. LAYAIDA, A. SCHMITT. *Efficient Static Analysis of XML Paths and Types*, in "PLDI '07: Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation", ACM Press, 2007.

[5] R. GUERRAOUI, N. KNEZEVIC, V. QUÉMA, M. VUKOLIC. *The next 700 BFT protocols*, in "5th European Conference on Computer systems (Eurosys)", ACM, 2010.

[6] R. GUERRAOUI, R. R. LEVY, B. POCHON, V. QUÉMA. *Throughput optimal total order broadcast for cluster environments*, in "ACM Trans. Comput. Syst.", 2010, vol. 28, n^{o} 2.

[7] M. LECLERCQ, A. E. ÖZCAN, V. QUÉMA, J.-B. STEFANI. *Supporting Heterogeneous Architecture Descriptions in an Extensible Toolset*, in "29th International Conference on Software Engineering (ICSE)", IEEE Computer Society, 2007.

[8] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Howe's method for calculi with passivation*, in "20th International Conference on Concurrency Theory (CONCUR 2009)", Lecture Notes in Computer Science, Springer, 2009.

[9] L. MALRAIT, S. BOUCHENAK, N. MARCHAND. *Fluid modeling and control for server system performance and availability*, in "39th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)", 2009.

[10] J. PHILIPPE, N. DE PALMA, F. BOYER, O. GRUBER. *Self-adaptation of service level in distributed systems*, in "Softw., Pract. Exper.", 2010, vol. 40, n^{o} 3.

[11] A. SCHMITT, J.-B. STEFANI. *The Kell Calculus: A Family of Higher-Order Distributed Process Calculi*, in "Global Computing", Lecture Notes in Computer Science, Springer, 2005, vol. 3267.

[12] S. SICARD, F. BOYER, N. DE PALMA. *Using Components for Architecture-Based Management: The Self-Repair Case*, in "30th International Conference on Software Engineering (ICSE)", ACM, 2008.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[13] J. ARNAUD. *Performance, disponibilité et coût de services Internet adaptatifs*, Université de Grenoble, Sep 2010.

[14] S. BOUCHENAK. *Performance et disponibilité de srvices répartis: des mécanismes intergiciels au contrôle adaptatif*, Université de Grenoble, Dec 2010, Habilitation à Diriger des Recherches.

[15] F. GAUD. *Etude et amélioration de la performance des serveurs de données pour les architectures multi-cœurs*, Université de Grenoble, Dec 2010.

[16] S. LENGLET. *Bisimulations dans les calculs avec passivation*, Université de Grenoble, Jan 2010.

[17] M. LIENHARDT. *Components and Types*, Université de Grenoble, Sep 2010.

[18] V. QUÉMA. *Contributions to Building Efficient and Robust State-Machine Replication Protocols*, Université de Grenoble, Nov 2010, Habilitation à Diriger des Recherches.

Articles in International Peer-Reviewed Journal

[19] M. D. FLOURIS, R. LACHAIZE, K. CHASAPIS, A. BILAS. *Extensible Block-Level Storage Virtualization in Cluster-Based Systems*, in "Journal of Parallel and Distributed Computing", 2010, vol. 70, n° 8.

[20] R. GUERRAOUI, R. R. LEVY, B. POCHON, V. QUÉMA. *Throughput optimal total order broadcast for cluster environments*, in "ACM Trans. Comput. Syst.", 2010, vol. 28, n° 2.

[21] L. MALRAIT, S. BOUCHENAK, N. MARCHAND. *Experience with ConSer: A System for Server Control Through Fluid Modeling*, in "IEEE Transactions on Computers", 2010.

[22] J. PHILIPPE, N. DE PALMA, F. BOYER, O. GRUBER. *Auto-adaptation du niveau de service dans les systèmes distribués*, in "Technique et Science Informatiques", 2010, vol. 29, n° 3.

[23] J. PHILIPPE, N. DE PALMA, F. BOYER, O. GRUBER. *Self-adaptation of service level in distributed systems*, in "Software, Practice and Experience", 2010, vol. 40, n° 3.

[24] I. R. QUADRI, H. YU, A. GAMATIÉ, S. MEFTALI, J.-L. DEKEYSER, É. RUTTEN. *Targeting Reconfigurable FPGA based SoCs using the MARTE UML profile: from high abstraction levels to code generation*, in "International Journal of Embedded Systems", Sep 2010.

International Peer-Reviewed Conference/Proceedings

[25] J. ARNAUD, S. BOUCHENAK. *Adaptive internet services through performance and availability control*, in "25th ACM Symposium on Applied Computing (SAC)", ACM, 2010.

[26] M. A. BELHAJ SEBOUI, N. BEN HADJ ALOUANE, G. DELAVAL, É. RUTTEN, M. YEDDES. *A Decentralized Supervisory Control Approach For Distributed Adaptive Systems*, in "4th International Workshop on Verification and Evaluation of Computer and Communication Systems (VECoS)", 2010.

[27] S. BEN MOKHTAR, A. PACE, V. QUÉMA. *FireSpam: Spam Resilient Gossiping in the BAR Model*, in "29th IEEE Symposium on Reliable Distributed Systems (SRDS)", IEEE, 2010.

[28] T. BRAIBANT, D. POUS. *An Efficient Coq Tactic for Deciding Kleene Algebras*, in "1st International Conference on Interactive Theorem Proving", Lecture Notes in Computer Science, Springer, 2010, vol. 6172.

[29] N. DE PALMA, G. DELAVAL, É. RUTTEN. *QoS and Energy Management Coordination using Discrete Controller Synthesis*, in "1st International Workshop on Green Computing Middleware (GCM), in conjunction with ACM/IFIP/USENIX 11th International Middleware Conference", 2010.

[30] G. DELAVAL, H. MARCHAND, É. RUTTEN. *Contracts for modular discrete controller synthesis*, in "ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)", ACM, 2010.

[31] G. DELAVAL, É. RUTTEN. *Reactive Model-Based Control of Reconfiguration in the Fractal Component-Based Model*, in "13th International Symposium on Component-Based Software Engineering (CBSE)", Lecture Notes in Computer Science, Springer, 2010, vol. 6092.

[32] E. DUMITRESCU, A. GIRAULT, H. MARCHAND, É. RUTTEN. *Multicriteria optimal reconfiguration of fault-tolerant real-time tasks*, in "Workshop on Discrete Event Systems, WODES'10", IFAC, 2010.

[33] A. GAMATIÉ, V. RUSU, É. RUTTEN. *Operational Semantics of the Marte Repetitive Structure Modeling Concepts for Data-Parallel Applications Design*, in "9th International Symposium on Parallel and Distributed Computing (ISPDC)", IEEE Computer Society, 2010.

[34] F. GAUD, S. GENEVÈS, R. LACHAIZE, B. LEPERS, F. MOTTET, G. MULLER, V. QUÉMA. *Efficient Work-stealing for Multicore Event-Driven Systems*, in "30th International Conference on Distributed Computing Systems (ICDCS)", IEEE Computer Society, 2010.

[35] R. GUERRAOUI, N. KNEZEVIC, V. QUÉMA, M. VUKOLIC. *The next 700 BFT protocols*, in "5th European Conference on Computer systems (Eurosys)", ACM, 2010.

[36] S. GUILLET, F. DE LAMOTTE, É. RUTTEN, G. GOGNIAT, J.-P. DIGUET. *Modeling and formal control of partial dynamic reconfiguration*, in "6th International Conference on ReConFigurable Computing and FPGAs (ReConFig)", 2010.

[37] D. HIRSCHKOFF, D. POUS. *On Bisimilarity and Substitution in Presence of Replication*, in "37th International Colloquium on Automata, Languages and Programming (ICALP), Part II", Lecture Notes in Computer Science, Springer, 2010, vol. 6199.

[38] I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *Reversing Higher-Order Pi*, in "21th International Conference on Concurrency Theory (CONCUR)", Lecture Notes in Computer Science, Springer, 2010, vol. 6269.

[39] I. LANESE, J. A. PÉREZ, D. SANGIORGI, A. SCHMITT. *On the Expressiveness of Polyadic and Synchronous Communication in Higher-Order Process Calculi*, in "37th International Colloquium on Automata, Languages and Programming (ICALP), Part II", Lecture Notes in Computer Science, Springer, 2010, vol. 6199.

[40] D. POUS. *Untyping Typed Algebraic Structures and Colouring Proof Nets of Cyclic Linear Logic*, in "24th International Workshop on Computer Science Logic (CSL)", Lecture Notes in Computer Science, Springer, 2010, vol. 6247.

Scientific Books (or Scientific Book chapters)

[41] J. ARNAUD, S. BOUCHENAK. *Performance, Availability and Cost of Self-Adaptive Internet Services*, in "Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions", IGI Global, 2011.

Books or Proceedings Editing

[42] S. BOUCHENAK, É. RUTTEN (editors). *Best papers from the 5th international workshop on feedback control implementation and design in computing systems and networks (FeBID 2010)*, 2010, vol. 44.

Research Reports

[43] A. S. ABOUBEKR, G. DELAVAL, R. PISSARD-GIBOLLET, É. RUTTEN, D. SIMON. *Automatic generation of discrete handlers of real-time continuous control tasks*, INRIA, Jul 2010, RR-7332.

[44] E. BARCENAS, P. GENEVÈS, N. LAYAIDA, A. SCHMITT. *On the Count of Trees*, INRIA, Aug 2010, RR-7251.

[45] F. GAUD, S. GENEVÈS, R. LACHAIZE, B. LEPERS, F. MOTTET, G. MULLER, V. QUÉMA. *Mely: Efficient Workstealing for Multicore Event-Driven Systems*, INRIA, Jan 2010, RR-7169.

References in notes

[46] *Research Projects in Autonomic Computing*, 2003, IBM Research, http://www.research.ibm.com/autonomic/research/projects.html.

[47] Z. LIU, H. JIFENG (editors). *Mathematical Frameworks for Component Software - Models for Analysis and Synthesis*, World Scientific, 2006.

[48] M. ABADI, L. CARDELLI. *A theory of objects*, Springer, 1996.

[49] J. ALDRICH, C. CHAMBERS, D. NOTKIN. *Architectural Reasoning in ArchJava*, in "Proceedings 16th European Conference on Object-Oriented Programming (ECOOP)", 2002.

[50] S. ANDROUTSELLIS-THEOTOKIS, D. SPINELLIS. *A survey of peer-to-peer content distribution technologies*, in "ACM Comput. Surv.", 2004, vol. 36, n^{o} 4.

[51] J. ARMSTRONG, M. WILLIAMS, C. WIKSTROM, R. VIRDING. *Concurrent Programming in Erlang*, Prentice Hall, 1996.

[52] L. BASS, P. CLEMENTS, R. KAZMAN. *Software Architecture in Practice*, SEI Series in Software Engineering, 2nd edition, Addison-Wesley, 2003.

[53] A. BAUMANN, P. BARHAM, P. DAGAND, T. L. HARRIS, R. ISAACS, S. PETER, T. ROSCOE, A. SCHÜPBACH, A. SINGHANIA. *The multikernel: a new OS architecture for scalable multicore systems*, in "22nd ACM Symposium on Operating Systems Principles (SOSP)", ACM, 2009.

[54] C. BENNETT. *Notes on the history of reversible computation*, in "IBM Journal of Research and Development", 1988, vol. 32, n^{o} 1.

[55] B. BERSHAD, S. SAVAGE, P. PRZEMYSLAW, E. SIRER, M. FIUCZYNSKI, D. BECKER, C. CHAMBERS, S. EGGERS. *Extensibility, Safety and Performance in the SPIN Operating System*, in "Proceedings 15th ACM Symposium on Operating Systems Principles", 1995.

[56] L. BETTINI, V. BONO, R. DE NICOLA, G. FERRARI, D. GORLA, M. LORETI, E. MOGGI, R. PUGLIESE, E. TUOSTO, B. VENNERI. *The KLAIM Project: Theory and Practice*, in "Global Computing: Programming Environments, Languages, Security and Analysis of Systems", Lecture Notes in Computer Science, Springer-Verlag, 2003, n^{o} 2874.

[57] G. S. BLAIR, G. COULSON, A. ANDERSEN, L. BLAIR, M. CLARKE, F. COSTA, H. DURAN-LIMON, T. FITZPATRICK, L. JOHNSTON, R. MOREIRA, N. PARLAVANTZAS, K. SAIKOSKI. *The Design and Implementation of OpenORB v2*, in "IEEE Distributed Systems Online, vol. 2 no 6, Special Issue on Reflective Middleware", 2001.

[58] S. BOYD-WICKIZER, H. CHEN, R. CHEN, Y. MAO, M. F. KAASHOEK, R. MORRIS, A. PESTEREV, L. STEIN, M. WU, Y. DAI, Y. ZHANG, Z. ZHANG. *Corey: an operating system for many cores*, in "8th Usenix Symposium on Operating System Design and Implementation (OSDI '08)", 2008.

[59] J. BRADBURY, J. CORDY, J. DINGEL, M. WERMELINGER. *A Survey of Self-Management in Dynamic Software Architecture Specifications*, in "Proceedings of the 1st ACM SIGSOFT Workshop on Self-Managed Systems, WOSS 2004", ACM, 2004.

[60] M. BUGLIESI, G. CASTAGNA, S. CRAFA. *Access control for mobile agents: the calculus of boxed ambients*, in "ACM. Trans. Prog. Languages and Systems, vol. 26, no 1", 2004.

[61] C. G. CASSANDRAS, S. LAFORTUNE. *Introduction to Discrete Event Systems*, 2nd edition, Springer, 2008.

[62] G. CASTAGNA, J. VITEK, F. Z. NARDELLI. *The Seal Calculus*, in "Inf. Comput.", 2005, vol. 201, n^{o} 1.

[63] R. COLLET. *The Limits of Network Transparency in a Distributed Programming Language*, Université Catholique de Louvain, Belgium, 2007.

[64] G. COULSON, G. S. BLAIR, P. GRACE, F. TAÏANI, A. JOOLIA, K. LEE, J. UEYAMA, T. SIVAHARAN. *A generic component model for building systems software*, in "ACM Trans. Computer Systems", 2008, vol. 26, n^{o} 1.

[65] S. DAL-ZILIO. *Mobile Processes: A Commented Bibliography*, in "Modeling and Verification of Parallel Processes, 4th Summer School, MOVEP 2000", LNCS, Springer, 2001, vol. 2067.

[66] A. DEARLE. *Software Deployment, Past, Present and Future*, in "Future of Software Engineering (FOSE '07)", IEEE, 2007.

[67] M. FAHNDRICH, M. AIKEN, C. HAWBLITZEL, O. HODSON, G. C. HUNT, J. R. LARUS, S. LEVI. *Language Support for Fast and Reliable Message-based Communication in Singularity OS*, in "1st EuroSys Conference", ACM, 2006.

[68] C. FOURNET, F. LE FESSANT, L. MARANGET, A. SCHMITT. *JoCaml: a language for concurrent, distributed and mobile programming*, in "Summer Schol Adv. Functional Programming", LNCS, 2003, vol. 2638.

[69] M. GOLM, M. FELSER, C. WAWERSICH, J. KLEINÖDER. *The JX Operating System*, in "USENIX Annual Technical Conference, General Track", USENIX, 2002.

[70] G. HEISER, K. ELPHINSTONE, I. KUZ, G. KLEIN, S. PETTERS. *Towards Trusworthy Computing Systems: Taking Microkernels to the Next Level*, in "Operating Systems Review", 2007, vol. 41, n^{o} 4.

[71] J. HELLERSTEIN, Y. DAO, S. PAREKH, D. TILBURY. *Feedback Control of Computing Systems*, Wiley-Interscience, 2004.

[72] M. HENNESSY, J. RATHKE, N. YOSHIDA. *SafeDpi: a language for controlling mobile code*, in "Acta Informatica", 2005, vol. 42, n^{o} 4-5.

[73] M. HERLIHY, N. SHAVIT. *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008.

[74] M. C. HUEBSCHER, J. A. MCCANN. *A survey of autonomic computing - degrees, models, and applications*, in "ACM Comput. Surv.", 2008, vol. 40, n° 3.

[75] G. C. HUNT, J. R. LARUS. *Singularity: rethinking the software stack*, in "Operating Systems Review", 2007, vol. 41, n° 2.

[76] J. O. KEPHART, D. M. CHESS. *The Vision of Autonomic Computing*, in "Computer - IEEE Computer Magazine", 2003, vol. 36, n° 1, p. 41–50.

[77] J. O. KEPHART. *Research Challenges of Autonomic Computing*, in "ICSE '05: Proceedings of the 27th international conference on Software engineering", New York, NY, USA, ACM Press, 2005, p. 15–22, http://doi.acm.org/10.1145/1062455.1062464.

[78] G. KICZALES, J. DES RIVIÈRES, D. BOBROW. *The Art of the Metaobject Protocol*, MIT Press, 1991.

[79] G. KICZALES. *Aspect-Oriented Programming*, in "ACM Computing Surveys", 1996, vol. 28, n° 4, 154, http://doi.acm.org/10.1145/242224.242420.

[80] G. KLEIN, K. ELPHINSTONE, G. HEISER, J. ANDRONICK, D. COCK, P. DERRIN, D. ELKADUWE, K. ENGELHARDT, R. KOLANSKI, M. NORRISH, T. SEWELL, H. TUCH, S. WINWOOD. *seL4: formal verification of an OS kernel*, in "22nd ACM Symposium on Operating Systems Principles 2009", ACM, 2009.

[81] F. KON, F. COSTA, G. S. BLAIR, R. H. CAMPBELL. *The Case for Reflective Middleware*, in "Communications of the ACM", June 2002, vol. 45, n° 6, p. 33–38.

[82] M. N. KROHN, E. KOHLER, M. F. KAASHOEK. *Events Can Make Sense*, in "USENIX Annual Technical Conference", 2007.

[83] K. LAU, Z. WANG. *Software Component Models*, in "IEEE Trans. Software Eng.", 2007, vol. 33, n° 10.

[84] G. LEAVENS, M. SITARAMAN. *Foundations of Component-Based Systems*, Cambridge University Press, 2000.

[85] Y. LIU, S. SMITH. *Interaction-Based Programming with Classages*, in "Proceedings OOPSLA", 2005.

[86] N. LYNCH. *Distributed Algorithms*, Morgan Kaufmann, 1996.

[87] N. MEDVIDOVIC, R. N. TAYLOR. *A Classification and Comparison Framework for Software Architecture Description Languages*, in "IEEE Transactions on Software Engineering, vol. 26, no. 1", 2000.

[88] P. MELL, T. GRANCE. *The NIST Definition of Cloud Computing*, 2010, http://csrc.nist.gov/groups/SNS/cloud-computing/.

[89] M. MILLER. *Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control*, Johns Hopkins University, baltimor, Maryland, USA, 2006.

[90] R. MILNER. *Communicating and mobile systems : the π-calculus*, Cambridge University Press, 1999.

[91] J. MITCHELL. *Foundations for Programming Languages*, MIT Press, 1996.

[92] M. ODERSKY, M. ZENGER. *Scalable component abstractions*, in "Proceedings OOPSLA", 2005.

[93] D. OPPENHEIMER, A. GANAPATHI, D. A. PATTERSON. *Why do Internet services fail, and what can be done about it?*, in "4th USENIX Symposium on Internet Technologies and Systems (USITS '03)", March 2003.

[94] J. RISSON, T. MOORS. *Survey of research towards robust peer-to-peer networks: Search methods*, in "Computer Networks", 2006, vol. 50, n^{o} 17.

[95] A. ROSSBERG, D. LE BOTLAN, G. TACK, T. BRUNKLAUS, G. SMOLKA. *Alice Through the Looking Glass*, in "Trends in Functional Programming", Intellect Books, 2006, vol. 5.

[96] A. ROSSBERG. *The Missing Link - Dynamic Components for ML*, in "11th International Conference on Functional Programming", ACM Press, 2006.

[97] D. SANGIORGI. *On the origins of bisimulation and coinduction*, in "ACM Trans. Program. Lang. Syst.", 2009, vol. 31, n^{o} 4.

[98] D. SANGIORGI. *Bisimulation for Higher-Order Process Calculi*, in "Information and Computation", 1996, vol. 131, n^{o} 2.

[99] D. SANGIORGI, D. WALKER. *The π-calculus: A Theory of Mobile Processes*, Cambridge University Press, 2001.

[100] A. SCHMITT, J.-B. STEFANI. *The Kell Calculus: A Family of Higher-Order Distributed Process Calculi*, in "Global Computing", Lecture Notes in Computer Science, Springer, 2005, vol. 3267.

[101] P. SEWELL, J. LEIFER, K. WANSBROUGH, F. Z. NARDELLI, M. ALLEN-WILLIANS, P. HABOUZIT, V. VAFEIADIS. *Acute: High-level programming language design for distributed computation*, in "Int. Conf. Functional Programming", 2005.

[102] D. SHAH. *Gossip Algorithms*, in "Foundations and Trends in Networking", 2009, vol. 3, n^{o} 1.

[103] M. SHAW, D. GARLAN. *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.

[104] A. SILBERSCHATZ, P. B. GALVIN, G. GAGNE. *Operating System Concepts*, 8th edition, Wiley, 2008.

[105] C. SZYPERSKI. *Component Software, 2nd edition*, Addison-Wesley, 2002.

[106] G. TEL. *Introduction to Distributed Algorithms*, 2nd edition, Cambdrige University Press, 2001.

[107] P. VAN ROY, S. HARIDI. *Concepts, Techniques and Models of Computer Programming*, MIT Press, 2004.

[108] N. WIRTH. *Project Oberon: The Design of an Operating System and Compiler*, ACM Press, 1992.

[109] N. ZELDOVICH, A. YIP, F. DABEK, R. MORRIS, D. MAZIÈRES, M. F. KAASHOEK. *Multiprocessor Support for Event-Driven Programs*, in "USENIX Annual Technical Conference, General Track", 2003.