IN PARTNERSHIP WITH:
CNRS

Ecole des Mines de Nantes

Université Nantes

Activity Report 2015

Project-Team ASCOLA

Aspect and composition languages

IN COLLABORATION WITH: Laboratoire d'Informatique de Nantes Atlantique (LINA)

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Distributed programming and Software engineering

Table of contents

Project-Team ASCOLA

Creation of the Project-Team: 2009 January 01

Keywords:

Computer Science and Digital Science:

1.1.13. - Virtualization
1.1.6. - Cloud
1.1.8. - Security of architectures
1.3. - Distributed Systems
1.6. - Green Computing
2.1. - Programming Languages
2.1.1. - Semantics of programming languages
2.1.10. - Domain-specific languages
2.1.11. - Proof languages
2.1.2. - Object-oriented programming
2.1.3. - Functional programming
2.1.4. - Aspect-oriented programming
2.1.6. - Concurrent programming
2.1.7. - Distributed programming
2.2.1. - Static analysis
2.4.2. - Verification
2.4.3. - Proofs
2.5. - Software engineering
2.6.2. - Middleware
2.6.3. - Virtual machines
3.1.3. - Distributed data
3.1.5. - Control access, privacy
4.5. - Formal methods for security
4.6. - Authentication
4.7. - Access control
4.8. - Privacy-enhancing technologies
7.1. - Parallel and distributed algorithms
7.4. - Logic in Computer Science

Other Research Topics and Application Domains:

3.1. - Sustainable development
4.4. - Energy consumption
4.4.1. - Green computing
5.1. - Factory of the future
6.1.1. - Software engineering
6.1.2. - Software evolution, maintenance
6.5. - Information systems

1. Members

Research Scientists

Adrien Lebre [Inria, Researcher; under delegation from MN]
Nicolas Tabareau [Inria, Researcher]

Faculty Members

Mario Südholt [Team leader, MN, Professor, HdR]
Pierre Cointe [MN, Professor, HdR]
Rémi Douence [MN, Associate Professor]
Hervé Grall [MN, Associate Professor]
Thomas Ledoux [MN, Associate Professor]
Jean-Marc Menaud [MN, Professor, HdR]
Jacques Noyé [MN, Associate Professor]
Jean-Claude Royer [MN, Professor, HdR]

PhD Students

Mohamed Abderrahim [Inria, CIFRE Orange, from Nov 2015]
Walid Benghabrit [MN]
Paul Blouët [MN, co-supervision with Prof. De Meuter, VUB, Belgium]
Simon Boulier [MN, ASN, from sept 2015]
Ronan-Alexandre Cherrueau [MN]
Bastien Confais [Polytech Nantes, CNRS, from Oct 2015]
Frédéric Dumont [MN, CIFRE EasyVirt]
Simon Dupont [MN, CIFRE Sigma]
Alexandre Garnier [MN]
Md Sabbir Hasan [Inria, co-supervision with Prof. Pazat, Myriads team, Inria]
Yacine Hebbal [MN, CIFRE Orange]
Gabriel Lewertowski [Inria, ERC CoqHoTT, from sept 2015]
Yunbo Li [Inria, co-supervision with Dr. Orgerie, Myriads team, Inria]
Florent Marchand de Kerchove de Denterghem [MN]
Linh-Thuy Nguyen [Inria, from Dec 2015]
Jonathan Pastor [MN]
Kevin Quirin [MN, ASN]
Jurgen Van Ham [MN, co-supervision with Prof. Mezini, TU Darmstadt, Germany, until March 2015]

Post-Doctoral Fellows

Frederico Alvares [MN, since Mar 2015]
Mehdi Haddad [MN, until Aug 2015]
Ali Kassem [MN, from Oct 2015]
Jonathan Lejeune [MN, until Aug 2015]
Guillaume Le Louët [MN]
Pierre-Marie Pédrot [Inria, ERC CoqHoTT, from oct 2015]
Rémy Pottier [MN]
Anthony Simonet [Inria, from Oct 2015]

Administrative Assistants

Anne-Claire Binétruy [Inria]
Florence Rogues [MN]

2. Overall Objectives

2.1. Presentation

The research team addresses the general problem of evolving software by developing concepts, languages, implementations and tools for building software architectures based on components and aspects. Its long term goal is the development of new abstractions for the programming of software architectures, their representation in terms of expressive programming languages and their correct and efficient implementation.

We pursue the following objectives:

- New concepts and techniques for the compositional definition and implementation of complex software systems, notably involving crosscutting concerns that cannot be handled modularly using traditional software development approaches.
- New programming techniques and algorithms for resource management in mutualized environments. We provide language abstractions and implementation techniques for large-scale applications in cloud- and grid-based systems, both on the level of (service-based) applications and (virtualized) infrastructures. We develop solutions, in particular, for the optimization of the energy consumption in such environments (data centers ...)
- We develop new formal theories for and apply formal methods to the correctness of software systems. We aim at developing more powerful techniques for theorem proving and enable complex, often dynamic, software systems to be proven correct using program transformations and analysis techniques. We develop solutions, in particular, for the constructive enforcement of security properties on the level of software systems.

Finally, we apply and validate our results based on real-world applications from numerous domains, notably enterprise information systems, the Cloud, and pervasive systems.

3. Research Program

3.1. Overview

Since we mainly work on new concepts for the language-based definition and implementation of complex software systems, we first briefly introduce some basic notions and problems of software components (understood in a broad sense, that is, including modules, objects, architecture description languages and services), aspects, and domain-specific languages. We conclude by presenting the main issues related to distribution and concurrency, in particular related to capacity planning issues that are relevant to our work.

3.2. Software Composition

Modules and services. The idea that building *software components*, i.e., composable prefabricated and parameterized software parts, was key to create an effective software industry was realized very early [75]. At that time, the scope of a component was limited to a single procedure. In the seventies, the growing complexity of software made it necessary to consider a new level of structuring and programming and led to the notions of information hiding, *modules*, and module interconnection languages [82], [58]. Information hiding promotes a black-box model of program development whereby a module implementation, basically a collection of procedures, is strongly encapsulated behind an interface. This makes it possible to guarantee logical invariant *properties* of the data managed by the procedures and, more generally, makes *modular reasoning* possible.

In the context of today's Internet-based information society, components and modules have given rise to *software services* whose compositions are governed by explicit *orchestration or choreography* specifications that support notions of global properties of a service-oriented architecture. These horizontal compositions have, however, to be frequently adapted dynamically. Dynamic adaptations, in particular in the context of software evolution processes, often conflict with a black-box composition model either because of the need for invasive modifications, for instance, in order to optimize resource utilization or modifications to the vertical compositions implementing the high-level services.

Object-Oriented Programming. Classes and objects provide another kind of software component, which makes it necessary to distinguish between *component types* (classes) and *component instances* (objects). Indeed, unlike modules, objects can be created dynamically. Although it is also possible to talk about classes in terms of interfaces and implementations, the encapsulation provided by classes is not as strong as the one provided by modules. This is because, through the use of inheritance, object-oriented languages put the emphasis on *incremental programming* to the detriment of modular programming. This introduces a white-box model of software development and more flexibility is traded for safety as demonstrated by the *fragile base class* issue [78].

Architecture Description Languages. The advent of distributed applications made it necessary to consider more sophisticated connections between the various building blocks of a system. The *software architecture* [87] of a software system describes the system as a composition of *components* and *connectors*, where the connectors capture the *interaction protocols* between the components [46]. It also describes the rationale behind such a given architecture, linking the properties required from the system to its implementation. *Architecture Description Languages* (ADLs) are languages that support architecture-based development [76]. A number of these languages make it possible to generate executable systems from architectural descriptions, provided implementations for the primitive components are available. However, guaranteeing that the implementation conforms to the architecture is an issue.

Protocols. Today, protocols constitute a frequently used means to precisely define, implement, and analyze contracts, notably concerning communication and security properties, between two or more hardware or software entities. They have been used to define interactions between communication layers, security properties of distributed communications, interactions between objects and components, and business processes.

Object interactions [80], component interactions [93], [84] and service orchestrations [59] are most frequently expressed in terms of *regular interaction protocols* that enable basic properties, such as compatibility, substitutability, and deadlocks between components to be defined in terms of basic operations and closure properties of finite-state automata. Furthermore, such properties may be analyzed automatically using, e.g., model checking techniques [56], [65].

However, the limited expressive power of regular languages has led to a number of approaches using more expressive *non-regular* interaction protocols that often provide distribution-specific abstractions, e.g., session types [69], or context-free or turing-complete expressiveness [85], [53]. While these protocol types allow conformance between components to be defined (e.g., using unbounded counters), property verification can only be performed manually or semi-automatically.

3.3. Programming languages for advanced modularization

The main driving force for the structuring means, such as components and modules, is the quest for clean *separation of concerns* [60] on the architectural and programming levels. It has, however, early been noted that concern separation in the presence of crosscutting functionalities requires specific language and implementation level support. Techniques of so-called *computational reflection*, for instance, Smith's 3-Lisp or Kiczales's CLOS meta-object protocol [88], [72] as well as metaprogramming techniques have been developed to cope with this problem but proven unwieldy to use and not amenable to formalization and property analysis due to their generality. Methods and techniques from two fields have been particularly useful in addressing such advanced modularization problems: Aspect-Oriented Software Development as the field concerned with the systematic handling of modularization issues and domain-specific languages that provide declarative and efficient means for the definition of crosscutting functionalities.

Aspect-Oriented Software Development [71], [44] has emerged over the previous decade as the domain of systematic exploration of crosscutting concerns and corresponding support throughout the software development process. The corresponding research efforts have resulted, in particular, in the recognition of *crosscutting* as a fundamental problem of virtually any large-scale application, and the definition and implementation of a large number of aspect-oriented models and languages.

However, most current aspect-oriented models, notably AspectJ [70], rely on pointcuts and advice defined in terms of individual execution events. These models are subject to serious limitations concerning the modularization of crosscutting functionalities in distributed applications, the integration of aspects with other modularization mechanisms such as components, and the provision of correctness guarantees of the resulting AO applications. They do, in particular, only permit the manipulation of distributed applications on a per-host basis, that is, without direct expression of coordination properties relating different distributed entities [89]. Similarly, current approaches for the integration of aspects and (distributed) components do not directly express interaction properties between sets of components but rather seemingly unrelated modifications to individual components [57]. Finally, current formalizations of such aspect models are formulated in terms of low-level semantic abstractions (see, e.g., Wand's et al semantics for AspectJ [92]) and provide only limited support for the analysis of fundamental aspect properties.

Different approaches have been put forward to tackle these problems, in particular, in the context of so-called *stateful* or *history-based aspect languages* [61], [62], which provide pointcut and advice languages that directly express rich relationships between execution events. Such languages have been proposed to directly express coordination and synchronization issues of distributed and concurrent applications [81], [51], [64], provide more concise formal semantics for aspects and enable analysis of their properties [47], [63], [61], [45]. Furthermore, first approaches for the definition of *aspects over protocols* have been proposed, as well as over regular structures [61] and non-regular ones [91], [79], which are helpful for the modular definition and verification of protocols over crosscutting functionalities.

They represent, however, only first results and many important questions concerning these fundamental issues remain open, in particular, concerning the semantics foundations of AOP and the analysis and enforcement of correctness properties governing its, potentially highly invasive, modifications.

Domain-specific languages (DSLs) represent domain knowledge in terms of suitable basic language constructs and their compositions at the language level. By trading generality for abstraction, they enable complex relationships among domain concepts to be expressed concisely and their properties to be expressed and formally analyzed. DSLs have been applied to a large number of domains; they have been particularly popular in the domain of software generation and maintenance [77], [95].

Many modularization techniques and tasks can be naturally expressed by DSLs that are either specialized with respect to the type of modularization constructs, such as a specific brand of software component, or to the compositions that are admissible in the context of an application domain that is targeted by a modular implementation. Moreover, software development and evolution processes can frequently be expressed by transformations between applications implemented using different DSLs that represent an implementation at different abstraction levels or different parts of one application.

Functionalities that crosscut a component-based application, however, complicate such a DSL-based transformational software development process. Since such functionalities belong to another domain than that captured by the components, different DSLs should be composed. Such compositions (including their syntactic expression, semantics and property analysis) have only very partially been explored until now. Furthermore, restricted composition languages and many aspect languages that only match execution events of a specific domain (e.g., specific file accesses in the case of security functionality) and trigger only domain-specific actions clearly are quite similar to DSLs but remain to be explored.

3.4. Distribution and Concurrency

While ASCOLA does not investigate distribution and concurrency as research domains per se (but rather from a software engineering and modularization viewpoint), there are several specific problems and corresponding approaches in these domains that are directly related to its core interests that include the structuring and modularization of large-scale distributed infrastructures and applications. These problems include crosscutting functionalities of distributed and concurrent systems, support for the evolution of distributed software systems, and correctness guarantees for the resulting software systems.

Underlying our interest in these domains is the well-known observation that large-scale distributed applications are subject to *numerous crosscutting functionalities* (such as the transactional behavior in enterprise information systems, the implementation of security policies, and fault recovery strategies). These functionalities are typically partially encapsulated in distributed infrastructures and partially handled in an ad hoc manner by using infrastructure services at the application level. Support for a more principled approach to the development and evolution of distributed software systems in the presence of crosscutting functionalities has been investigated in the field of *open adaptable middleware* [52], [74]. Open middleware design exploits the concept of reflection to provide the desired level of configurability and openness. However, these approaches are subject to several fundamental problems. One important problem is their insufficient, framework-based support that only allows partial modularization of crosscutting functionalities.

There has been some *criticism* on the use of *AspectJ-like aspect models* (which middleware aspect models like that of JBoss AOP are an instance of) for the modularization of distribution and concurrency related concerns, in particular, for transaction concerns [73] and the modularization of the distribution concern itself [89]. Both criticisms are essentially grounded in AspectJ's inability to explicitly represent sophisticated relationships between execution events in a distributed system: such aspects therefore cannot capture the semantic relationships that are essential for the corresponding concerns. History-based aspects, as those proposed by the ASCOLA project-team provide a starting point that is not subject to this problem.

From a point of view of language design and implementation, aspect languages, as well as domain specific languages for distributed and concurrent environments share many characteristics with existing distributed languages: for instance, event monitoring is fundamental for pointcut matching, different synchronization strategies and strategies for code mobility [67] may be used in actions triggered by pointcuts. However, these relationships have only been explored to a small degree. Similarly, the formal semantics and formal properties of aspect languages have not been studied yet for the distributed case and only rudimentarily for the concurrent one [47], [64].

3.5. Security

Security properties and policies over complex service-oriented and standalone applications become ever more important in the context of asynchronous and decentralized communicating systems. Furthermore, they constitute prime examples of crosscutting functionalities that can only be modularized in highly insufficient ways with existing programming language and service models. Security properties and related properties, such as accountability properties, are therefore very frequently awkward to express and difficult to analyze and enforce (provided they can be made explicit in the first place).

Two main issues in this space are particularly problematic from a compositional point of view. First, information flow properties of programming languages, such as flow properties of Javascript [49], and service-based systems [55] are typically specially-tailored to specific properties, as well as difficult to express and analyze. Second, the enforcement of security properties and security policies, especially accountability-related properties [83], [90], is only supported using ad hoc means with rudimentary support for property verification.

The ASCOLA team has recently started to work on providing formal methods, language support and implementation techniques for the modular definition and implementation of information flow properties as well as policy enforcement in service-oriented systems as well as, mostly object-oriented, programming languages.

3.6. Green IT

With the emergence of the Future Internet and the dawn of new IT architecture and computation models such as cloud computing, the usage of data centers (DC) as well as their power consumption increase dramatically

[54]. Besides the ecological impact [68], energy consumption is a predominant criterion for DC providers since it determines the daily cost of their infrastructure. As a consequence, power management becomes one of the main challenges for DC infrastructures and more generally for large-scale distributed systems.

To address this problem, we study two approaches: a workload-driven [50] and power-driven one [86]. As part of the workload-driven solution, we adapt the power consumption of the DC depending on the application workload, and predict this workload to be more reactive. We develop a distributed system from the system to the service-oriented level mainly based on hardware and virtualization capabilities that is managed in a user-transparent fashion. As part of the power-driven approach, we address energy consumption issues through a strong synergy inside the infrastructure software stack and more precisely between applications and resource management systems. This approach is characterized by adapting QoS properties aiming at the best trade-off between cost of energy (typically from the regular electric grid), its availability (for instance, from renewable energy), and service degradation caused, for instance, by application reconfigurations to jobs suspensions.

3.7. Capacity Planning for Large Scale Distributed System

Since the last decade, cloud computing has emerged as both a new economic model for software (provision) and as flexible tools for the management of computing capacity [48]. Nowadays, the major cloud features have become part of the mainstream (virtualization, storage and software image management) and the big market players offer effective cloud-based solutions for resource pooling. It is now possible to deploy virtual infrastructures that involve virtual machines (VMs), middleware, applications, and networks in such a simple manner that a new problem has emerged since 2010: VM sprawl (virtual machine proliferation) that consumes valuable computing, memory, storage and energy resources, thus menacing serious resource shortages. Scientific approaches that address VM sprawl are both based on classical administration techniques like the lifecycle management of a large number of VMs as well as the arbitration and the careful management of all resources consumed and provided by the hosting infrastructure (energy, power, computing, memory, network etc.) [66], [94].

The ASCOLA team investigates fundamental techniques for cloud computing and capacity planning, from infrastructures to the application level. Capacity planning is the process of planning for, analyzing, sizing, managing and optimizing capacity to satisfy demand in a timely manner and at a reasonable cost. Applied to distributed systems like clouds, a capacity planning solution must mainly provide the minimal set of resources necessary for the proper execution of the applications (i.e., to ensure SLA). The main challenges in this context are: scalability, fault tolerance and reactivity of the solution in a large-scale distributed system, the analysis and optimization of resources to minimize the cost (mainly costs related to the energy consumption of datacenters), as well as the profiling and adaptation of applications to ensure useful levels of quality of service (throughput, response time, availability etc.).

Our solutions are mainly based on virtualized infrastructures that we apply from the IaaS to the SaaS levels. We are mainly concerned by the management and the execution of the applications by harnessing virtualization capabilities, the investigation of alternative solutions that aim at optimizing the trade-off between performance and energy costs of both applications and cloud resources, as well as arbitration policies in the cloud in the presence of energy-constrained resources.

4. Application Domains

4.1. Enterprise Information Systems and Services

Large IT infrastructures typically evolve by adding new third-party or internally-developed components, but also frequently by integrating already existing information systems. Integration frequently requires the addition of glue code that mediates between different software components and infrastructures but may also consist in more invasive modifications to implementations, in particular to implement crosscutting functionalities. In more abstract terms, enterprise information systems are subject to structuring problems involving horizontal composition (composition of top-level functionalities) as well as vertical composition (reuse and sharing of implementations among several top-level functionalities). Moreover, information systems have to be more and more dynamic.

Service-Oriented Computing (SOC) that is frequently used for solving some of the integration problems discussed above. Indeed, service-oriented computing has two main advantages:

- Loose-coupling: services are autonomous: they do not require other services to be executed;
- Ease of integration: Services communicate over standard protocols.

Our current work is based on the following observation: similar to other compositional structuring mechanisms, SOAs are subject to the problem of crosscutting functionalities, that is, functionalities that are scattered and tangled over large parts of the architecture and the underlying implementation. Security functionalities, such as access control and monitoring for intrusion detection, are a prime example of such a functionality in that it is not possible to modularize security issues in a well-separated module. Aspect-Oriented Software Development is precisely an application-structuring method that addresses in a systemic way the problem of the lack of modularization facilities for crosscutting functionalities.

We are considering solutions to secure SOAs by providing an aspect-oriented structuring and programming model that allows security functionalities to be modularized. Two levels of research have been identified:

- Service level: as services can be composed to build processes, aspect weaving will deal with the orchestration and the choreography of services.
- Implementation level: as services are abstractly specified, aspect weaving will require to extend service interfaces in order to describe the effects of the executed services on the sensitive resources they control.

In 2015, we have published results on constructive mechanisms for security and accountability properties in service-based systems as well as results on service provisioning problems, in particular, service interoperability and mediation, see Sec. 6.3. Furthermore, we take part in the European project A4Cloud on accountability challenges, that is, the responsible stewardship of third-party data and computations, see Sec. 8.3.

4.2. Capacity Planning in Cluster, Grid and Cloud Computing

Cluster, Grid and more recently Cloud computing platforms aim at delivering large capacities of computing power. These capacities can be used to improve performance (for scientific applications) or availability (e.g., for Internet services hosted by datacenters). These distributed infrastructures consist of a group of coupled computers that work together and may be spread across a LAN (cluster), across a WAN (Grid), and across the Internet (Clouds). Due to their large scale, these architectures require permanent adaptation, from the application to the system level and call for automation of the corresponding adaptation processes. We focus on self-configuration and self-optimization functionalities across the whole software stack: from the lower levels (systems mechanisms such as distributed file systems for instance) to the higher ones (i.e. the applications themselves such as J2EE clustered servers or scientific grid applications).

In 2015, we have proposed VMPlaces, a dedicated framework to evaluate and compare VM placement algorithms. Globally the framework is composed of two major components: the injector and the VM placement algorithm. The injector constitutes the generic part of the framework (i.e. the one you can directly use) while the VM placement algorithm is the component a user wants to study (or compare with other existing algorithms), see Sec. 6.4.

In the energy field, we have designed a set of techniques, named Optiplace, for cloud management with flexible power models through constraint programming. OptiPlace supports external models, named views. Specifically, we have developed a power view, based on generic server models, to define and reduce the power consumption of a datacenter's physical servers. We have shown that OptiPlace behaves at least as good as our previous system, Entropy, requiring as low as half the time to find a solution for the constrained-based placement of tasks for large datacenters.

4.3. Pervasive Systems

Pervasive systems are another class of systems raising interesting challenges in terms of software structuring. Such systems are highly concurrent and distributed. Moreover, they assume a high-level of mobility and context-aware interactions between numerous and heterogeneous devices (laptops, PDAs, smartphones, cameras, electronic appliances...). Programming such systems requires proper support for handling various interfering concerns like software customization and evolution, security, privacy, context-awareness... Additionally, service composition occurs spontaneously at runtime.

Like Pervasive systems, Internet of thing is a major theme of these last ten years. Many research works has been led on the whole chain, from communicating sensors to big data management, through communication middlewares. Few of these works have addressed the problem of gathered data access.

The more a sensor networks senses various data, the more the users panel is heterogeneous. Such an heterogeneity leads to a major problem about data modeling: for each user, to aim at precisely addressing his needs and his needs only; ie to avoid a data representation which would overwhelm the user with all the data sensed from the network, regardless if he needs it or not. To leverage this issue, we propose in [24], [35] a multitree modeling for sensor networks which addresses each of these specific usages.With this modeling comes a domain specific language (DSL) which allows users to manipulate, parse and aggregate information from the sensors.

In 2014, we have extended the language EScala, which integrates reactive programming through events with aspect-oriented and object-oriented mechanisms, see Sec. 6.3.

5. New Software and Platforms

5.1. CSLA

Cloud Service Level Agreement language
KEYWORDS: Cloud computing - Service-level agreement - Elasticity
FUNCTIONAL DESCRIPTION

CSLA, the Cloud Service Level Agreement language, allows the definition of SLA properties for arbitrary Cloud services (XaaS). CSLA addresses QoS uncertainty in unpredictable and dynamic environment and provides a cost model of Cloud computing. Besides the standard formal definition of contracts – comprising validity, parties, services definition and guarantees/violations – CSLA is enriched with features, such as QoS degradation and an advanced penalty model, thus introducing fine-grained language support for Cloud elasticity management.

- Participants: Thomas Ledoux and Md Sabbir Hasan
- Contact: Thomas Ledoux
- URL: http://www.emn.fr/z-info/csla/

5.2. JEScala

FUNCTIONAL DESCRIPTION

JEScala is a Scala library which implements a seamless programming model combining object-oriented, aspect-oriented, event-based and concurrent programming. Events are object members. They can be explicitly triggered as in standard event-based programming but also implicitly triggered, as join points in aspect-oriented programming. Event expressions make it possible to compose events, filter them, and alter their content. Event handlers can be registered, and unregistered, dynamically.

Concurrency can be handled without any explicit thread manipulation. By default, primitive events are synchronous but they can also be declared as asynchronous. This creates concurrency between the source of the event and its handlers. This concurrency can then be handled by composing events with the join operator and creating disjunctions of the resulting events.

- Participants: Jurgen Van Ham, Guido Salvaneschi, Mira Mezini and Jacques Noyé
- Partners: *Technische Universität Darmstadt*
- Contact: Jacques Noyé
- URL: http://www.stg.tu-darmstadt.de/research/jescala_menu/index.en.jsp

5.3. SimGrid

KEYWORDS: Large-scale Emulators - Grid Computing - Distributed Applications
FUNCTIONAL DESCRIPTION

Scientific Instrument for the study of Large-Scale Distributed Systems. SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments.

The contributions by the ASCOLA team are related to the virtualization abstractions in particular the different models to simulate VM manipulations (*e.g.*, the live migration model).

- Participants: Jonathan Rouzaud-Cornabas, Frédéric Suter, Martin Quinson, Arnaud Legrand, Adrien Lebre, Jonathan Pastor, Mario Südholt, Luka Stanisic, Augustin Degomme, Jean-Marc Vincent and Florence Perronnin
- Partners: CNRS - Université de Nancy - University of Hawaii - Université de Reims Champagne-Ardenne - Femto-st
- Contact: Arnaud Legrand
- URL: http://simgrid.gforge.inria.fr/

5.4. VMPlaces

FUNCTIONAL DESCRIPTION

VMPlaces is a dedicated framework to evaluate and compare VM placement algorithms. This framework is composed of two major components: the injector and the VM placement algorithm. The injector is the generic part of the framework (i.e. the one you can directly use) while the VM placement algorithm is the part you want to study (or compare with available algorithms). Currently, the VMPlaceS is released with three algorithms:

Entropy, a centralized approach using a constraint programming approach to solve the placement/reconfiguration VM problem

Snooze, a hierarchical approach where each manager of a group invokes Entropy to solve the placement/reconfiguration VM problem. Note that in the original implementation of Snooze, it is using a specific heuristic to solve the placement/reconfiguration VM problem. As the sake of simplicity, we have simply reused the entropy scheduling code.

DVMS, a distributed approach that dynamically partitions the system and invokes Entropy on each partition.

- Participants: Adrien Lebre, Jonathan Pastor, and Mario Südholt
- Contact: Adrien Lebre
- URL: http://beyondtheclouds.github.io/VMPlaceS/

5.5. btrCloud

KEYWORDS: Cloud computing - Virtualization - Grid - Energy - Orchestration - Autonomic system - Placement - Cluster - Data center - Scheduler
FUNCTIONAL DESCRIPTION

Orchestration, virtualization, energy, autonomic system, placement, cloud computing, cluster, data center, scheduler, grid

btrCloud is a virtual machine manager for clusters and provides a complete solution for the management and optimization of virtualized data centers. btrCloud (acronym of better cloud) is composed of three parts.

The analysis function enables operatives and people in charge to monitor and analyze how a data-center works - be it on a daily basis, on the long run, or in order to predict future trends. This feature includes boards for performance evaluation and analysis as well as trends estimation.

btrCloud, by the integration of btrScript, provides (semi-)automated VM lifecycle management, including provisioning, resource pool management, VM tracking, cost accounting, and scheduled deprovisioning. Key features include a thin client interface, template-based provisioning, approval workflows, and policy-based VM placement.

Finally, several kinds of optimizations are currently available, such as energy and load balancing. The former can help save up to around 20% of the data-center energy consumption. The latter provides optimized quality of service properties for applications that are hosted in the virtualized datacenters.

- Participants: Guillaume Le Louët, Frédéric Dumont and Jean-Marc Menaud
- Contact: Guillaume Le Louët
- URL: http://www.btrcloud.org/btrCloud/index_EN.html

6. New Results

6.1. Highlights of the year

Nicolas Tabareau has been awarded a starting grant from the European Research Council (ERC), the most prestigious type of research projects of the European Union for young researchers. From 2015–2020 he will pursue research on "CoqHoTT: Coq for Homotopy Type Theory."

In the domain of resource management notably for Cloud infrastructures, the team has produced several very visible results. These include contributions to popular and new simulation tools and platforms [17], [27], [28] as well as new techniques for the energy-efficient execution of Cloud applications [15].

On the topics of software composition and programming languages, the team has, among others, two remarkable results: a new notion of effect capabilities and corresponding monadic analysis techniques [14] as well as the first comprehensive survey of domain-specific aspect languages [13].

6.2. Programming Languages

Participants: Walid Benghrabit, Ronan-Alexandre Cherrueau, Rémi Douence, Hervé Grall, Florent Marchand de Kerchove de Denterghem, Jacques Noyé, Jean-Claude Royer, Mario Südholt.

6.2.1. Formal Methods, logics and type theory

This year we have proposed "Gradual Certified Programming" as a bridge between type-based expressive proofs and programming languages, have extended previous type theories by new homotopy-based means, and have introduced "effect capabilities" to control monad-based effects in Haskell.

6.2.1.1. Gradual Certified Programming in Coq

Expressive static typing disciplines are a powerful way to achieve high-quality software. However, the adoption cost of such techniques should not be under-estimated. Just like gradual typing allows for a smooth transition from dynamically-typed to statically-typed programs, it seems desirable to support a gradual path to certified programming. We have explored gradual certified programming in Coq [33], providing the possibility to postpone the proofs of selected properties, and to check "at runtime" whether the properties actually hold. Casts can be integrated with the implicit coercion mechanism of Coq to support implicit cast insertion à la gradual typing. Additionally, when extracting Coq functions to mainstream languages, our encoding of casts supports lifting assumed properties into runtime checks. Much to our surprise, it is not necessary to extend Coq in any way to support gradual certified programming. A simple mix of type classes and axioms makes it possible to bring gradual certified programming to Coq in a straightforward manner.

6.2.1.2. Homotopy Hypothesis in Type Theory

In classical homotopy theory, the homotopy hypothesis asserts that the fundamental omega-groupoid construction induces an equivalence between topological spaces and weak omega-groupoids. In the light of Voevodsky's univalent foundations program, which puts forward an interpretation of types as topological spaces, we have considered the question of transposing the homotopy hypothesis to type theory [16]. Indeed such a transposition could stand as a new approach to specifying higher inductive types. Since the formalization of general weak omega-groupoids in type theory is a difficult task, we have only taken a first step towards this goal, which consists in exploring a shortcut through strict omega-categories. The first outcome is a satisfactory type-theoretic notion of strict omega-category, which has hsets of cells in all dimensions. For this notion, defining the 'fundamental strict omega-category' of a type seems out of reach. The second outcome is an 'incoherently strict' notion of type-theoretic omega-category, which admits arbitrary types of cells in all dimensions. These are the 'wild' omega-categories of the title. They allow the definition of a 'fundamental wild omega-category' map, which leads to our (partial) homotopy hypothesis for type theory (stating an adjunction, not an equivalence). All of our results have been formalized in the Coq proof assistant. Our formalization makes systematic use of the machinery of coinductive types.

6.2.1.3. Effect Capabilities For Haskell

Computational effects complicate the tasks of reasoning about and maintaining software, due to the many kinds of interferences that can occur. While different proposals have been formulated to alleviate the fragility and burden of dealing with specific effects, such as state or exceptions, there is no prevalent robust mechanism that addresses the general interference issue. Building upon the idea of capability-based security, we have proposed effect capabilities [14] as an effective and flexible manner to control monadic effects and their interferences. Capabilities can be selectively shared between modules to establish secure effect-centric coordination. We have further refined capabilities with type-based permission lattices to allow fine-grained decomposition of authority. An implementation of effect capabilities in Haskell has been done, using type classes to establish a way to statically share capabilities between modules, as well as to check proper access permissions to effects at compile time.

6.2.1.4. Correct Refactoring Tools

Most integrated development environments provide refactoring tools. However, these tools are often unreliable. As a consequence, developers have to test their code after applying an automatic refactoring.

Refactoring tools for industrial languages are difficult to test and verify. We have developed a refactoring operation for C programs (renaming of global variables) for which we have proved that it preserves the set of possible behaviors of the transformed programs [39]. That proof of correctness relies on the operational semantics provided by CompCert C in Coq. We have also proved some properties of the transformation which are used to establish properties of a composed refactoring operations.

6.2.2. Language Mechanisms

This year we have contributed new results on domain-specific aspect languages, concurrent event-based programming, model transformations as well as the relationship between functional and constraint programming.

Furthermore, we have proposed language support for the definition and enforcement of security properties, in particular related to the accountability of service-based systems, see Sec. 6.3.

6.2.2.1. Domain-Specific Aspect Languages

Domain-Specific Aspect Languages (DSALs) are Domain-Specific Languages (DSLs) designed to express crosscutting concerns. Compared to DSLs, their aspectual nature greatly amplifies the language design space. In the context of the Associate Team RAPIDS/REAL, we have structured this space in order to shed light on and compare the different domain-specific approaches to deal with crosscutting concerns [13]. We have reported on a corpus of 36 DSALs covering the space, discussed a set of design considerations and provided a taxonomy of DSAL implementation approaches. This work serves as a frame of reference to DSAL and DSL researchers, enabling further advances in the field, and to developers as a guide for DSAL implementations.

6.2.2.2. Concurrent Event-Based Programming

The advanced concurrency abstractions provided by the Join calculus overcome the drawbacks of low-level concurrent programming techniques. However, with current approaches, the coordination logic involved in complex coordination schemas is still fragmented. In [11], Jurgen Van Ham presents JEScala, a language that captures coordination schemas in a more expressive and modular way by leveraging a seamless integration of an advanced event system with join abstractions. The implementation of joins-based state machines is discussed with alternative faster implementations made possible through a domain specific language. Event monitors are introduced as a way of synchronizing event handling and building concurrent event-based applications from sequential event-based parts.

6.2.2.3. Model Lazy Transformation

The Object Constraint Language (OCL) is a central component in modeling and transformation languages such as the Unified Modeling Language (UML), the Meta Object Facility (MOF), and Query View Transformation (QVT). OCL is standardized as a strict functional language. We have proposed a lazy evaluation strategy for OCL [36]. This lazy evaluation semantics is beneficial in some model-driven engineering scenarios for speeding up the evaluation times for very large models, simplifying expressions on models by using infinite data structures (e.g., infinite models) and increasing the reusability of OCL libraries. We have implemented the approach on the ATL virtual machine EMFTVM. This is a joint work with the Inria team Atlanmod.

6.2.2.4. Composition Mechanisms for Constraints Generalization

Structural time series (pattern for sequences of values) can be described with numerous automata-based constraints. In [12], we describe a large family of constraints for structural time series by means of function composition. We formalize the patterns using finite transducers. Based on that description, we automatically synthesize automata with accumulators, as well as constraint checkers. The description scheme not only unifies the structure of the existing 30 time-series constraints, but also leads to over 600 new constraints, with more than 100,000 lines of synthesized code. This is a joint work with the Inria team Tasc.

6.3. Software Composition

Participants: Walid Benghrabit, Ronan-Alexandre Cherrueau, Rémi Douence, Hervé Grall, Jean-Claude Royer, Mario Südholt.

6.3.1. Constructive Security

Nowadays we are witnessing the wide-spread use of cloud services. As a result, more and more end-users (individuals and businesses) are using these services for achieving their electronic transactions (shopping, administrative procedures, B2B transactions, etc.). In such scenarios, personal data is generally flowing between several entities and end-users need (i) to be aware of the management, processing, storage and retention of personal data, and (ii) to have necessary means to hold service providers accountable for the usage of their data. Usual preventive security mechanisms are not adequate in a world where personal data can be exchanged on-line between different parties and/or stored at multiple jurisdictions. Accountability becomes a necessary principle for the trustworthiness of open computer systems. It regards the responsibility and liability for the data handling performed by a computer system on behalf of an organization. In case of misconduct (e.g. security breaches, personal data leak, etc.), accountability should imply remediation and redress actions, as in the real life.

In 2015, we have contributed two main results: first, techniques for the logic-based definition, analysis and verification of accountability properties; second, a new framework for the compositional definition of privacy-properties and their type-based enforcement.

6.3.1.1. Logic-based accountability properties

We have proposed a framework for the representation of accountability policies [37]. This framework comes with two novel accountability policy languages; the Abstract Accountability Language (AAL), which is devoted to the representation of preferences/obligations in an human readable fashion, and a concrete one for the mapping to concrete enforceable policies. Our efforts have focused on a formal foundation for the AAL language and some applications.

We have also introduced an approach to assist the design of accountable applications [21]. In particular, we consider an application's abstract component design and we introduce a logical approach allowing various static verification. This approach offers effective means to early check the design and the behavior of an application and its offered/required services. We motivate our work with a realistic use case coming from the A4Cloud project and validate our proposal with experiments using the TSPASS theorem prover. This prover is competitive with other model-checkers and sat solvers and we gain a more abstract approach than with our previous experiment with a model-checker. It makes also easier the link with end users, for instance privacy officers.

To give a formal foundation of the AAL language we define a translation into first-order temporal logic [20]. We introduce a formula to interpret accountability and a natural criterion to achieve the accountability compliance of two clauses. We continue to apply it to an health care system taking into account data privacy features, data transfers and location processing. We demonstrate few heuristics to speed up the resolution time and to assist in conflict detection. Tool support (AccLab) has been provided to support editing, checking and proving AAL clauses.

6.3.1.2. Composition of Privacy-Enforcement Techniques

Today's large-scale computations, e.g., in the Cloud, are subject to a multitude of risks concerning the divulging and ownership of private data. Privacy risks are mainly addressed using a large variety of encryption-based techniques. We have proposed a compositional approach for the declarative and correct composition of privacy-preserving applications in the Cloud [22], [38]. Our approach provides language support for the compositional definition of encryption-based and fragmentation-based privacy-preserving algorithms. This language comes equipped with a set of laws that allows us to verify privacy properties. We have provided implementation support in Scala that ensures certain privacy properties by construction using advanced features of Scala's type system.

6.3.2. Modular systems

6.3.2.1. Modularity for Javascript Interpreters.

With an initial motivation based on the security of web applications written in JavaScript, we have provided new techniques for the instrumentation of an interpreter for a dynamic analysis as a crosscutting concern [31]. We have defined the instrumentation problem — an extension to the expression problem with a focus on modifying interpreters. We have then shown how we can instrument an interpreter for a simple language using only the bare language features provided by JavaScript.

6.4. Cloud applications and infrastructures

Participants: Frederico Alvares, Simon Dupont, Md Sabbir Hasan, Adrien Lebre, Thomas Ledoux, Jonathan Lejeune, Guillaume Le Louët, Jean-Marc Menaud, Jonathan Pastor, Mario Südholt.

In 2015, we have provided solutions for Cloud-based and distributed programming, virtual environments and data centers.

6.4.1. Cloud and distributed programming

6.4.1.1. Cloud elasticity

Cloud Computing has provided important new means for the capacity management of resources. The elasticity and the economy of scale are the intrinsic elements that differentiate it from traditional computing paradigm.

A good capacity planning method is a necessary factor but not sufficient to fully exploit Cloud elasticity. In [26], we propose innovative policies for resource management to achieve the optimal balance between capacity and quality of Cloud services The main idea is to finely control the scalability and the termination of virtual machines with respect to several criteria such as the lifecycle of the instances (*e.g.* initialization time) or their cost. The approach was evaluated on an Amazon EC2 cluster. Experimental results illustrate the soundness of the proposed approach and the impact of scalability/termination resource policies: a cost saving of as much as 30% can be achieved with a minimal number of violations, as small as 1%.

In order to improve Cloud elasticity, we advocate that the software layer can take part in the elasticity process as the overhead of software reconfiguration can be usually considered negligible compared to infrastructural costs. Thanks to this extra level of elasticity, we are able to define cloud reconfigurations that enact elasticity in both the software and infrastructure layers. In [23], we present an autonomic approach to manage cloud elasticity in a cross-layered manner. First, we enhance cloud elasticity with the software elasticity model. Then, we describe how our autonomic cloud elasticity model relies on the dynamic selection of elasticity tactics. We present an experimental analysis of a subset of those elasticity tactics under different scenarios in order to provide insights on strategies that could drive the autonomic selection of the proper tactics to be applied.

6.4.1.2. Service-level agreement for the Cloud

Quality-of-service and SLA guarantees are among the major challenges of cloud-based services. In [18], we first present a new cloud model called SLAaaS — SLA aware Service. SLAaaS considers QoS levels and SLA as first class citizens of cloud-based services. This model is orthogonal to other SaaS, PaaS, and IaaS cloud models, and may apply to any of them. More specifically, we make three contributions: (i) we provide a domain-specific language that allows to define SLA constraints in cloud services; (ii) we present a general control-theoretic approach for managing cloud service SLA; (iii) we apply our approach to MapReduce, locking, and e-commerce services.

6.4.1.3. Distributed multi-resource allocation

Generalized distributed mutual exclusion algorithms allow processes to concurrently access a set of shared resources. However, they must ensure an exclusive access to each resource. In order to avoid deadlocks, many of them are based on the strong assumption of a prior knowledge about conflicts between processes' requests. Some other approaches, which do not require such a knowledge, exploit broadcast mechanisms or a global lock, degrading message complexity and synchronization cost. We propose in [29] [41] a new solution for shared resources allocation which reduces the communication between non-conflicting processes without a prior knowledge of processes conflicts. Performance evaluation results show that our solution improves resource use rate by a factor up to 20 compared to a global lock based algorithm.

6.4.2. Virtualization and data centers

In 2015, we have produced results and tools for the simulation of large-scale distributed algorithms, notably VM scheduling algorithms, have contributed new abstractions for storage systems and have devised new means for the introspection of Cloud infrastructures.

6.4.2.1. SimGrid / VMPlaceS

We have developed VMPlaceS [28], a framework providing programming support for the definition of VM placement algorithms, execution support for their simulation at large scales, as well as new means for their trace-based analysis. VMPlaceS enables, in particular, the investigation of placement algorithms in the context of numerous and diverse real-world scenarios. To illustrate relevance of such a tool, we evaluated three different classes of virtualization environments: centralized, hierarchical and fully distributed placement algorithms. We showed that VMPlaceS facilitates the implementation and evaluation of variants of placement algorithms. The corresponding experiments have provided the first systematic results comparing these algorithms in environments including up to one thousand of nodes and ten thousands of VMs in most cases.

While such a number is already valuable and although we finalized the virtualization abstractions in SimGrid [17], we are in touch with the core developers in order to improve the code of VMPlaceS with the ultimate objective of addressing infrastructures up to 100K physical machines and 1 Millions virtual machines over a period of one day.

The current version of VMPlaceS is available on a public git repository :http://beyondtheclouds.github.io/VMPlaceS/.

6.4.2.2. Storage abstractions within the SimGrid framework

With the recent data deluge, storage is becoming the most important resource to master in modern computing infrastructures. Dimensioning and assessing the performance of storage systems are challenges for which simulation constitutes a sound approach. Unfortunately, only a few existing simulators of large scale distributed computing systems go beyond providing merely a notion of storage capacity. In 2015, we contributed to the SimGrid efforts toward the simulation of such systems [27]. Concretely, we characterized the performance behavior of several types of disks to derive a first model of storage resource. This model has been integrated within the SimGrid framework available under the LGPL license (http://simgrid.gforge.inria.fr).

6.4.2.3. Cloud Introspection

Cloud Computing has become a new technical and economic model for many IT companies. By virtualizing services, it allows for a more flexible management of datacenters capacities. However, its elasticity and its flexibility led to the explosion of virtual environments to manage. It's common for a system administrator to manage several hundreds or thousands virtual machines. Without appropriate tools, this administration task may be impossible to achieve.

We purpose in [32] a decision support tool to detect virtual machines with atypical behavior. Virtual machines whose behavior is different from other VMs running in the data center are tagged as atypicals. Our analysis tool is based on a specific partitioning algorithm which identifies VM behaviors. This tool has been validated in production environments and is used by several companies.

To collect finer metrics (for security, energy management etc.), VM introspection an agent can be installed in a VM to intrusively supervise it or the hypervisor can be used to non-intrusively recover the introspection metrics. In the case of intrusive introspection, the agent installed on the VM operating system will retrieve a set of information related to the operating system operation. However, the installation of an agent in the virtual machine increases the cost of deploying the virtual machine and its resource consumption. The Virtual Machine Introspection (VMI) at the hypervisor level (non intrusively) offer a complete, consistent and untainted view of the VM state. This solution allows an isolation of the VMI mechanism from the guest OS, while allowing monitoring and modifying any state of the VM.

We have also provided a comprehensive summary on VM introspection techniques [25]. Existing VMI techniques are analyzed with respect to their approach to closing the "semantic gap" between the (low level) information provided by the hypervisor and the input to the security analysis.

Finally, we have introduced an extension to LibVmi to detect and monitor a process resource consumption inside a VM from the hypervisor [34]. This extension monitor process cpu and ram ressources without probe. This extension can detect abusive cpu resource usage and atypical ram utilization. This fine monitoring system can be used in many context (security, power consumption, fault tolerance).

6.5. Green IT

Participants: Simon Dupont, Md Sabbir Hasan, Thomas Ledoux, Jonathan Lejeune, Guillaume Le Louët, Jean-Marc Menaud.

In 2015, we have provided new models and solutions for the energy-optimal execution of cloud applications in data centers.

6.5.1. Renewable energy

With the emergence of the Future Internet and the emergence of new IT models such as cloud computing, the usage of data centers (DC) and consequently their power consumption increase dramatically. Besides the ecological impact, the energy consumption is a predominant criteria for DC providers since it determines the daily cost of their infrastructure. As a consequence, power management becomes one of the main challenges for DC infrastructures and more generally for large-scale distributed systems.

6.5.1.1. Renewable energy for data centers

We have presented the EPOC project which focuses on optimizing the energy consumption of mono-site DCs connected to the regular electrical grid and to renewable energy sources [19]. A first challenge in this context consists in developing a (for users) transparent distributed system that enables energy-proportional computations from the system to service-oriented levels. The second challenge addresses the corresponding energy issues through collaborative measurements and energy-optimizing actions inside infrastructure-software stack, more precisely between applications and resource management systems. This approach must manage Service Level Agreement (SLA) constraints by striving for the best trade-off between energy cost (from the regular electric grid), its availability (from renewable energy sources), and service degradation (from application reconfiguration issues to job suspension ones). The third challenge embarks pursues energy efficient optical networks as key enablers of the future internet and cloud-networking service deployment through the convergence of optical infrastructure with the upper network layers.

The second challenge is more precisely describe in [30]. In this paper we present PIKA, a framework aiming at reducing the Brownian energy consumption (ie. from non renewable energy sources), and improving the usage of renewable energy for mono-site data centers. PIKA exploits jobs with slack periods, and executes and suspends them depending on the available renewable energy supply. By consolidating the virtual machines (VMs) on the physical servers, PIKA adjusts the number of powered-on servers in order for the overall energy consumption to match the renewable energy supply. Using simulations driven by real-world workloads and solar power traces, we demonstrate that PIKA consumes 41% less Brownian energy and increases 35.3% renewable energy integration ratio in comparison with the baseline algorithm from the literature.

6.5.1.2. Energy monotoring

We have designed SensorScript, a Business-Oriented Domain-Specific Language for Sensor Networks [24], [35]. In smart grids, or more generally the Internet of Things, many research work has been performed on the whole chain, from communication sensors to big data management, through communication middlewares. Few of this work have addressed the problem of gathered data access. In fact, being able, as a system administrator, to manipulate and gather data collected from a set of sensors in a simple and efficient way represents an essential need.

To address this issue, the solution we considered consists of a multi-context modeling for raw data, in the form of a multi-tree: a directed acyclic graph consisting of multiple intricate trees, each of them describing a hierarchy corresponding to a given use context. The objectives are to provide not only a means to rationalize users needs before writing queries, but also to offer a domain-specific language (DSL) which takes advantage of the multi-tree modeling to simplify the experience of pre-identified users that query data.

6.5.1.3. Green SLA and virtualization of green energy

The demand for energy-efficient services is increasing considerably as people are getting more environmentally-conscious in order to build a sustainable society. The main challenge for Cloud providers is to manage Green SLA (Service Level Agreement) constraints for their customers while satisfying their business objectives, such as maximizing profits by lowering expenditure for so-called green (renewable) energy. Since, Green SLA needs to be proposed based on the presence of green energy, the intermittent nature of renewable sources makes it difficult to be achieved. In response, we propose a scheme for green energy management based on three contributions [15]: i) we introduce the concept of virtualization of green energy to address the uncertainty of green energy availability, ii) we extend the Cloud Service Level Agreement (CSLA) language to support Green SLA by introducing two new threshold parameters and iii) we introduce algorithms for Green SLA which leverage the concept of virtualization of green energy to provide interval-specific Green SLA. We have conducted experiments with real workload profiles from PlanetLab and server power model from SPECpower to demonstrate that Green SLA can be successfully established and satisfied without incurring higher cost.

7. Bilateral Contracts and Grants with Industry

7.1. Cooperation with SIGMA group

Participants: Thomas Ledoux [correspondent], Simon Dupont.

In 2012, we have started a cooperation with Sigma Group (http://www.sigma.fr), a software editor and consulting enterprise. The cooperation consists in a joint (a so-called Cifre) PhD on eco-elasticity of software for the Cloud and the sponsorship of several engineering students at the MSc-level.

As a direct consequence of the increasing popularity of Cloud computing solutions, data centers are rapidly growing in number and size and have to urgently face with energy consumption issues. The aim of Simon Dupont's PhD, started in November 2012, is to explore the *software elasticity* capability in Software-as-a-Service (SaaS) development to promote the management of SaaS applications that are more flexible, more reactive to environment changes and therefore self-adaptive for a wider range of contexts. As a result, SaaS applications become more elastic and by transitivity more susceptible to energy constraints and optimization issues.

In 2015, we have presented an autonomic approach to manage cloud elasticity that obey cross-layer constraints [23].

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. Competitiveness cluster Images-et-Reseaux

8.1.1.1. EcoCloud

Participant: Jean-Marc Menaud.

The project EcoCloud is a cooperative research project running for 2 years. Three other partners collaborate within the project that is coordinated by the company EasyVirt: the Ascola team and another company Pentasonic. The partners aim at developing an economically-valid and ecologic cloud platform in the context of micro and mono-site data centers (all resources are in the same physical location). A high SLA level must be provided with a specific focus on high availability satisfying strong redundancy and placement constraints.

8.2. National Initiatives

8.2.1. CominLabs laboratory of excellence

8.2.1.1. EPOC

Participants: Jean-Marc Menaud [coordinator], Thomas Ledoux, Md Sabbir Hasan, Yunbo Li.

The project EPOC (Energy Proportional and Opportunistic Computing system) is an (academic) Labex CominLabs project running for 4 years. Four other partners collaborate within the project that is coordinated by ASCOLA: Myriads team, and the three institutions ENIB, ENSTB and University of Nantes. In this project, the partners focus on energy-aware task execution from the hardware to application's components in the context of a *mono-site* data center (all resources are in the same physical location) which is connected to the *regular electric Grid and to renewable energy sources* (such as windmills or solar cells). Three major challenges are addressed in this context: Optimize the energy consumption of distributed infrastructures and service compositions in the presence of ever more dynamic service applications and ever more stringent availability requirements for services; Design a clever cloud's resource management which takes advantage of renewable energy availability to perform opportunistic tasks, then exploring the trade-off between energy saving and performance aspects in large-scale distributed system; Investigate energy-aware optical ultra high-speed interconnection networks to exchange large volumes of data (VM memory and storage) over very short periods of time.

One of the strengths of the project is to provide a systematic approach, and use a single model for the system (from hard to soft) by mixing constraint programming and behavioral models to manage energy consumption in data centers.

8.2.1.2. SecCloud

Participants: Jacques Noyé [coordinator], Florent Marchand de Kerchove de Denterghem, Mario Südholt.

The high-level objective of the 3-year SecCloud (Secure Scripting for the Cloud) project is to enhance the security of devices on which web applications can be downloaded, i.e. to enhance client-side security in the context of the Cloud. In order to do so, the project relies on a language-based approach, focusing on three related issues:

- The definition of security policies for web architectures, especially on the client-side.
- Formally-proven analyses of web programming languages.
- Multi-level enforcement mechanisms for the security policies (based on static and dynamic analysis encompassing application-level and system-level software).

ASCOLA members are mainly interested in JavaScript as a programming language as well as the use of aspects as a seamless path from the definition of security policies and their composition to their implementation.

This year we have investigated how to extend real-world Javascript environments, such as Narcissus in a modular way.

8.2.2. ANR

8.2.2.1. SONGS (ANR/INFRA)

Participants: Adrien Lebre [coordinator], Jonathan Pastor, Anthony Simonet.

The SONGS project (Simulation of Next Generation Systems) is an ANR/INFRA project running for 48 months (starting in January 2012 with an allocated budget of 1.8MEuro, 95KEuro for ASCOLA).

The consortium is composed of 11 academic partners from Nancy (AlGorille, coordinator), Grenoble (MESCAL), Villeurbanne (IN2P3 Computing Center, GRAAL/Avalon - LIP), Bordeaux (CEPAGE, HiePACS, RUNTIME), Strasbourg (ICPS - LSIIT), Nantes (ASCOLA), Nice (MASCOTTE, MODALIS).

The goal of the SONGS project (http://infra-songs.gforge.inria.fr) is to extend the applicability of the SimGrid simulation framework from Grids and Peer-to-Peer systems to Clouds and High Performance Computation systems.

8.2.3. FSN

8.2.3.1. OpenCloudware (FSN)

Participants: Jean-Marc Menaud [coordinator], Thomas Ledoux.

The OpenCloudware project is coordinated by France Telecom, funded by the French Fonds National pour la Société Numérique (FSN, call Cloud n°1) and endorsed by competitiveness clusters Minalogic, Systematic and SCS. OpenCloudware is developed by a consortium of 18 partners bringing together industry and academic leaders, innovative technology start-ups and open source community expertise. The project started in 2012 for a duration of 42 months.

The OpenCloudware project aims at building an open software engineering platform, for the collaborative development of distributed applications to be deployed on multiple Cloud infrastructures. It will be available through a self-service portal. We target virtualized multi-tier applications such as JavaEE - OSGi. The results of OpenCloudware will contain a set of software components to manage the lifecycle of such applications, from modelling(Think), developing and building images (Build), to a multi-IaaS compliant PaaS platform (Run).

The ASCOLA project-team is mainly involved in the sub-projects "Think" (SLA model across Cloud layers) and "Run" (virtual machine manager for datacenters and placement constraints). The team has developed btrCloudStack, a private cloud based on the OpenSource CloudStack and integrating the work on placement rules and energy optimization. This software system has been extended this year.

8.2.3.2. Hosanna (FSN)

Participants: Jean-Marc Menaud [coordinator], Rémy Pottier.

The Hosanna project (aims to scientifically and technically addresses the problem of deploying applications on a distributed multi-cloud virtual infrastructure (private cloud, Amazon, OVH, CloudWatt, Numergy etc.). This recent need is an important topic issue highlighted by recent major Outages in 2013 by the biggest players in the cloud such as Amazon or Netflix. This project aims to provide services that allow users to deploy their cloud multi-tier applications on hybrid Clouds infrastructures without any separation between IaaS. The Ascola team is extending its optimization solution to address the task placement problem in a multi-cloud environment and will develop a case study on a secure distributed file system. The project started in 2015 for a duration of 2 years.

8.2.4. CPER

8.2.4.1. SeDuCe

Participants: Jean-Marc Menaud [coordinator], Adrien Lebre.

The SeDuCe project (Sustainable Data Centers: Bring Sun, Wind and Cloud Back Together), aims to design an experimental infrastructure dedicated to the study of data centers with low energy footprint. This innovative data center will be the first experimental data center in the world for studying the energy impact of cloud computing and the contribution of renewable energy (solar panels, wind turbines) as well on the scientific, technological, that economical. This project is integrated in the national context of grid computing (Grid'5000), and the Constellation project, which will be an inter-node (Pays de la Loire, Brittany). He also participated in the validation of scientific work in interdisciplinary axis STIC and energy efficiency of the laboratory of excellence COMIN Labs.

8.2.5. Inria Project Labs

8.2.5.1. DISCOVERY

Participants: Adrien Lebre [coordinator], Mario Südholt.

To accommodate the ever-increasing demand for Utility Computing (UC) resources, while taking into account both energy and economical issues, the current trend consists in building larger and larger Data Centers in a few strategic locations. Although such an approach enables UC providers to cope with the actual demand while continuing to operate UC resources through centralized software system, it is far from delivering sustainable and efficient UC infrastructures for future needs.

The DISCOVERY initiative [40] aims at exploring a new way of operating Utility Computing (UC) resources by leveraging any facilities available through the Internet in order to deliver widely distributed platforms that can better match the geographical dispersal of users as well as the ever increasing demand. Critical to the emergence of such locality-based UC (LUC) platforms is the availability of appropriate operating mechanisms. The main objective of DISCOVERY is to design, implement, demonstrate and promote the LUC Operating System (OS), a unified system in charge of turning a complex, extremely large-scale and widely distributed infrastructure into a collection of abstracted computing resources which is efficient, reliable, secure and at the same time friendly to operate and use.

The consortium is composed of experts in the following research areas: large-scale infrastructure management systems, networking and P2P algorithms. Moreover, two key network operators, namely Orange and RENATER, are involved in the project.

By deploying and using a LUC Operating System on backbones, our ultimate vision is to enable large parts of the Internet to be hosted and operated by its internal structure itself: a scalable set of resources delivered by any computing facilities forming the Internet, starting from the larger hubs operated by ISPs, governments and academic institutions, to any idle resources that may be provided by end users.

ASCOLA leads the DISCOVERY IPL and contributes mainly around two axes: VM life cycle management and security concerns.

8.3. European Initiatives

8.3.1. FP7 & H2020 Projects

8.3.1.1. ERC Starting Grant: The CoqHoTT project

Participant: Nicolas Tabareau [coordinator].

CoqHoTT stands for Coq for Homotopy Type Theory. The goal of this project is to go further in the correspondence between proofs and programs which has allowed in the last 20 years the development of useful proof assistants, such as Coq (developed by Inria). This project starts from the recent discovery by field medal Vladimir Voevosdky, of the strong link between homotopy theory (which studies the notion of continuous deformation in topology) and type theory (which is at the heart of the Coq proof assistant). The main goal of the CoqHoTT project is to provide a new generation of proof assistants based on this fascinating connection.

The CoqHoTT project has started in June 2015 with a budget of 1,5M€.

8.3.1.2. A4Cloud (IP)

Participants: Mario Südholt [coordinator], Walid Benghabrit, Ronan-Alexandre Cherrueau, Rémi Douence, Hervé Grall, Jean-Claude Royer.

The integrated project “Accountability for the Cloud” (A4Cloud) is coordinated by HP Labs, UK, and fosters cooperation of a consortium of five industrial and eight academic partners. It has been started in Oct. 2012 for a duration of 42 months.

A4Cloud focuses on accountability properties for the cloud and other future internet services as the most critical prerequisite for effective governance and control of corporate and private data processed by cloud-based IT services. The research being conducted in the project will increase trust in cloud computing by devising methods and tools, through which cloud stakeholders can be made accountable for the privacy and confidentiality of information held in the cloud. These methods and tools will combine risk analysis, policy enforcement, monitoring and compliance auditing. They will contribute to the governance of cloud activities, providing transparency and assisting legal, regulatory and socio-economic policy enforcement. For further information, see http://www.a4cloud.eu. ASCOLA, whose financial support consists of 550 K€, is mainly involved in the sub-projects on the enforcement of accountability and security policies, as well as tool validation efforts.

This year we have proposed new logic-based and language-level means for the formal specification and implementation of accountability properties and have proposed a new composition approach for distributed systems that enforces privacy-properties through statically-verified types (see 6.3).

8.3.1.3. BigStorage (MSCA-ETN)

Participants: Adrien Lebre [coordinator], Linh-Thuy Nguyen, Mario Südholt.

BigStorage is a European Training Network (ETN) whose main goal is to train future data scientists in order to enable them and us to apply holistic and interdisciplinary approaches for taking advantage of a data-overwhelmed world, which requires HPC and Cloud infrastructures with a redefinition of storage architectures underpinning them – focusing on meeting highly ambitious performance and energy usage objectives.

Nowadays there is a lack of professionals who know how to deal with storage, management and analysis of Big Data. Indeed, there is a gap between infrastructures for dealing with Big Data and applications using these volumes of data. In 2011, the McKinsey Global Institute published a study that found that, by 2018, there could be a shortage of up to 190,000 data scientists in the United States, representing a 50 percent to 60 percent gap between supply and demand. Similarly, European officials estimate that 300,000 data scientists will be needed in Europe in the forthcoming years. Other reports, such as those from PRACE and ETP4HPC, have also emphasized the need of skills in HPC, Cloud, Storage, Energy, or Big Data to maintain Europe's economy. In this context, a major goal of this project is to bring a substantial contribution to the training process of these future experts.

Within this project, ASCOLA leads the WP 3 that deals with the convergence between HPC and Cloud storage backends.

8.3.1.4. GRACeFUl (FETPROACT)

Participant: Rémi Douence [coordinator].

The GRACeFUl project is coordinated by Universitat Politecnica de Catalunya, Spain, and fosters cooperation of a consortium of two institutes and five academic partners. It has been started in Feb. 2015 for a duration of 36 months. For information, see https://www.graceful-project.eu.

Global Systems Science is a FET Proactive initiative under Horizon 2020 that seeks to improve the way scientific knowledge can help inform and evaluate policy and societal responses to global challenges like climate change and global financial crises.

The GRACeFUL project strives for a base for domain-specific languages aimed at building scalable rapid assessment tools for collective policy making in global systems. It involves several different disciplines. ASCOLA is involved in WP5 in order to provide expertise in functional programming . In this context Rémi Douence codirects the PhD thesis of Ekaterina Arafailova. This work has already produced a generalization of automata-based constraints [12].

8.4. International Initiatives

8.4.1. Inria Associate Teams

8.4.1.1. REAL

Title: Reasoning about Effects in Aspect Languages

International Partner (Institution - Laboratory - Researcher):

Universidad de Chile (CHILI)

Duration: 2010 - 2015

See also: http://real.gforge.inria.fr

During the period 2013-2015, REAL has studied means to reason about aspect interference, providing foundations for secure aspects, and the link of secure aspects with security aspects. This last year has been devoted to developing a general mechanism for modular composition of session types, applying our ideas on modular instrumentation of interpreters to Narcissus and information flow analyses, and exploring ideas for future collaboration, in particular in the area of gradual certification.

8.4.2. Inria International Partners

8.4.2.1. Informal International Partners

Apart from the Inria associate team rapids with the Pleiad group (Prof. Éric Tanter) at U. Chile, the Ascola team has formalized cooperations, notably in the context of co-financed and co-supervised PhD theses with the PROG group (Prof. Wofgang de Meuter) at VU Brussel, Belgium, and the Software Technology group (Prof. Mira Mezini) at TU Darmstadt, Germany.

Furthermore, the Ascola team has long-term cooperations that resulted in common results in 2015, typically joint publications or common software artifacts, with partners from the AIST research institute (Dr. Takahiro Hirofuchi) and U. of Bogota, Colombia (Prof. Rubby Casallas).

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. General chair, scientific chair

- A. Lebre co-organized cloudDays@Paris. Supported by the GDR RSD consortium, the DISCOVERY Inria Project Labs, the LIP6 labs and Telecom Sud Paris, this national event gathered 80 researchers and phd students to discuss about latest results in Virtualization and Cloud Computing [1] from the network and distributed system communities.
- A. Lebre was general chair of the ACM VTDC [2] 2015 workshop collocated with HPDC (30 participants).

9.1.1.2. Member of the organizing committee

- T. Ledoux is member of the board of the Green Lab Center association. This association promotes and disseminates Green IT practices and research prototypes to the world of education, research and companies [3].
- M. Südholt has been a member of the steering committee of the international conference Modularity since 2011.
- J.-M. Menaud member of GIS (Groupement d'intérêt scientifique) PERLE - Pôle d'Excellence de la Recherche Ligérienne en Energie.
- J.-M. Menaud member of GIS (Groupement d'intérêt scientifique) SyMeTRIC.
- J.-M. Menaud is the animator of "Pôle Science du Logiciel et des Systèmes Distribués" in Laboratoire des Sciences du Numérique à Nantes (LS2N) since June 2015.
- A. Lebre is member of the executive committee of the GDR CNRS RSD (Reseau et Système distribué). He is also co-leading the transversal action Virtualization and Clouds of this GDR.
- A. Lebre is member of the executive and architect committees of the Grid'5000 GIS (Groupement d'intérêt scientifique).
- N. Tabareau has co-organized with Peter LeFAnu Lumsdaine the HoTT-UF workshop, satellite to the RDP conferences.

[1] CloudDays@Paris, 2015.
[2] http://people.rennes.inria.fr/Adrien.Lebre/VTDC/vtdc15.html
[3] Green Lab Center

9.1.2. Scientific events selection

9.1.2.1. Member of the conference program committee

- A. Lebre was member of the program committees of ACM/IEEE CCGRID 2015, IEEE IC2E 2015, SCRAMBL 2015, EuroPar 2015, IEEE CloudCom 2015, and DSDIS 2015.
- T. Ledoux was member of the program committees of the following conferences: Int. Conf. on Eco-friendly Computing and Communication Systems (ICECCS'15), Int. Workshop on Green and Sustainable Software (GREENS'15) @ ICSE, Workshop on Adaptive and Reflective Middleware (ARM'15) @ Middleware, ACM Symposium On Applied Computing (SAC'15)- track Software Engineering Aspects of Green Computing.
- J.-M. Menaud has served on the program committee of SMARTGREENS 2015, ENERGY 2015, VHPC'15, CFSE-10, CLOUD COMPUTING 2015, 11th AICT 2015, 4th E2DC 2015, GreenCom2015.
- J.-C. Royer was a member of the program committes of WETICE 2015, CAL 2015, and ICIS 2015.

9.1.3. Journal

9.1.3.1. Member of the editorial board

- M. Südholt has been a member of the editorial board of the Springer journal "Transactions of Aspect-Oriented Software Development" (TAOSD) since 2010 and has been appointed co-editor-in-chief since April 2015.
- A. Lebre is associate editor for the IEEE Transactions on Big Data journal.

9.1.3.2. Reviewer

- A. Lebre has been a reviewer for the Software: Practice and Experience and IEEE TPDS Journals.
- T. Ledoux has been a reviewer for the Journal of Parallel and Distributed Computing (JPDC)-Elsevier (2015).
- J.-M. Menaud has been a reviewer for IEEE Transactions on Parallel and Distributed Systems (TPDS).
- M. Südholt has been a reviewer for IEEE "Transactions in Software Engineering" (TSE) and the Springer journal "Transactions on Aspect-Oriented Software Development" (TOASD).

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

The team is involved in the following undergraduate and graduate-level programs at Mines Nantes and University of Nantes (the institutions all of eaching staff belongs to):

- The team is a main contributor to the **engineering program of EMN**.
- Within this engineering program, the team is steering, chairing and the main contributor to a two-year **graduate-level informatics specialization**. H. Grall is managing this program.
- Since 2011 our team has defined and set up a new three-year **engineering program on software engineering**. T. Ledoux is managing this program.

The team has also been involved in the following MSc programs that have been carried out with partners from French and foreign universities:

- The team participates in the **MSc program "Alma"** on software architecture and distributed systems, a joint program steered by colleagues from University of Nantes. In this context, we are responsible for a 48-hour module on advanced software composition and take part in the program's governing board. M. Südholt is managing the participation of Mines Nantes in this program.
- Members of the team have taught different **courses at different study levels in Rennes** mainly organized by University of Rennes and the research institutes IRISA and Inria.

m members have taught for about 220 hours on average in 2015 (hours of presence in front of students). Hereby, we have taken into account that researchers and some professors have not taught at times. In addition, another significant part of the program is taught by temporary staff, whose participation is managed by ASCOLA members.

In addition, J. Noyé has been interim head of the DAPI department of École de Mines de Nantes until Oct. 2015 and has been its vice-head since then.

9.2.2. Supervision

The team has been supervising 18 PhD thesis (at least partially) in 2015, of which four have been co-supervised with external academic partners (one with partners from TU Darmstadt, Germany, and VU Brussel, Belgium) and two with another Inria team (Myriads from Rennes). Four PhDs have been co-supervised with industrial partners (Orange group, Sigma group, and the EasyVirt SME).

One PhD thesis has been defended this year: Jurgen Van Ham has presented a model and corresponding extension of Scala integrating event-based, aspect-oriented and object-oriented programming features. Three PhD students are preparing her defense for end of the first quarter 2015.

Rémi Douence has defended his habilitation (HDR) on the topic: “Composition non modulaire modulaire” presenting new means for the modularization of composition problems.

9.2.3. Juries

- J.-C. Royer was a member of the HDR committee of Mohamed Graiet (Université Paris Dauphine), July 2015.
- A. Lebre was a member of the PhD committee of Houssem Medhioub, “Architectures et mécanismes de fédération dans les environnements Cloud Computing et Cloud Networking”, University Pierre et Marie Curie/TelecomSud Paris, April 2015.
- J. Noyé was a member of the PhD committee of Dries Harnie: "Blame Prediction: Early detection of type errors in dynamically typed programming languages", Vrije Universiteit Brussel, Aug. 2015.
- J.-M. Menaud was a reviewer of the PhD of : Christina Herzog (Nov. 26, 2015) "Contributions à la modélisation avec un Système Multi Agent du transfert technologique en Green IT", Reviewer, Toulouse. Ibrahim Safieddine (Oct. 29, 2015) "Optimisation d’Infrastructures de Cloud Computing dans des Green Datacenters", Reviewer, Grenoble. Ge LI (Jull. 22, 2015) "Contrôle des applications fondé sur la qualité de service pour les plateformes logicielles dématérialisées (Cloud)", Reviewer, Annecy. Ahmed El Rheddane (Fev. 25, 2015) "Elasticité dans le Cloud Computing", Member, Grenoble.
- J.-M. Menaud was a reviewer of the **HdR** committee of Patricia Stolf (Nov. 13, 2015) "La gestion des ressources pour des infrastructures vertes par la reconfiguration", Reviewer, Toulouse.

10. Bibliography

Major publications by the team in recent years

[1] B. DE FRAINE, E. ERNST, M. SÜDHOLT. *Essential AOP: The A Calculus*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", December 2012, http://hal.inria.fr/hal-00676082

[2] I. FIGUEROA, T. SCHRIJVERS, N. TABAREAU, É. TANTER. *Compositional Reasoning About Aspect Interference*, in "Modularity’14: 13th International Conference on Modularity", Lugano, Switzerland, April 2014, https://hal.inria.fr/hal-00919935

[3] M. S. HASAN, Y. KOUKI, T. LEDOUX, J.-L. PAZAT. *Exploiting Renewable sources: when Green SLA becomes a possible reality in Cloud computing*, in "IEEE Transactions on Cloud Computing", July 2015, vol. PP, n^o 99, 1 p. [*DOI :* 10.1109/TCC.2015.2459710], https://hal.archives-ouvertes.fr/hal-01187907

[4] G. JABER, N. TABAREAU, M. SOZEAU. *Extending Type Theory with Forcing*, in "LICS'12 : Logic In Computer Science", Dubrovnik, Croatia, June 2012, http://hal.inria.fr/hal-00685150

[5] Y. KOUKI, F. ALVARES DE OLIVEIRA JR., S. DUPONT, T. LEDOUX. *A Language Support for Cloud Elasticity Management*, in "CCGrid'14: IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing", Chicago, United States, May 2014, pp. 1-8, https://hal.archives-ouvertes.fr/hal-00941945

[6] A. LEBRE, J. PASTOR, M. SÜDHOLT. *VMPlaceS: A Generic Tool to Investigate and Compare VM Placement Algorithms*, in "Europar 2015", Vienne, Austria, August 2015, https://hal.inria.fr/hal-01159033

[7] F. QUESNEL, A. LEBRE, M. SÜDHOLT. *Cooperative and Reactive Scheduling in Large-Scale Virtualized Platforms with DVMS*, in "Concurrency and Computation: Practice and Experience", December 2012, http://hal.inria.fr/hal-00675315

[8] M. SOZEAU, N. TABAREAU. *Universe Polymorphism in Coq*, in "ITP'14: Interactive Theorem Proving", Vienna, Austria, July 2014, https://hal.inria.fr/hal-00974721

[9] N. TABAREAU, M. SÜDHOLT, É. TANTER. *Aspectual Session Types*, in "Modularity'14 - 13th International Conference on Modularity", Lugano, Switzerland, April 2014, https://hal.inria.fr/hal-00872791

[10] J. M. VAN HAM, G. SALVANESCHI, M. MEZINI, J. NOYÉ. *JEScala: Modular Coordination with Declarative Events and Joins*, in "Modularity'14 - 13th International Conference on Modularity", Lugano, Switzerland, E. ERNST (editor), April 2014, https://hal.inria.fr/hal-00862332

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] J. M. VAN HAM. *Seamless concurrent programming of objects, aspects and events*, Ecole des Mines de Nantes, March 2015, https://tel.archives-ouvertes.fr/tel-01238752

Articles in International Peer-Reviewed Journals

[12] N. BELDICEANU, M. CARLSSON, R. DOUENCE, H. SIMONIS. *Using finite transducers for describing and synthesising structural time-series constraints*, in "Constraints", August 2015, 19 p. [*DOI :* 10.1007/S10601-015-9200-3], https://hal.inria.fr/hal-01186662

[13] J. FABRY, T. DINKELAKER, J. NOYÉ, É. TANTER. *A Taxonomy of Domain-Specific Aspect Languages*, in "ACM Computing Surveys", February 2015, vol. 47, n^o 3, 44 p. [*DOI :* 10.1145/2685028], https://hal.inria.fr/hal-01085063

[14] I. FIGUEROA, N. TABAREAU, É. TANTER. *Effect capabilities for Haskell: Taming effect interference in monadic programming*, in "Science of Computer Programming", November 2015 [*DOI :* 10.1016/J.SCICO.2015.11.010], https://hal.inria.fr/hal-01236145

[15] M. S. HASAN, Y. KOUKI, T. LEDOUX, J.-L. PAZAT. *Exploiting Renewable sources: when Green SLA becomes a possible reality in Cloud computing*, in "IEEE Transactions on Cloud Computing", July 2015, vol. PP, n^{O} 99, 1 p. [*DOI :* 10.1109/TCC.2015.2459710], https://hal.archives-ouvertes.fr/hal-01187907

[16] A. HIRSCHOWITZ, T. HIRSCHOWITZ, N. TABAREAU. *Wild omega-Categories for the Homotopy Hypothesis in Type Theory*, in "Leibniz International Proceedings in Informatics (LIPIcs)", 2015, vol. 38, pp. 226-240 [*DOI :* 10.4230/LIPICS.TLCA.2015.226], https://hal.archives-ouvertes.fr/hal-01178301

[17] L. POUILLOUX, T. HIROFUCHI, A. LEBRE. *SimGrid VM: Virtual Machine Support for a Simulation Framework of Distributed Systems*, in "IEEE Transactions on Cloud Computing", January 2016, forthcoming, https://hal.inria.fr/hal-01197274

[18] D. SERRANO, S. BOUCHENAK, Y. KOUKI, F. J. ALVARES DE OLIVEIRA, T. LEDOUX, J. LEJEUNE, J. SOPENA, L. ARANTES, P. SENS. *SLA guarantees for cloud services*, in "Future Generation Computer Systems", April 2015, 1 p. [*DOI :* 10.1016/J.FUTURE.2015.03.018], https://hal.archives-ouvertes.fr/hal-01162654

International Conferences with Proceedings

[19] N. BELDICEANU, B. DUMAS FERIS, P. GRAVEY, M. S. HASAN, C. JARD, T. LEDOUX, Y. LI, D. LIME, G. MADI-WAMBA, J.-M. MENAUD, P. MOREL, M. MORVAN, M.-L. MOULINARD, A.-C. ORGERIE, J.-L. PAZAT, O. ROUX, A. SHARAIHA. *The EPOC project: Energy Proportional and Opportunistic Computing system*, in "International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)", Lisbonne, Portugal, May 2015, https://hal.archives-ouvertes.fr/hal-01131602

[20] W. BENGHABRIT, H. GRALL, J.-C. ROYER, M. SELLAMI. *Abstract Accountability Language: Translation, Compliance and Application*, in "Asia-Pacific Software Engineering Conference", New Delhi, India, December 2015, https://hal.archives-ouvertes.fr/hal-01214365

[21] W. BENGHABRIT, H. GRALL, J.-C. ROYER, M. SELLAMI. *Checking Accountability with a Prover*, in "39th IEEE Annual Computer Software and Applications Conference (COMPSAC)", Taichung, Taiwan, 39th IEEE Annual Computer Software and Applications Conference (COMPSAC), July 2015, pp. 83-88 [*DOI :* 10.1109/COMPSAC.2015.8], https://hal.archives-ouvertes.fr/hal-01214360

[22] R.-A. CHERRUEAU, R. DOUENCE, M. SÜDHOLT. *A Language for the Composition of Privacy-Enforcement Techniques*, in "IEEE RATSP 2015, The 2015 IEEE International Symposium on Recent Advances of Trust, Security and Privacy in Computing and Communications", Helsinki, Finland, August 2015, pp. 1037 - 1044 [*DOI :* 10.1109/TRUSTCOM.2015.480], https://hal.inria.fr/hal-01168303

[23] S. DUPONT, J. LEJEUNE, F. ALVARES, T. LEDOUX. *Experimental Analysis on Autonomic Strategies for Cloud Elasticity*, in "2015 IEEE International Conference on Cloud and Autonomic Computing (ICCAC)", Cambridge, United States, September 2015, https://hal.inria.fr/hal-01178419

[24] A. GARNIER, R. POTTIER, J.-M. MENAUD. *SensorScript: a Domain-Specific Language for Sensor Networks*, in "International Conference on Future Internet of Things and Cloud (FiCloud-2015)", Rome, Italy, August 2015, https://hal.inria.fr/hal-01158259

[25] Y. HEBBAL, L. SYLVIE, J.-M. MENAUD. *Virtual Machine Introspection: Techniques and Applications*, in "International Conference on Availability, Reliability and Security", Toulouse, France, August 2015, https://hal.inria.fr/hal-01165285

[26] Y. KOUKI, M. S. HASAN, T. LEDOUX. *Delta Scaling: How Resources Scalability/Termination Can Be Taken Place Economically?*, in "11th World Congress on Services (SERVICES)", New York, United States, IEEE World Congress on Services (SERVICES) 2015, IEEE, June 2015, pp. 55-62 [*DOI :* 10.1109/SERVICES.2015.17], https://hal.archives-ouvertes.fr/hal-01162745

[27] A. LEBRE, A. LEGRAND, F. SUTER, P. VEYRE. *Adding Storage Simulation Capacities to the SimGrid Toolkit: Concepts, Models, and API*, in "CCGrid 2015 - Proceedings of the 15th IEEE/ACM Symposium on Cluster, Cloud and Grid Computing", Shenzhen, China, IEEE/ACM, May 2015, pp. 251-260 [*DOI :* 10.1109/CCGRID.2015.134], https://hal.inria.fr/hal-01197128

[28] A. LEBRE, J. PASTOR, M. SÜDHOLT. *VMPlaceS: A Generic Tool to Investigate and Compare VM Placement Algorithms*, in "Europar 2015", Vienne, Austria, August 2015, https://hal.inria.fr/hal-01159033

[29] J. LEJEUNE, L. ARANTES, J. SOPENA, P. SENS. *Reducing synchronization cost in distributed multi-resource allocation problem*, in "44th International Conference on Parallel Processing", Beijing, China, 44th International Conference on Parallel Processing, September 2015, https://hal.archives-ouvertes.fr/hal-01162329

[30] Y. LI, A.-C. ORGERIE, J.-M. MENAUD. *Opportunistic Scheduling in Clouds Partially Powered by Green Energy*, in "IEEE International Conference on Green Computing and Communications (GreenCom)", Sydney, Australia, December 2015, https://hal.archives-ouvertes.fr/hal-01205911

[31] F. MARCHAND DE KERCHOVE, J. NOYÉ, M. SÜDHOLT. *Towards Modular Instrumentation of Interpreters in JavaScript*, in "FOAL - Foundations of Aspect-Oriented Languages", Fort Collins, United States, MODULARITY Companion 2015, March 2015 [*DOI :* 10.1145/2735386.2736753], https://hal.archives-ouvertes.fr/hal-01116662

[32] J.-M. MENAUD, F. DUMONT. *Synthesizing Realistic Cloud Workload Traces for Studying Dynamic Ressource System Management*, in "International Conference on Cloud Computing and Big Data", Huangshan, China, June 2015, https://hal.inria.fr/hal-01132126

[33] É. TANTER, N. TABAREAU. *Gradual Certified Programming in Coq*, in "11th ACM Dynamic Languages Symposium (DLS 2015)", Pittsburgh, United States, October 2015 [*DOI :* 10.1145/2816707.2816710], https://hal.inria.fr/hal-01238774

National Conferences with Proceedings

[34] F. DUMONT, J.-M. MENAUD, L. SYLVIE. *Monitoring de l'utilisation mémoire et CPU par processus, basé sur l'introspection de machines virtuelles*, in "Conférence d'informatique en Parallélisme, Architecture et Système", Lille, France, July 2015, https://hal.inria.fr/hal-01154241

[35] A. GARNIER, J.-M. MENAUD, R. POTTIER. *SensorScript : un langage de requête dédié, orienté métiers, pour les réseaux de capteurs*, in "Conférence d'informatique en Parallélisme, Architecture et Système", Lille, France, July 2015, https://hal.inria.fr/hal-01154240

Conferences without Proceedings

[36] M. TISI, R. DOUENCE, D. WAGELAAR. *Lazy Evaluation for OCL*, in "Models 2015: 15th International Workshop on OCL and Textual Modeling", Ottawa, Canada, September 2015, https://hal.inria.fr/hal-01221072

Scientific Books (or Scientific Book chapters)

[37] W. BENGHABRIT, H. GRALL, J.-C. ROYER, M. SELLAMI, M. AZRAOUI, K. ELKHIYAOUI, M. ONEN, S. OLIVEIRA DE ANDERSON, K. BERNSMED. *From Regulatory Obligations to Enforceable Accountability Policies in the Cloud*, in "Cloud Computing and Services Sciences", Springer International Publishing Switzerland, 2015, CLOSER 2014, Barcelona Spain, April 3-5, 2014, Revised Selected Papers, https://hal.archives-ouvertes.fr/hal-01214387

Research Reports

[38] R.-A. CHERRUEAU, R. DOUENCE, M. SÜDHOLT. *A Language for the Composition of Privacy-Enforcement Techniques*, Inria Rennes ; École des Mines de Nantes ; Inria, April 2015, n^{o} RR-8720, https://hal.inria.fr/hal-01145694

[39] J. COHEN. *A Correct Refactoring Operation to Rename Global Variables in C Programs*, LINA-University of Nantes, December 2015, https://hal.archives-ouvertes.fr/hal-01248121

[40] A. LEBRE, J. PASTOR, . THE DISCOVERY CONSORTIUM. *The DISCOVERY Initiative - Overcoming Major Limitations of Traditional Server-Centric Clouds by Operating Massively Distributed IaaS Facilities*, Inria, September 2015, n^{o} RR-8779, 14 p. , https://hal.inria.fr/hal-01203648

[41] J. LEJEUNE, L. ARANTES, J. SOPENA, P. SENS. *Reducing synchronization cost in distributed multi-resource allocation problem*, Ecole des Mines de Nantes, Inria, LINA ; Sorbonne Universités, UPMC, CNRS, Inria, LIP6 ; Inria, February 2015, n^{o} RR-8689, https://hal.inria.fr/hal-01120808

Other Publications

[42] G. LEWERTOWSKI. *Ensembles nominaux dans Coq/SSreflect*, Univeristé Paris Diderot Paris 7, September 2015, https://hal.inria.fr/hal-01250862

[43] F. MARCHAND DE KERCHOVE, J. NOYÉ, M. SÜDHOLT. *Open Scope: A Pragmatic JavaScript Pattern for Modular Instrumentation*, June 2015, working paper or preprint, https://hal.archives-ouvertes.fr/hal-01181143

References in notes

[44] M. AKŞIT, S. CLARKE, T. ELRAD, R. E. FILMAN (editors). *Aspect-Oriented Software Development*, Addison-Wesley Professional, September 2004

[45] C. ALLAN, P. AVGUSTINOV, A. S. CHRISTENSEN, L. HENDREN, S. KUZINS, O. LHOTÁK, O. DE MOOR, D. SERENI, G. SITTAMPALAM, J. TIBBLE. *Adding trace matching with free variables to AspectJ*, in "ACM Conference on Object-Oriented Programming, Systems and Languages (OOPSLA)", R. P. GABRIEL (editor), ACM Press, 2005

[46] R. ALLEN, D. GARLAN. *A Formal Basis for Architectural Connection*, in "ACM Transactions on Software Engineering and Methodology", July 1997, vol. 6, n^{o} 3, pp. 213–49

[47] J. H. ANDREWS. *Process-Algebraic Foundations of Aspect-Oriented Programming*, in "Proceedings of the 3rd International Conference on Metalevel Architectures and Separation of Crosscutting Concerns", Lecture Notes in Computer Science, 2001, vol. 2192, pp. 187–209

[48] M. ARMBRUST, A. FOX, R. GRIFFITH, A. D. JOSEPH, R. KATZ, A. KONWINSKI, G. LEE, D. PATTERSON, A. RABKIN, I. STOICA, M. ZAHARIA. *A view of cloud computing*, in "Communications of the ACM", 2010, vol. 53, n^o 4, pp. 50–58

[49] T. H. AUSTIN, C. FLANAGAN. *Multiple facets for dynamic information flow*, in "Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages", New York, USA, POPL '12, ACM, 2012, pp. 165–178, http://doi.acm.org/10.1145/2103656.2103677

[50] A. BELOGLAZOV, R. BUYYA. *Energy efficient resource management in virtualized cloud data centers*, in "in: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID'10, IEEE Computer Society", 2010, pp. 826–831

[51] L. D. BENAVIDES NAVARRO, M. SÜDHOLT, W. VANDERPERREN, B. DE FRAINE, D. SUVÉE. *Explicitly distributed AOP using AWED*, in "Aspect-Oriented Software Development (AOSD)", ACM Press, March 2006, pp. 51-62

[52] G. S. BLAIR, G. COULSON, P. ROBIN, M. PAPATHOMAS. *An architecture for next generation middleware*, in "Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing", Springer-Verlag, 1998

[53] A. BRACCIALIA, A. BROGI, C. CANAL. *A formal approach to component adaptation*, in "Journal of Systems and Software", 2005

[54] R. E. BROWN, E. R. MASANET, B. NORDMAN, W. F. TSCHUDI, A. SHEHABI, J. STANLEY, J. G. KOOMEY, D. A. SARTOR, P. T. CHAN. *Server and Data Center Energy Efficiency: Public Law 109-431*, in "Report to Congress", 06/2008 2008

[55] S. CAPECCHI, I. CASTELLANI, M. DEZANI-CIANCAGLINI, T. REZK. *Session Types for Access and Information Flow Control*, in "CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings", P. GASTIN, F. LAROUSSINIE (editors), Lecture Notes in Computer Science, Springer, 2010, vol. 6269, pp. 237–252, http://dx.doi.org/10.1007/978-3-642-15375-4_17

[56] E. M. CLARKE, O. GRUMBERG, D. A. PELED. *Model Checking*, The MIT Press, Cambridge, Massachusetts, 1999

[57] A. COLYER, A. CLEMENT. *Large-scale AOSD for Middleware*, in "Proceedings of the 3rd ACM Int. Conf. on Aspect-Oriented Software Development (AOSD), Lancaster", K. LIEBERHERR (editor), ACM Press, 2004, pp. 56–65

[58] F. DEREMER, H. H. KRON. *Programming-in-the-large versus programming-in-the-small*, in "IEEE Transactions on Software Engineering", 1976, vol. SE-2, n^o 2, pp. 80-86

[59] G. DECKER, O. KOPP, F. LEYMANN, M. WESKE. *BPEL4Chor: Extending BPEL for Modeling Choreographies*, in "IEEE International Conference on Web Services (ICWS 2007)", IEEE Computer Society, 2007, pp. 296–303

[60] E. W. DIJKSTRA. *On the role of scientific thought*, in "Selected Writings on Computing: A Personal Perspective", Springer-Verlag, 1974, pp. 60–66, Published in 1982

[61] R. DOUENCE, P. FRADET, M. SÜDHOLT. *A framework for the detection and resolution of aspect interactions*, in "Proceedings of the ACM SIGPLAN/SIGSOFT Conference on Generative Programming and Component Engineering (GPCE'02)", Lecture Notes in Computer Science, Springer-Verlag, October 2002, vol. 2487, pp. 173–188, http://hal.inria.fr/inria-00072153

[62] R. DOUENCE, P. FRADET, M. SÜDHOLT. *Trace-Based Aspects*, in "Aspect-Oriented Software Development", M. AKŞIT, S. CLARKE, T. ELRAD, R. E. FILMAN (editors), Addison-Wesley, 2004, pp. 201-218

[63] R. DOUENCE, O. MOTELET, M. SÜDHOLT. *A formal definition of crosscuts*, in "Proceedings of the 3rd International Conference on Metalevel Architectures and Separation of Crosscutting Concerns", Lecture Notes in Computer Science, Springer-Verlag, 2001, vol. 2192, pp. 170–186

[64] R. DOUENCE, D. LE BOTLAN, J. NOYÉ, M. SÜDHOLT. *Concurrent Aspects*, in "Proc. of the Int. ACM Conf. on Generative Programming and Component Engineering (GPCE)", ACM Press, October 2006, pp. 79-88

[65] H. FOSTER, S. UCHITEL, J. MAGEE, J. KRAMER. *Model-based Verification of Web Service Compositions*, in "Proceedings of the 18th IEEE Int. Conf. on Automated Software Engineering (ASE'03)", IEEE Computer Society, 2003, pp. 152–163

[66] I. FOSTER, Y. ZHAO, I. RAICU, S. LU. *Cloud computing and grid computing 360-degree compared*, in "Grid Computing Environments Workshop, 2008. GCE'08", Ieee, 2008, pp. 1–10

[67] A. FUGGETTA, G. P. PICCO, G. VIGNA. *Understanding Code Mobility*, in "IEEE Transactions on Software Engineering", May 1998, vol. 24, n^{o} 5, pp. 342–361

[68] GREENPEACE. *Make IT green: Cloud computing and its contribution to climate change*, Greenpeace International, March 2010

[69] K. HONDA, N. YOSHIDA, M. CARBONE. *Multiparty asynchronous session types*, in "Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008", G. C. NECULA, P. WADLER (editors), ACM, 2008, pp. 273–284, http://www.doc.ic.ac.uk/~yoshida/multiparty/multiparty.pdf

[70] G. KICZALES, E. HILSDALE, J. HUGUNIN, M. KERSTEN, J. PALM, W. G. GRISWOLD. *An Overview of AspectJ*, in "ECOOP 2001 — Object-Oriented Programming 15th European Conference, Budapest Hungary", Berlin, J. L. KNUDSEN (editor), Lecture Notes in Computer Science, Springer-Verlag, June 2001, vol. 2072, pp. 327–353, http://www.eclipse.org/aspectj/

[71] G. KICZALES. *Aspect Oriented Programming*, in "Proc. of the Int. Workshop on Composability Issues in Object-Orientation (CIOO'96) at ECOOP", July 1996, Selected paper published by dpunkt press, Heidelberg, Germany

[72] G. KICZALES, J. DES RIVIERES, DANIEL G. BOBROW. *The Art of the Meta-Object Protocol*, MIT Press, Cambridge (MA), USA, 1991

[73] J. KIENZLE, R. GUERRAOUI. *AOP - Does It Make Sense? The Case of Concurrency and Failures*, in "16th European Conference on Object-Oriented Programming (ECOOP'2002)", Malaga, Spain, B. MAGNUSSON (editor), Lecture Notes in Computer Science, Springer-Verlag, 2002

[74] T. LEDOUX. *OpenCorba: a Reflective Open Broker*, in "ACM Meta-Level Architectures and Reflection, Second International Conference, Reflection'99", Saint-Malo, France, P. COINTE (editor), Lecture Notes in Computer Science, Springer-Verlag, July 1999, vol. 1616, pp. 197–214

[75] M. MCILROY. *Mass produced software components*, in "Mass produced software components", Garmish, Germany, P. NAUR, B. RANDELL (editors), NATO Science Committee, October 1968, pp. 138-155

[76] N. MEDVIDOVIC, R. N. TAYLOR. *A Classification and Comparison Framework for Software Architecture Description Languages*, in "IEEE Transactions on Software Engineering", January 2000, vol. 26, n^{o} 1, pp. 70-93

[77] M. MERNIK, J. HEERING, A. M. SLOANE. *When and How to Develop Domain-Specific Languages*, in "ACM Computing Surveys", December 2005, vol. 37, n^{o} 4, pp. 316-344

[78] L. MIKHAJLOV, E. SEKERINSKI. *A study of the fragile base class*, in "A study of the fragile base class", Brussels, Belgium, E. JUL (editor), Lecture Notes in Computer Science, July 1998, vol. 1445, pp. 355-382

[79] D. H. NGUYEN, M. SÜDHOLT. *VPA-based aspects: better support for AOP over protocols*, in "4th IEEE International Conference on Software Engineering and Formal Methods (SEFM'06)", IEEE Computer Society Press, September 2006

[80] O. NIERSTRASZ. *Regular Types for Active Objects*, in "Object-Oriented Software Composition", O. NIERSTRASZ, D. TSICHRITZIS (editors), Prentice Hall, 1995, chap. 4, pp. 99–121

[81] M. NISHIZAWA, S. CHIBA, M. TATSUBORI. *Remote Pointcut - A Language Construct for Distributed AOP*, in "Proceedings of the 3rd ACM Int. Conf. on Aspect-Oriented Software Development (AOSD), Lancaster", ACM Press, 2004

[82] D. L. PARNAS. *On the criteria for decomposing systems into modules*, in "Communications of the ACM", December 1972, vol. 15, n^{o} 12, pp. 1053-1058

[83] S. PEARSON. *Toward Accountability in the Cloud*, in "Internet Computing, IEEE", July-Aug. 2011, vol. 15, n^{o} 4, pp. 64-69, http://dx.doi.org/10.1109/MIC.2011.98

[84] F. PLASIL, S. VISNOVSKY. *Behavior Protocols for Software Components*, in "Transactions on Software Engineering", January 2002, vol. 28, n^{o} 9

[85] F. PUNTIGAM. *Coordination Requirements Expressed in Types for Active Objects*, in "ECOOP'97—Object-Oriented Programming", M. AKŞIT, S. MATSUOKA (editors), Lecture Notes in Computer Science, Springer-Verlag, 1997, vol. 1241, pp. 367–388

[86] N. SHARMA, S. BARKER, D. IRWIN, P. SHENOY. *Blink: managing server clusters on intermittent power*, in "SIGARCH Comput. Archit. News", March 2011, vol. 39, pp. 185–198, http://dx.doi.org/10.1145/1961295.1950389

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

[88] B. C. SMITH. *Reflection and Semantics in LISP*, Xerox Palto Alto Research Center, Palo Alto, 1984, n^{o} P84-00030

[89] S. SOARES, E. LAUREANO, P. BORBA. *Implementing distribution and persistence aspects with AspectJ* , in "Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications (OOPSLA-02)", C. NORRIS, J. J. B. FENWICK (editors), ACM SIGPLAN Notices, ACM Press, November 4–8 2002, vol. 37, 11, pp. 174–190

[90] S. SUNDARESWARAN. *Ensuring Distributed Accountability for Data Sharing in the Cloud*, in "Dependable and Secure Computing", 2012, vol. 9, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6165313

[91] R. J. WALKER, K. VIGGERS. *Implementing Protocols via Declarative Event Patterns*, in "Proceedings of the ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE-12)", ACM Press, 2004, pp. 159 - 169

[92] M. WAND, G. KICZALES, C. DUTCHYN. *A Semantics for Advice and Dynamic Join Points in Aspect-Oriented Programming*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", 2004, vol. 26, n^{o} 5, pp. 890–910

[93] D. M. YELLIN, R. E. STROM. *Protocol specifications and component adaptors*, in "ACM Transactions of Programming Languages and Systems", March 1997, vol. 19, n^{o} 2, pp. 292–333

[94] Q. ZHANG, L. CHENG, R. BOUTABA. *Cloud computing: state-of-the-art and research challenges*, in "Journal of internet services and applications", 2010, vol. 1, n^{o} 1, pp. 7–18

[95] A. VAN DEURSEN, P. KLINT, J. VISSER. *Domain-Specific Languages: An Annotated Bibliography*, in "ACM SIGPLAN Notices", June 2000, vol. 35, n^{o} 6, pp. 26-36