Activity Report 2014

Team SCALE

Safe Composition of Autonomous applications with Large-SCALE Execution environment

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Distributed Systems and middleware

Table of contents

Team SCALE

Keywords: Distributed Objects, Component Programming, Distributed Computing, Formal Methods, Simulation

The SCALE Team is a followup to the OASIS Project Team that ended in 2013. See http://raweb.inria.fr/rapportsactivite/RA2013/oasis/ for the last activity report of OASIS.

Creation of the Team: 2014 January 01.

1. Members

Research Scientists

Ludovic Henrio [Team leader, CNRS, Researcher, HdR]
Eric Madelaine [Inria, Researcher, co-leader, HdR]

Faculty Members

Françoise Baude [Univ. Nice, Professor, HdR]
Olivier Dalle [Univ. Nice, Associate Professor]
Fabien Hermenier [Univ. Nice, Associate Professor]
Fabrice Huet [Univ. Nice, Associate Professor]

Engineers

Iyad Alshabani [CNRS, SONGS ANR Project, until December 2014]
Marco Biazzini [Inria, since May 2014]
Nassim Jibai [Inria, Spinnaker Oseo, and OpenCloudWare FUI, since May 2014]
Bartlomiej Szejna [Inria, Spinnaker Oseo, until May 2014]

PhD Students

Maeva Antoine [Univ. Nice]
Michael Benguigui [Inria]
Yanwen Chen [co-tutelle Univ. Nice - ECNU Shanghai, until December 2014]
Nuno Gaspar [Cifre ActiveEon SA.]
Oleksandra Kulankhina [Inria]
Sophie Ge Song [E.C. Paris and Univ. Nice]
Vincent Kherbache [Inria]
Vincenzo Mastandrea [Univ. Nice (Labex UCN@Sophia), from Oct 2014]
Laurent Pellegrino [Inria then Univ. Nice, until Aug 2014]
Justine Rochas [Univ. Nice]
Alexandra Savu [Inria, until Apr 2014]
Damian Vicino [co-tutelle Univ. Nice - Univ. Carleton (Canada)]

Visiting Scientists

Florian Kammuller [Invited Researcher (Middlesex Univ., UK)]
Gustavo Pabon [Invited Researcher (SCADA Associate Team), from Feb 2014 until Mar 2014]
Min Zhang [Invited Researcher (DAESD Associate Team), from Oct until Nov 2014]

Administrative Assistants

Christel Boulard Kozinski [Inria]
Françoise Dubois [CNRS]

Other

Yu Feng [Associate member, SIGMA ORIONIS]

2. Overall Objectives

2.1. From Oasis to Scale

The SCALE team aims at contributing to the safety of distributed applications, in particular through the research and development of programming models and runtime environments adapted to modern distributed architectures.

Since 2000, the OASIS team, successively led by Isabelle Attali, Denis Caromel, and Eric Madelaine has done research in the fields of distributed systems, semantics, and active objects, addressing problems in formal models and calculi, programming languages, and middleware for developing distributed applications.

Since 2013, the SCALE group (Safe Composition of adaptable Applications and Large-scale execution Environment) has succeeded to OASIS. SCALE maintains the balance and strong links between theory and application that made the success of OASIS. However, while OASIS was more focused on Grid-Computing and programming models for HPC, SCALE is more interested in the support for multi-scale and multi-level parallelism.

Indeed, a common programming idiom is to dissociate local parallelism targeted at multicore machines and distributed aspects, sometimes with several levels of distribution. The research in SCALE aims at providing a unified approach for multi-level parallel applications, with extended coverage of all forms and levels of parallelism, from multi-core to federation of clouds, with a strong focus on safety.

2.2. Overall Objectives

Nowadays, distributed applications can be found on an ever increasing number of interconnected computing infrastructures, ranging from small scale embedded devices to personnal devices, including laptops, desktops, palmtops, and tablets, and finally to large scale data-centers and high performance computing infrastructures. In addition to this *inter*-platform heteregeneity, we also face higher levels of *intra*-platform hetrogeneity, with the advent of multi-core architectures, virtual threads, GPUs, and new virtualization technologies. Programming distributed applications in such an heterogeneous environment, with high level of confidence both in terms of performance and security, is still a challenge, and raises many questions.

The overall objective of SCALE is to provide a unified programming language and execution model for such a large-scale multi-level parallelism. We want to contribute to the design of the programming languages that will be used to program large-scale distributed applications in 2020. The characteristics we want to put forward are: *ease of programming* and *guarantees of correctness*. For this purpose our approach mixes distributed systems, theory of programming languages, middleware implementations, and resource management. Guided by this interaction the SCALE team contributes to the creation of safe and efficient environments for programming and running distributed applications.

2.2.1. The Scale team and its positioning

The strength of SCALE is to put together researchers in programming languages, from a rather theoretical perspective, with researchers in middleware and object-oriented programming with a strong expertise on distributed systems and their applications.

During the last decade, the landscape of programming languages for distributed systems has changed. Actors and active objects have gained interest both from the industrial point of view, as illustrated by the success of Scala and Akka [1], and from the academic point of view, for example through the projects around ABS and Creol. Our new proposal, multi-active objects, gives a novel alternative in this world. It is expected to be more efficient and more easily applicable than academic languages, but with a better formalisation and more proved properties than the languages targeting industrial usage. In particular we believe we are able to provide the expertise allowing the transfer of academic results on active objects into language constructs and runtime support that can be adopted by industrial platforms like Akka.

[1] http://akka.io

The global objective to make safe programming accessible to most programmers directs the choice of the target programming language for our developments toward Java. This choice is not only the result of our expertise, but also because it is well known, accessible by most programmers, and also because the underlying JVM can be shared with other languages. This last feature opens interesting perspectives in terms of sharing and reuse of our software contributions. However, this choice is not exclusive. In particular, the approach we propose should be easy to transfer to other programming languages.

While we still rely on active objects as a medium-grain parallelism abstraction, we also propose new ways of parallelizing the execution inside active objects (fine-grain parallelism), and autonomic ways to deploy and run the coarse-grain entities composing the application. However, these various levels of programming and execution environments cannot be designed independently. They all rely on a small set of common abstractions. Typically the service of a request is a notion that can be reasoned about at all the considered levels. We think that providing such unifying programming and execution abstractions, and their formal specification, makes the originality and the strength of SCALE, while unifying our contributions.

Finally, the SCALE team wants to put a strong emphasis on safety for distributed applications. We want to apply our expertise about reasoning on language, programs, and protocols in order to ensure the safe execution of distributed systems and applications. The fact that we are able to prove the safe behaviour of applications that achieve performances comparable to unproven and unsafe distributed systems is probably the biggest strength of the SCALE team. Concerning safety, our approach is threefold: 1) provide high-level programming abstractions that makes the writing of distributed programs easier (and thus reduce the chances to write bugged programs); 2) prove the correctness of the platform we propose, by proving both properties of the programming language and properties on the tools and protocols we use; and 3) provide tools so that the users can specify the behaviour of their application and verify that it is correct.

2.2.2. Research challenges and objectives

To summarise, the challenges we address in the SCALE team are the following:

- *Safe and easy programming of large-scale distributed applications*. We work to design a unified programming model for the development of applications mixing local concurrency, e.g. at the level of multicore hardware, and large-scale distributed parallelism, like in cloud architecture. The language is based on the multi-active object language we designed in the last two years, extended with features for easing its programming.

  Among the improvements of the language, the strongest challenges are: 1) provide in the same programming model, simplicity of programming for non-experts, and optimisation capabilities for advanced users; and 2) safety guarantees, including a wide variety of analysis methods supporting the development of correct applications; these analysis methods include static verification of Java annotations and behavioural analysis of dynamic systems of unbounded size.

- *Easy, safe and efficient execution of large-scale distributed applications*. We work to design a runtime environment supporting our approach and showing its practical effectiveness. Also, and more originally, this challenge includes resource management aspects, and the possibility for the programmer to *easily* express resource requirements on his/her applications. Finally, the runtime support for application is also crucial here, and will include in Scale a wide range of aspects ranging from elasticity concerns to support for distributed debugging.

  Here, the biggest research challenges are: 1) allow the non-expert programmer to express constraints on his/her program that will allow us to better deploy the application and run it efficiently; 2) prove the correctness of the runtime platform and the tools we propose.

- *Experiments on real life scenarios*. Those scenarios will mostly be taken from big-data and simulation application domains.

  Here, the biggest research challenges are 1) Provide a convenient environment for programming and running big-data analytics. One of the key challenge we want to address is how to scale and adapt analytics at runtime both from the technical and from the business point of views. The streaming data variability, velocity, and volume evolve at runtime, so should the analytics computation. Even the computation itself might evolve depending on preliminary results, thus the business computation also has to be adaptable. 2) Provide an efficient support for the distributed execution of large discrete-event simulations; we target deployment of large instances of component-based simulation models, and processing of the large amount of data samples produced by simulations.

3. Research Program

3.1. Safely and easily programming large-scale distributed applications

Our first objective is to provide a programming model for multi-level parallelism adapted to the programming of both multi-core level parallelism, and of large-scale distributed systems. Experience shows that achieving efficient parallelism at different levels with a single abstraction is difficult, however we will take particular care to provide a set of abstractions that are well integrated and form a safe and efficient global programming model. This programming model should also provide particular support for adaptation and dynamicity of applications.

3.1.1. Basic model

The main programming abstraction we have started to explore is multi-active object. This is a major change in the programming model since we remove the strongest constraint of active objects: their mono-threaded nature. Mono-threaded active objects bring powerful properties to our programming model, but also several limitations, including inefficiency on multicore machines, and deadlocks difficult to avoid. Thus, our objective here is to gain efficiency and expressiveness while maintaining as many properties of the original ASP calculus as possible, including ease of programming. Multi-active objects is a valuable alternative to the languages *à la* Creol/JCobox/ABS, as it is more efficient and potentially easier to program. This programming model better unifies the notions of concurrent programming and distributed programming, it is thus a crucial building block of our unified programming model.

It is also important to study related concurrency paradigms. Indeed, multi-active objects will not provide a complete solution to low-level concurrency; for this we should study the relation and the integration with other models for concurrency control (different programming languages, transactional memory models, ...).

Even if a first version of the language is available, further developments are necessary. In particular, the formal study of its properties is still an open subject. This formalisation is crucial in order to guarantee the correctness of the programming model. We have a good informal vision of the properties of the language but proving and formalising them is challenging due to the richness of the language.

3.1.2. Higher-level features

Multi-active objects should provide a good programming model integrating fine grain parallelism with large-scale distribution. We also think that the programming abstractions existing at the lower levels should nicely be integrated and interact with coarser-grain composition languages, in order to provide a unified programming model for multi-level parallelism. We think that it is also crucial, for the practical usability of the language to *design higher-level synchronisation primitives*. Indeed, a good basic programming language is not sufficient for its adoption in a real setting. Richer synchronisation primitives are needed to simply write complex interactions between entities running in parallel. The coexistence of several levels of parallelism will trigger the need for new primitives synchronising those several levels. Then the implementation of those primitives will require the design of new communication protocols that should themselves be formalised and verified.

One of the objectives of SCALE is also to provide frameworks for composing applications made of interacting distributed entities. The principle here would be to build basic composing blocks, typically made of a few multi-active objects, and then to compose an application made of these blocks using a coarser grain composition, like software components. What is particularly interesting is that we realised that software components also provide a component abstraction for reasoning on (compositional) program verification, or on autonomic adaptation of software and that active objects provide programming abstractions that fit well with software components. In the last years, the researchers of SCALE proposed GCM, a component model adapted to distribution and autonomic behaviour. We will reuse these results and adapt them. An even more challenging perspective consists in the use of component models for specifying discrete-event based simulations made up of different concerns; this will be a strong connection point between objective 1 and 3.

Finally, there still exists a gap between traditional programming languages like multi-active objects and coarse-grain composition languages like map-reduce paradigm. We want to investigate the interactions between these multiple layers of parallelism and provide a unified programming model.

3.1.3. Reliability of distributed applications

From the rigorous formalisation of the programming model(s), to the (assisted) proofs of essential properties, the use of model-checking-based methods for validating early system development, the range of formal method tools we use is quite large but the members of the teams are knowledgeable in those aspects. We also expect to provide tools to the programmers based on MDE approaches (with code-generation). While we might provide isolated contribution to theoretical domains, our objective is more to contribute to the applicability of formal methods in real development and runtime environments. We shall adapt our behavioural specification and verification techniques to the concurrency allowed in multi-active objects. Being able to ensure safety of multi-active objects will be a crucial tool, especially because those objects will be less easy to program than mono-threaded active objects.

Our experience has shown that model-checking methods, even when combining advanced abstraction techniques, state-of-the-art state-space representation, compositional approaches, and large-scale distributed model-checking engines, is (barely) able to master “middle-size” component systems using one complex interaction pattern (many-to-many communications), and/or a simple set of reconfiguration. If we want to be able to model complex features of distributed systems, and to reason on autonomic software components, verification techniques must scale. We strongly believe that further scalability will come from combination of theorem-proving and model-checking approaches. In a first step, theorem-proving can be used to prove generic properties of the model, that can be used to build smaller behavioural models, and reduce the model-checking complexity (reducing the model size, using symmetry properties, etc.). In a second step, we will use model-checking techniques on symbolic models that will rely on theorem proving for discharging proof obligations.

3.2. Easily, safely and efficiently running large-scale distributed applications

Concerning runtime aspect, a first necessary step is to to provide a runtime that can run efficiently the application written using the programming model described in objective 1. The proposed runtime environment will rely on commodity hosting platforms such as testbeds or clouds for being able to deploy and control, on demand, the necessary software stacks that will host the different applications components. The ProActive platform will be used as a basis that we will extend. Apart from autonomic adaptation aspects and their proof of correctness, we do not think that any new major research challenges will be solved here. However it is crucial to perform the necessary developments in order to show the practical effectiveness of our approach, and to provide a convenient and adaptable runtime to run the applications developed in the third objective about application domains.

3.2.1. Mapping and deploying virtual machines

The design of a cloud native application must follow established conventions. Among other things, true elasticity requires stateless components, load balancers, and queuing systems. The developer must also establish, with the cloud provider, the Service Level Agreements (SLAs) that state the quality of services to offer. For example, the amount of resources to allocate, the availability rate or possible placement criteria. In a private cloud, when the SLA implementation is not available, the application developer might be interested in implementing its own. Each developer must then master cloud architecture patterns and design his/her code accordingly. For example, he must be sure there is no single point of failures, that every elastic components is stateless that the balancing algorithms do not loose requests upon slave arrival and departure or the messaging protocol inside the queuing system is compatible with his/her usage. To implement a SLA enforcement algorithm, the developer must also master several families of combinatorial problems such as assignment and task scheduling, and ensure that the code fits the many possible situations. For example, he must consider the implication of every possible VM state on the resource consumption. As a result, the development and the deployment of performant cloud application require excessive skills for the developers.

The first original aspect we will push in this domain is related to safety and verification. It is established that OS kernels are critical softwares and many works proposed design to make them trustable through kernels and driver verifications. The VM scheduler is the new OS kernel but despite the economical damages a bug can cause, no one currently proposes any solution other than unit testing to improve the situation. As a result, production clouds currently run defective implementations. To address this critical situation we propose to formalise the specifications of VM scheduling primitives. Any developer should be able to specify his/her primitives. To fit their limited expertise in existing formal language, we will investigate for a domain specific language. This language will be used to prove the specified primitives with respect to the scheduler invariants. Second, it will make possible to generate the code of critical scheduler components. Typically the SLA enforcement algorithms. Third, the language will be used to assist at debugging legacy code and exhibit implementation bugs. Fabien Hermenier is already developing a language for specifying constraints for our research prototype VM Scheduler *BtrPlace*. *SafePlace* will be the name of the verification platform, we started its design and development in 2014.

The second challenge in this domain is to investigate the relation between programming languages, VM placement algorithms, allocation of resources, elasticity and adaptation concerns. The goal here is to enable the programmer to easily write and deploy scalable cloud applications by hiding with our programming model, the mechanisms the developer currently has to deal with explicitly today. This includes among other things to make transparent the notion of elastic components, elasticity rules, load balancing, or message queuing.

3.2.2. Debugging and fault-tolerance

We also aim at contributing to aspects that usually belong to pure distributed systems, generally from an algorithmic perspective. Indeed, we think that the approach we advocate is particularly interesting to bring new ideas to these research domains because of the interconnection between language semantics, protocols, and middleware. Typically, the knowledge we have on the programming model and on the behaviour of programs should help us provide dedicated debuggers and fault-tolerance protocols.

In fact some research has already been conducted in those domains, especially on reversible debuggers that allow the navigation inside a concurrent execution, doing forward and backward steps [2]. We think that those related works show that our approach is both relevant and timely. Moreover, little has been done for systems based on actors and active objects. The contribution we aim here is to provide debuggers able to better observe, introspect, and replay distributed executions. Such a tool will be of invaluable help to the programmer. Of course we will rely on existing tool for the local debugging and focus on the distributed aspects.

4. Application Domains

4.1. Simulation

4.1.1. Discrete-event simulation

[2] Causal-Consistent Reversible Debugging. Elena Giachino, Ivan Lanese, and Claudio Antares Mezzina. *FASE 2014*.

Simulation is an example of an application with ever increasing computation needs that would benefit from the SCALE research results. In emergency planning and response, for example, users need to access the power of large scale distributed computing facilities to run faster than real-time simulations of the situations they face on the field; Such a computation can mix heterogeneous distributing computing platforms (PDA and laptops on the field, Cloud and HPC in background) and use a number of external services (eg. weather forecast).

Simulations made of multi-party contributed software models also demonstrate the need for a unifying and user-friendly programming model. Indeed, since the early 70's, the simulation field have been the subject of many efforts in order to abstract the computation models from their actual application domain. DEVS (Discrete Event Systems specification), is an example of such a popular formalism in the simulation community that breaks-down the representation of a simulation model into hierarchical components.

Our objective is to focus on the operational support of execution for such simulation models. For example, considering that the model of a single node of a Peer-to-peer network requires several (and possibly many) DEVS components, it is easy to see that running simulations of a realistic large-scale peer-to-peer network rapidly ends-up involving millions of DEVS components. In addition to the problems posed by the execution of a distributed simulation application made of millions of components, such a use-case is also challenging in terms of analytics, because when millions of components are instrumented to collect observations, it becomes a typical instance of a big-data analytics problem.

4.1.2. Stochastic simulation platform

Understanding how complex objects, as found in finance/insurance (option contracts), biology (proteins structure), etc. evolve is often investigated by stochastic simulations (e.g. Monte-Carlo based). These can be very computational intensive and the associated communities are always seeking adequate parallel computing infrastructures and simulation software. Being able to harness all the available computing power, while ensuring the simulation is at first performant but also robust, capable to self-adapt, e.g. to failures, is a real opportunity for research and validation of our approach. Many other simulation applications could also benefit from our models and techniques, and we may in the future set up specific collaborations, e.g. in biocomputing, data-center activity management, or other engineering domains. We have recently solved pricing of high-performance demanding financial products on heterogeneous GPUs and multicore CPUs clusters, mixing use of active objects and OpenCL codes. This kind of application could continue to serve as a benchmark for our multi-level programming model.

4.2. Big data

4.2.1. Big data analytics

The amount of data digitally produced is increasing at an exponential rate. Having a dedicated programming model and runtime, such as Hadoop-MapReduce, has proved very useful to build efficient big data mining and analysis applications albeit for very static environments. However, if we consider that not only the environment is dynamic (node sharing, failures...) but so are the data (variation in popularity, arrival rate...), it becomes a much more complex problem. This domain is thus a very good candidate as an application field for our work.

More precisely, we plan to contribute at the deployment level, runtime level, and at the analytics programming model for the end-user level. We already worked on close topics with the distributed P2P storage and publish/subscribe system for Semantic Web data (named *EventCloud*). However, expressing a particular interest about data through simple or even more complex subscriptions (CEP) is only a first step in data analytics. Going further requires the full expressivity of a programming language to express how to mine into the real-time data streams, aggregate intermediate analytics results, combine with past data when relevant, etc. We intend to enlarge this effort about extracting meaningful information by also creating tighter collaborations with groups specialized in data mining algorithms (e.g. the Mind team at I3S).

We think that the approach advocated in SCALE is particularly adapted to the programming and support of analytics. Indeed, the mix of computational aspects and of large amount of data make the computation of analytics the perfect target for our programming paradigms. We aim at illustrating the effectiveness of our approach by experimenting on different computations of analytics, but we will put a particular focus on the case of data streams, where the analysis is made of chains (even cyclic graphs) of parallel and distributed operators. These operators can naturally be expressed as coarse grained composition of fine grained parallel entities, both granularity levels featuring autonomic adaptation. Also, the underlying execution platform that supports this execution also has to feature autonomic adaptation in order to deal with an unstable and heterogeneous execution environment. Here autonomic adaptation is also crucial because the programmer of analytics is not expected to be an expert in distributed systems.

Overall, this second application domain target should illustrate the effectiveness of our runtime platform and of our methodology for dynamic and autonomic adaptation.

5. New Software and Platforms

5.1. Platforms

5.1.1. EventCloud

Participants : Iyad Alshabani, Maéva Antoine, Françoise Baude, Fabrice Huet, Laurent Pellegrino.

The *EventCloud* is an open source middleware that aims to act as a distributed datastore for data fulfiling the W3C RDF specification (http://www.w3.org/RDF/). It allows to store and retrieve quadruples (RDF triples with context) through SPARQL but also to manage events represented as quadruples. The *EventCloud* architecture is based on a structured P2P overlay network targetting high-performance elastic data processing. Consequently it aims to be deployed on infrastructures like grids, clouds, i.e. whose nodes acquisition and relinquishment can be dynamic and subject to a pay-per-use mode. Each node participating in the overlay networks constituting EventCloud instances, is responsible for managing the storage of subsets of the events, and helps in matching potential looked up events and disseminating them in a collaborative manner. As such, each node is also potentially an event broker responsible for managing subscriptions and routing notifications.

The *EventCloud* provides a high level publish-subscribe API where users can register their interests using SPARQL. When matching RDF data are added, subscribers are automatically notified. Recent work around the *EventCloud* has focused on efficient algorithms for managing subscription and notification.

5.1.2. BtrPlace

Participants : Fabien Hermenier, Vincent Kherbache, Ludovic Henrio.

BtrPlace is an open source virtual machine (VM) scheduler for datacenters. *BtrPlace* has been designed to be extensible. It can be customized by plugins from third party developers to address new SLAs or optimization constraints. Its extensibility is possible thanks to a composable core scheduling algorithm implemented using Constraint Programming. *BtrPlace* is currently bundled with a catalog of more than 20 constraints to address performance, fault tolerance, isolation, infrastructure management or energy efficiency concerns. It is currently used inside the FSN project OpenCloudWare (http://opencloudware.org/) and the European project DC4Cities (http://dc4cities.eu/).

This year we first put an emphase on *BtrPlace* dissemination. BtrPlace has been frequently released and it is now available online on a dedicated Web site (http://btrplace.org). To increase its visibility and to ease its integration, we decided to made *BtrPlace* directly available from the central repository of maven, the standard system to manage Java projects. Finally, BtrPlace has been registered on the *Agence pour la Protection des Programmes*.

5.1.3. OSA

Participants : Olivier Dalle.

OSA stands for Open Simulation Architecture. OSA (http://osa.inria.fr/) is primarily intended to be a federating platform for the simulation community: it is designed to favor the integration of new or existing contributions at every level of its architecture. The platform core supports discrete-event simulation engine(s) built on top of the ObjectWeb Consortium?s Fractal component model. In OSA, the systems to be simulated are modeled and instrumented using Fractal components. In OSA, the event handling is mostly hidden in the controller part of the components, which alleviates noticeably the modeling process, but also eases the replacement of any part of the simulation engine. Apart the simulation engine, OSA aims at integrating useful tools for modeling, developing, experimenting, and analysing simulations. OSA is also a platform for experimenting new techniques and approaches in simulation, such as aspect oriented programming, separation of concerns, innovative component architectures, and so on.

5.1.4. VerCors

Participants: Eric Madelaine, Ludovic Henrio, Bartlomiej Szejna, Nassim Jibai, Oleksandra Kulankhina, Siqi Li.

The Vercors tools (http://www-sop.inria.fr/oasis/Vercors) include front-ends for specifying the architecture and behaviour of components in the form of UML diagrams. We translate these high-level specifications, into behavioural models in various formats, and we also transform these models using abstractions. In a final step, abstract models are translated into the input format for various verification toolsets. Currently we mainly use the various analysis modules of the CADP toolset.

We have achieved this year a major version of the platform frontend, named VCE-v3, that is now distributed on our website, and used by some of our partners. It includes integrated graphical editors for GCM component architecture descriptions, UML classes, interfaces, and state-machines. The user diagrams can be checked using the recently published validation rules from [11]; then the corresponding GCM components can be executed using an automatic generation of the application ADL, and skeletons of Java files.

But VCE-v3 is using the Obeo-designer platform, which is a commercial product, and we have started a port to the newly available Sirius platform (http://eclipse.org/sirius/), with the goal to distribute the next major release of VCE, next year, under Sirius.

6. New Results

6.1. Programming Languages for Distributed Systems

One of the objectives of the Scale team is to design programming models easing the development and safe execution of distributed systems. This section describes our results in this direction.

6.1.1. Multi-active Objects

Participants: Ludovic Henrio, Fabrice Huet, Justine Rochas, Vincenzo Mastandrea.

The active object programming model is particularly adapted to easily program distributed objects: it separates objects into several *activities*, each manipulated by a single thread, preventing data races. However, this programming model has its limitations in terms of expressiveness – risk of deadlocks – and of efficiency on multicore machines. We proposed to extend active objects with *local multi-threading*. We rely on declarative *annotations* for expressing potential concurrency between requests, allowing easy and high-level expression of concurrency. This year we realized the following:

- We published the extension of multi-active objects to support scheduling and thread limitation [12].
- We developed a compiler from ABS language into ProActive multi-active objects. This translation can be generalised to many other active object languages. This work has been published as a research report [25], and is under submission to a conference.

- We started to work on static detection of deadlocks for multi-active object. This is the work of Vincenzo Mastandrea who is starting a Labex PhD in collaboration with the FOCUS EPI (Univ of Bologna).
- Extensive use of multiactive objects in our CAN P2P network and implementation of usecases [2].
- We formalised in Isabelle/HOL a first version of the semantics of multiactive objects. This work was done in collaboration with Florian Kammuller

We plan to continue to improve the model, especially about compile-time checking of annotations and about fault tolerance of multiactive objects.

6.1.2. Autonomic Monitoring and Management of Components

Participants: Françoise Baude, Ludovic Henrio.

We have completed the design of a framework for autonomic monitoring and management of component-based applications. We have provided an implementation using GCM/ProActive taking advantage of the possibility of adding components in the membrane. The framework for autonomic computing allows the designer to describe in a separate way each phase of the MAPE autonomic control loop (Monitoring, Analysis, Planning, and Execution), and to plug them or unplug them dynamically.

- This year, we published a journal paper summarising our approach in the GCM/ProActive framework and our contribution on componentised membranes for autonomic computing [3].
- We also improved, in the context of the SCADA associate team and during the internship of Matias Ibañez, the support for autonomic components, providing all the architecture and API so that the programmer of autonomic aspect can do them in a DSL reconfiguration language, called GCMScript. This was implemented and experimented, a publication is under submission on this work.

6.1.3. Algorithmic skeletons

Participant: Ludovic Henrio.

In the context of the SCADA associated team, we worked on the algorithmic skeleton programming model. The structured parallelism approach (skeletons) takes advantage of common patterns used in parallel and distributed applications. The skeleton paradigm separates concerns: the distribution aspect can be considered separately from the functional aspect of an application. In the previous year we designed the possibility for a skeleton to output events, which increases the control and monitoring capabilities. This year we published our previous results in [14] and realised additional steps:

- Study of different ways to predict the execution time for a skeleton, inspired from simple statistic functions. This improvement together with the distributed execution of skeletons should allow us to publish a journal paper on this subject in 2015

6.1.4. Optimization of data transfer in event-based programming models

Participants: Iyad Alshabani, Françoise Baude, Laurent Pellegrino.

In [6], we extended a previous work with conceptual and experimental performance evaluations. This previous and collaborative work [1] developed an innovative approach of “lazy copy and transfer” of the data parts of event objects exchanged by peers in the context of event-driven architecture applications.

While event notifications are routed in a conventional manner through an event service, data parts of the events are directly and transparently transferred from publishers to subscribers. The theoretical analysis shows that we can reduce the average event delivery time by half, compared to a conventional approach requiring the full mediation of the event service. The experimental analysis confirms that the proposed approach outperforms the conventional one (both for throughput and delivery time) even though the middleware overhead, introduced by the specific adopted model, slightly reduces the expected benefits.

6.1.5. Behavioural Semantics

Participants: Ludovic Henrio, Eric Madelaine, Min Zhang.

We have studied Parameterised Networks of Automata (pNets) from a theoretical perspective. We started with some 'pragmatic' expressiveness of the pNets formalism, showing how to express a wide range of classical constructs of (value-passing) process calculi, but also complex interaction patterns used in modern distributed systems. Our framework can model full systems, using (closed) hierarchies of pNets; we can also build (open) pNet systems expressing composition operators. Concerning more fundamental aspects, we defined a strong bisimulation theory specifically for the pNet model, proved its properties, and illustrated it on some examples. One of the original aspects of the approach is to relate the compositional nature of pNets with the notion of bisimulation; this was exemplified by studying the properties of a flattening operator for pNets. This work has been accepted for publication at PDP'2015 ([24]).

6.1.6. A Time-sensitive Heterogeneous Behavioural Model

Participants: Eric Madelaine, Yanwen Chen.

This work concludes the PhD research of Yanwen Chen, targetting a timed-sensitive extension of the pNets model with logical clocks inherited from the CCSL language. The main results of this year are: 1) a new notion of Time Specification (TS), used to handle the abstract properties of each level of processes in a pNet structure, 2) algorithms to compute such TSs for basic parameterized and timed processes, and from composition of timed-pNets, 3) conditions for checking the compatibility of composition, 4) a use-case from the area of intelligent transportation systems, illustrating the whole chain of modeling, upto a symbolic simulation of the full composed system, with the TimeSquare tool. This work was published as [4], [23], and in the PhD thesis of Y. Chen, defended on 2014, Nov. 30th.

6.1.7. Structure and structural correctness for GCM components

Participants: Ludovic Henrio, Oleksandra Kulankhina, Eric Madelaine.

We have defined a set of rules characterizing the well-formed composition of components in order to guarantee their safe deployment and execution. This work focuses on the structural aspects of component composition; it puts together most of the concepts common to many component models, but never formalized as a whole. Our formalization characterizes correct component architectures made of functional and non-functional aspects, both structured as component assemblies. So-called 'Interceptor chains' can be used for a safe and controlled interaction between the two aspects. Our well-formed components guarantee a set of properties ensuring that the deployed component system has a correct architecture and can run safely. Those definitions constitute the formal basis for VerCors tool. This work was done in the context of O. Kulankhina phd research, and in collaboration with Dongqian Liu (ECNU Shanghai), as part of the Associated Team DAESD.

6.2. Run-time/middle-ware level

6.2.1. Scalable and robust Middleware for distributed event based computing

Participants: Françoise Baude, Fabrice Huet, Laurent Pellegrino, Maeva Antoine.

In the context of the FP7 STREP PLAY and French SocEDA ANR research projects terminated late 2013, we initiated and pursued the design and development of the Event Cloud. This has been the core content of Laurent Pellegrino PhD thesis [2], and the corresponding software deposit at the APP for this middleware.

As a distributed system, this middleware can suffer from failures. To resist to such situations, we have added a capability of checkpointing. In [18] we present how to design an adaptation of the famous Chandy and Lamport algorithm for distributed snapshot taking, to the case of the Event Cloud. Indeed, as the Event Cloud peers are multi-active objects, we need to take care when and how to serve the chekpointing request and so, when to apply the Chandy Lamport protocol operations. Consequently, we have make sure that the obtained distributed snapshot is indeed consistent. As publication of events are triggered from the outside of the Event Cloud, we however are not able to recover them from the last saved snapshot in case of peer crash and subsequent whole Event Cloud recovery. However, we ensure any event injected through a peer, before this peer was participating in the last global checkpoint taking is safely part of it.

As a distributed system handling huge amount of information, this middleware can suffer from data imbalances. In [22], [8], we have reviewed the litterature of structured peer to peer systems regarding the way they handle load imbalance. We have generalized those popular approaches by proposing a core API that we have proved to be indeed also applicable to the Event Cloud middleware way of implementing a load balancing policy.

Storing highly skewed data in a distributed system has become a very frequent issue, in particular with the emergence of semantic web and big data. This often leads to biased data dissemination among nodes. Addressing load imbalance is necessary, especially to minimize response time and avoid workload being handled by only one or few nodes. We have proposed a protocol which allows a peer to change its hash function at runtime, without a priori knowledge regarding data distribution. This provides a simple but efficient adaptive load balancing mechanism. Moreover, we have shown that a structured overlay can still be consistent event when all peers do not apply the same hash function on data [7].

6.2.2. *Virtual Machines Placement Algorithms*

Participants: Fabien Hermenier, Vincent Kherbache.

In [21], [19], we present BtrPlace as an application of the dynamic bin packing problem with a focus on its dynamic and heterogeneous nature. We advocate flexibility to answer these issues and present the theoretical aspects of BtrPlace and its modeling using Constraint Programming. In [5] we rely on BtrPlace to achieve energy efficiency. To maintain an energy footprint as low as possible, data centres manage their VMs according to conventional and established rules. Each data centre is however made unique due to its hardware and workload specificities. This prevents the *ad-hoc* design of current VM schedulers from taking these particularities into account to provide additional energy savings. In this paper, we present Plug4Green, an application that relies on BtrPlace to customize an energy-aware VM scheduler. This flexibility is validated through the implementation of 23 SLA constraints and 2 objectives aiming at reducing either the power consumption or the greenhouse gas emissions. On a heterogeneous test bed, Plug4Green specialization to fit the hardware and the workload specificities allowed to reduce the energy consumption and the gas emission by up to 33% and 34%, respectively. Finally, simulations showed that Plug4Green is capable of computing an improved placement for 7,500 VMs running on 1,500 servers within a minute.

Finally, we started to investigate on easing the jobs of data centre operators using BtrPlace. For example, server maintenance is a common but still critical operation. A prerequisite is indeed to relocate elsewhere the VMs running on the production servers to prepare them for the maintenance. When the maintenance focuses several servers, this may lead to a costly relocation of several VMs so the migration plan must be chose wisely. This however implies to master numerous human, technical, and economical aspects that play a role in the design of a quality migration plan. In[13], we study migration plans that can be decided by an operator to prepare for an hardware upgrade or a server refresh on multiple servers. We exhibit performance bottleneck and pitfalls that reduce the plan efficiency. We then discuss and validate possible improvements deduced from the knowledge of the environment peculiarities.

6.3. Application level

6.3.1. *Simulation Software Architecture*

Participant: Olivier Dalle.

In general purpose software engineering (as opposed to simulation software engineering), the motivations for reuse have long been advocated and demonstrated: lower risks of defects, collective support of potentially larger user community, lower development costs, and so on. In simulation software architectures, we can also cite business-specific motivations, such as providing a better reproducibility of simulation experiments, or avoiding a complex validation process. In [20], we show that although it is rarely discussed, reuse is a problem that may be considered in two opposite directions: reusing and being reused.

6.3.2. *DEVS-based Modeling & Simulation*

Participants: Olivier Dalle, Damian Vicino.

DEVS is a formalism for the specification of discrete-event simulation models, proposed by Zeigler in the 70's, that is still the subject of many research in the simulation community. Surprisingly, the problem of representing the time in this formalism has always been somehow neglected, and most DEVS simulators keep using Floating Point numbers for their arithmetics on time values, which leads to a range of systematic errors, including severe ones such as breaking the causal relations in the model. In [16] we propose a new data type for discretized time representation in DEVS, based on rational numbers. Indeed, we show that rational numbers offer good stability properties for the arithmetics used in DEVS, with a limited impact on the simulation execution performance.

6.3.3. GPU-based High Performance Cloud Computing

Participants: Michael Benguigui, Françoise Baude, Fabrice Huet.

To address HPC, GPU devices are now considered as unavoidable cheap, energy efficient, and very efficient alternative computing units. Our long term goal is to devise some generic solutions in order to incorporate GPU-specific code whenever relevant into a parallel and distributed computation.

As a challenging example, we have pursued our work on pricing American multi-dimensional (so very computation intensive) options in finance. From our previous work that achieved pricing a 40-assets based american option within 8 hours of computation on a single GPU, the work in [9] allows us to reach approximatively one hour of computation time. For this, we run using active objects coupled with OpenCL codes, on 18 GPU nodes acquired from the Grid'5000 platform (the maximum amount of available GPU on Grid'5000 that we could book at once).

Moreover, the balancing of work is taking in consideration the heterogeneous nature of the involved GPUs, and is capable to harness the computing power of multi-core CPUs that also support running OpenCL codes. This parallel and distributed pricing approach is also extended in the forthcoming PhD thesis of Michael Benguigui: it successfully tackles the Value At Risk computation of a portofolio composed of such complex financial products.

6.3.4. Simulation of Software-Defined Networks

Participants: Olivier Dalle, Damian Vicino.

Software Defined Networks (SDN) is a new technology that has gained a lot of attention recently. It introduces programmatic ways to reorganize the network logical topology. To achieve this, the network interacts with a set of controllers, that can dynamically update the configuration of the network routing equipments based on the received events. As often with new network technologies, discrete-event simulation proves to be an invaluable tool for understanding and analzing the performance and behavior of the new systems. In [17], we use such smulations for evaluating the impact of Software-Defined Networks' Reactive Routing on BitTorrent performance. Indeed, BitTorrent uses choking algorithms that continuously open and close connections to different peers. Software Defined Networks implementing Reactive Routing may be negatively affecting the performances of the system under specific conditions because of it lack of knowledge of BitTorrent strategies.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

7.1.1. Amadeus

Duration: May 2014 - April 2015

Inria teams: Scale, Coati

Abstract:This collaboration aims to assess the benefits that digital technologies can bring in complex travel distribution applications. Indeed, these applications require both high performance algorithms and distributed programming methods to search for the best solutions among billions of combinations, in a very short time thanks to the simultaneous use of several hundreds (if not thousands) of computers. These benefits will be demonstrated in an application to build 'of the shelf' optimized packages, fully customized to best meet the complex demands of the traveler.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR Songs

Title: Simulation of Next Generation Systems

Program: Infra 13

Duration: January 2012 - December 2015

Coordinator: Inria (Nancy, Grenoble, Bordeaux)

Others partners: IN2P3 Villeurbanne, LSIIT Strasbourg, I3S Sophia-Antipolis, LINA Nantes

See also: http://infra-songs.gforge.inria.fr/

Abstract: SONGS (2012-2015) is the continuity of SIMGRID project (2009-2012), in the ANR INFRA program. The aim of SONGS is to continue the development of the SimGrid simulation platform for the study of large distributed architectures, including data grids, cloud computing facilities, peer-to-peer applications and HPC/exascale architectures.

8.1.2. FUI CloudForce (now OpenCloudWare)

Program: FSN, labelled by Minalogic, Systematic and SCS.

Duration: January 2012 - September 2015

Coordinator: France-Telecom Research

Others partners: ActiveEon, Armines, Bull, eNovance, eXo Platform, France Telecom (coordinator), Inria, IRIT-INP Toulouse, Linagora, OW2, Peergreen, Télécom Paris Tech, Télécom Saint Etienne, Thales Communications, Thales Services, Université Joseph Fourier, Université de Savoie - LISTIC, UShareSoft

See also: http://www.opencloudware.org/

Abstract: 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.

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) for their deployment, orchestration, performance testing, self-management (elasticity, green IT optimisation), and provisioning. Applications will be deployed potentially on multi IaaS (supporting either one IaaS at a time, or hybrid scenarios).The results of the project will be made available as open source components through the OW2 Open Source Cloudware initiative.

8.1.3. Oseo-Isis Spinnaker

Duration: June 2011 - May 2015

Coordinator: Tagsys-RFID

Others partners: SMEs: Inside-Secure, STIC, Legrand; Academic: IPG, ENS des Mines de St Etienne, Un. du Maine, Un, F. Rabelais Tours, AETS ESEO Angers, Un. Marne la Vallée, Un. Paris 6, Un. Rennes 1, Inria.

See also: http://www.spinnaker-rfid.com/

Abstract: The objective of Spinnaker is to really allow RFID technology to be widely and easily deployed. The role of the OASIS team in this project is to allow the wide scale deployment and management of the specific RFID application servers in the cloud, so to build an end-to-end robust and flexible solution using GCM technology.

8.2. European Initiatives

8.2.1. FP7 & H2020 Projects

8.2.1.1. FI-WARE

Type: COOPERATION

Defi: PPP FI: Technology Foundation: Future Internet Core Platform

Instrument: Integrated Project

Objectif: PPP FI: Technology Foundation:Future Internet Core Platform

Duration: September 2011 - May 2014

Coordinator: Telefonica (Spain)

Others partners: Thales, SAP, Inria

Inria contact: Olivier Festor

See also: http://www.fi-ware.eu/

Abstract: FI-WARE will deliver a novel service infrastructure, building upon elements (called Generic Enablers) which offer reusable and commonly shared functions making it easier to develop Future Internet Applications in multiple sectors. This infrastructure will bring significant and quantifiable improvements in the performance, reliability, and production costs linked to Internet applications, building a true foundation for the future Internet.

8.2.1.2. DC4Cities

Type: COOPERATION

Defi: FP7 Smartcities 2013

Instrument: Specific Targeted REsearch Project

Objectif: ICT-2013.6.2: Data Centers in an energy-efficientand environmentally friendly Internet

Duration: September 2013 - February 2016

Coordinator: Freemind Consulting (BE)

Partners: U. Mannheim (DE), U. Passau (DE), HP Italy Innovation Center (IT), Create-Net (IT), ENEA (IT), CESCA Catalunia (ES), Gas Natural SA (ES), Inst. Munic. Informatica Barcelona (ES), Inria (FR)

Inria contact: Eric Madelaine

See also:

Abstract: Data centres play two different and complementary roles in Smart Cities' energy policies: as ICT infrastructures supporting Smart City resource optimization systems - more in general, delivering ICT services to the citizens - and as large energy consumers. Therefore there are huge expectations on data centres being able to run at the highest levels of renewable energy sources: this is the great challenge of DC4Cities project.

The goal of DC4Cities is to make existing and new data centres energy adaptive, without requiring any modification to the logistics, and without impacting the quality of the services provided to their users. Finally new energy metrics, benchmarks, and measurement methodologies will be developed and proposed for the definition of new related standards. DC4Cities will promote the data centres role as an " eco-friendly" key player in the Smart Cities energy policies, and will foster the integration of a network of local renewable energy providers (also interconnected with local Smart Grids and Micro Grids) to support the pursued increase of renewable energy share.

8.2.2. Collaborations with Major European Organizations

Program: EIT ICTLabs

Project acronym: Data Science programme, Activity 15 327 from Master School action line (MSL)

Project title: EIT ICT Labs Data Science Master

Duration: submitted in 2014, funded from 2014 onwards

Coordinator: Martin Klabbers, Technische Universiteit Eindhoven

Other partners (besides UNS, with Françoise Baude as local coordinator): Univ. Politechnico Madrid, Univ. Trento, Politechnico Milano, Tech. Univ. Berlin, KTH

Abstract: The activity aims to create a new major for the ICT Labs master called "Data Science", with the purpose of breeding a new generation of ICT professionals, equipped with advanced technical and entrepreunarial skills in the key area of data science and data engineering. There is a tremendous demand in industry/society for data scientists, and hence a huge market potential for DS programs. DS positions in the industry requires a different educational program, with next to technical skills, more emphasis on awareness of multifaceted challenges and improving business efficiency based on the challenge outcomes. Expected impact is that DS graduates will be quickly recruited for attractive positions as they can help EU ICT industry achieve a higher rate of innovation successes.

8.3. International Initiatives

8.3.1. Inria International Labs

8.3.1.1. CIRIC Chili

Ciric research line: Telecommunications

Inria principal investigator: Eric Madelaine

Duration: 2012 - 2021

This CIRIC activity is loosely coupled with our SCADA associated team with the Universidad de Chile (UdC). We have had some contacts with a software company in Santiago, and starting exploring some possible collaboration in the area of formal specification of distributed applications for Android systems, and generation of "safe by construction" android code. But the effective involvment of CIRIC manpower in this activity has not yet started.

8.3.1.2. LIAMA Shanghai

Liama project: HADES

Inria principal investigator: Robert de Simone

Oasis researchers involved: Eric Madelaine, Ludovic Henrio

Duration: 2013 - 2016

Modern computing architectures are becoming increasingly parallel, at all levels. Meanwhile, typical applications also display increasing concurrency aspects, specially streaming applications involving data and task parallelism. Cyber physical system interactions also add extra-functional requirements to this high degree of concurrency. The goal of best fitting applications onto architectures becomes a crucial problem, which must be tackled from any possible angle. Our position in the HADES LIAMA project is to consider modeling of applications using formal models of concurrent computation, and specialized model-driven engineering approaches to embody the design flow for such models (analysis, verification, mapping allocation, representation of non-functional properties and constraints). We build on various previous domains of expertise : synchronous languages for embedded system design, asynchronous languages for high-performance cloud computing, and real-time specification languages for cyber-physical interaction aspects.

8.3.2. Inria Associate Teams

8.3.2.1. DAESD

Title: Distributed/Asynchronous, Embedded/synchronous System Development

Inria principal investigator: Eric Madelaine

International Partner (Institution - Laboratory - Researcher):

East China Normal University (ECNU) Shanghai - SEI - Yixiang Chen

Duration: 2012 - 2014

See also: http://team.inria.fr/DAESD

The development of concurrent and parallel systems has traditionally been clearly split in two different families; distributed and asynchronous systems on one hand, now growing very fast with the recent progress of the Internet towards large scale services and clouds; embedded, reactive, or hybrid systems on the other hand, mostly of synchronous behaviour. The frontier between these families has attracted less attention, but recent trends, e.g. in industrial systems, in "Cyber-Physical systems", or in the emerging "Internet of Things", give a new importance to research combining them.

The aim of the DAESD associate team is to combine the expertise of the Oasis/Scale and Aoste teams at Inria, the SEI-Shone team at ECNU-Shanghai, and to build models, methods, and prototype tools inheriting from synchronous and asynchronous models. We plan to address modelling formalisms and tools, for this combined model; to establish a method to analyze temporal and spatial consistency of embedded distributed real-time systems; to develop scheduling strategies for multiple tasks in embedded and distributed systems with mixed constraints.

In 2014, the DAESD associated team co-organized a "Summer School" at ECNU Shanghai.

8.3.2.2. SCADA

Title: Safe Composition of Autonomic Distributed Applications

Inria principal investigator: Ludovic Henrio

International Partner (Institution - Laboratory - Researcher):

University of Chile (Chile) - NIC Chile Research Labs - Javier Bustos

Duration: 2012 - 2014

See also: http://team.inria.fr/scada

The SCADA project aims at promoting the collaboration between NIC LABS (Santiago - Chile) and OASIS team, now SCALE (Inria Sophia Antipolis - France) in the domain of the safe composition of applications. More precisely the project will extend existing composition patterns dedicated to parallel or distributed computing to ease the reliable composition of applications. The strong interactions between formal aspects and practical implementation are a key feature of that project, where formal methods, and language theory will contribute to the practical implementation of execution platforms, development and debugging tools, and verification environments. The composition models we focus on are algorithmic skeletons, and distributed components; and we will particularly focus on the programming and verification of non-functional features. Overall, from formal specification and proofs, this project should lead to the implementation of tools for the design and execution of distributed and parallel applications with a guaranteed behavior.

8.3.3. Inria International Partners

8.3.3.1. Informal International Partners

- Florian Kammuller, Middlesex University.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

- Min Zhang, ECNU Shanghai, from Sep. 25th to Nov. 9th

8.4.1.1. Internships

- Siqi Li, ECNU Shanghai, master internship, from Oct. 15 to Dec. 15th.

8.4.2. Visits to International Teams

- Eric Madelaine visited ECNU Shanghai July. 6-12th.
- Ludovic Henrio, Oleksandra Kulankhina, and Eric Madelaine visited ECNU Shanghai from Nov. 29th to Dec. 6th.

8.4.2.1. Research stays abroad

- Damian Vicino, ARS Laboratory at Carleton University, Ottawa, Canada, January 2014-December 2014 (12 months)

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. general chair, scientific chair

- Eric Madelaine, scientific co-chair of the 11th International Conference on Formal Aspects of Component Software (FACS'14), Bertinoro, Italy, Sep. 2014. http://facs2014.cs.unibo.it/.

9.1.1.2. member of the organizing committee

- Fabien Hermenier was a co-chair of the poster session for Compas.

9.1.2. Scientific events selection

9.1.2.1. responsable of the conference program committee

- Olivier Dalle, member of the SIMUTools Steering Committee;
- Eric Madelaine, member of the FACS Steering Comitee;

9.1.2.2. member of the conference program committee

- Françoise Baude: program committee member of CCGrid'14, ISPA'14, HPCS 2014, Compas2014
- Olivier Dalle: program committee member of DEVS/TMS'14, ACM SIGSIM PADS'14
- Ludovic Henrio: program committee member of ICE'14
- Eric Madelaine: program committee member of EDˆC'14

9.1.2.3. reviewer

- Françoise Baude: reviewer for EuroPar 2014
- Fabien Hermenier: reviewer for the international conference PDP - Special session on energy-aware computing, and the national conference Compas. He was also a sub-reviewer for ICDCS.

9.1.3. Journal

9.1.3.1. member of the editorial board

- Françoise Baude: Concurrency and Computation: Practice and Experience, special issue *2013 Special Issue on High Performance Computing and Simulation*.

9.1.3.2. reviewer

- Françoise Baude was a reviewer for journals Parallel and Distributed Computing, Journal of Computational Finance, and International Journal of Computer Mathematics;
- Olivier Dalle was a reviewer for journals ACM TOMACS, SCS SIMULATION, Elsevier SIMPRA;

- Fabien Hermenier was a reviewer for journals Concurrency and Computation: Practice and Experiment, IEEE Transaction on Cloud Computing, IEEE Transaction on Parallel and Distributed Systems, Journal of Grid Computing;
- Ludovic Henrio was reviewer for the journal SCP (Science of Computer Programming), Mathematical Structures in Computer Science, and STVR;
- Eric Madelaine was reviewer for the journal SCP (Science of Computer Programming), special issue *2013 on Formal Aspects of Component Software*.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence : Françoise Baude, Algorithmique et Programmation en Java, 70 H eqTD, niveau L2, Polytech'Nice Sophia, UNS, France

Licence : Françoise Baude, Introduction au Web (new course setup), 30 H eqTD, niveau L2, Polytech'Nice Sophia, UNS, France

Licence : Olivier Dalle, Systèmes Informatiques, 36 H eqTD, niveau L1, UFR Sciences, UNS, France

Licence : Olivier Dalle, Projet Scientifique Informatique, 33 H eqTD, niveau L2, UFR Sciences, UNS, France

Licence : Olivier Dalle, Systèmes d'Exploitation, 54 H eqTD, niveau L2, UFR Sciences, UNS, France

Licence : Olivier Dalle, Initiation Framework Programmation Serveur, 63 H eqTD, niveau L3, UFR Sciences, UNS, France

Licence : Fabien Hermenier, Algorithmique et Programmation en Java, 70 H eqTD, niveau L2, Polytech'Nice Sophia, UNS, France

Licence : Fabien Hermenier, Introduction à Internet, 103 H eqTD, niveau L2, Polytech'Nice Sophia, UNS, France

Licence : Fabien Hermenier, Outils pour le Génie Logiciel 14 H eqTD, niveauau L3/SI3, Polytech'Nice Sophia, UNS, France

Licence : Justine Rochas, TP de Programmation Orientée Objet 18 H eqTD, niveau L3, UFR Sciences, UNS, France

Master : Françoise Baude, Applications Réparties, 45 H eqTD, niveau M1/SI4, Polytech'Nice Sophia, UNS, France

Master : Françoise Baude and Ludovic Henrio, Distributed systems: an Algorithmic approach, 17 H + 17 h eqTD, niveau M2, Polytech'Nice Sophia/UFR Sciences, UNS, France

Master : Olivier Dalle, Systèmes Distribués, 9 H eqTD, niveau M1, UFR Sciences, UNS, France

Master : Fabrice Huet, Programmation Parallèle et Distribuée, 63h eqTD, niveau M1, UFR SCiences, UNS, France

Master : Fabrice Huet, Programmation Avancée, 78h eqTD, niveau M1, UFR Sciences, UNS, France

Master : Justine Rochas, TP de Systèmes Distribués 24 H eqTD, niveau M1, UFR Sciences, UNS, France

Master : Justine Rochas, TP de Technologies XML 24 H eqTD, niveau M1, UFR Sciences, UNS, France

Summer School : Eric Madelaine, "Designing, programming, and verifying distributed systems", 9 H eqTD, East China Normal University Shanghai, China.

Françoise Baude was Innovation and Entrepreneurship coordinator for the ICT Labs Summer school on Urban Life and Mobility, organized from june 29th to july 11th 2014 at Eurecom (supervision, evaluation and ranking of student teams from an educational perspective).

9.2.2. Supervision

PhD : Laurent Pellegrino, “Pushing dynamic and ubiquitous event-based interaction in the Internet of services: a middleware for event clouds”, UNS, 3/4/2014, Françoise Baude, Fabrice Huet.

PhD: Yanwen Chen, “Formal model and Scheduling for cyberphysical systems”, cotutelle between UNS and ECNU Shanghai, 30/11/2014, advisors Eric Madelaine and Yixiang Chen

PhD: Nuno Gaspar, “Integrated, Autonomic, and Reliable Deployment and Management for SaaS composite applications”, UNS, 16/12/2014, advisor: Eric Madelaine

PhD in progress: Michaël Benguigui, “Modèles de programmation pour l’analyse de données sur machines multi-cœurs, grilles et clouds”, since Dec 2011, advisor Françoise Baude.

PhD in progress: Maeva Antoine, “Plateforme élastique pour le stockage et la notification d’Evènements”, since Oct. 2012, advisors Eric Madelaine and Fabrice Huet

PhD in progress: Sophie Song, “Publish-Subscribe Models for Large Scale Data Analysis Frameworks”, since Feb. 2013, advisor Fabrice Huet and Frédéric Margoulés (ECP)

PhD in progress: Damian Vicino, “Simulation-based Open Science in the Cloud”, since Jan. 2013, advisors Françoise Baude, Gabriel Wainer (Carleton University, Canada), and Olivier Dalle.

PhD in progress : Vincent Kerbache, “Resource Management in a Cloud Supplied with Renewable Energies”, since Sep. 2013, advisors Eric Madelaine and Fabien Hermenier

PhD in progress : Oleksandra Kulankhina, “Model-driven environment for the development of safe component-based distributed applications”, since Oct. 2013, advisor Eric Madelaine

PhD in progress: Justine Rochas, “Programming model and middleware support for distributed and concurrent applications”, since Oct. 2013, advisor Ludovic Henrio

PhD in progress: Hlib Mykhailenko,“Probabilistic Approaches for Big Data Analytics”, since March. 2014, advisors Fabrice Huet, Philippe Nain (Inria)

PhD in progress: Vincenzo Mastandrea, “Deadlock analysis for concurrent and distributed programming”, since Oct. 2014, advisors Ludovic Henrio and Cosimo Laneve.

9.2.3. Juries

- Françoise Baude was jury president for the PhD of Gauthier Berthou (Université de Grenoble), jury member for the PhD of Pierre-Louis Aublin (Université de Grenoble), reviewer and jury member for the PhD of Fawaz Paraiso (Université de Lille 1), of Mohamed Mohamed (INT Telecom Paris-Sud, Université d’Evry), of Soguy Mak-Karé Gueye (Université de Grenoble).
- Ludovic Henrio was jury member for the PhD of Karl Palmskog (KTH); he was reviewer and jury member for the PhD of Thomas Pinsard (Université d’Orléans).
- Fabien Hermenier co-reviewed (with F. Baude) the PhD thesis of Christine Mayap Kamga (University of Toulouse) and an expert for a CIFRE thesis grant.
- Fabrice Huet reviewed the PhD thesis of Roelof Kemp (Vrije Universiteit, Amsterdam).
- Eric Madelaine was reviewer for the PhD of M. Donc Ruzhen, University of Pisa

9.3. Popularization

- Françoise Baude is in charge since September 2011 of the set up and animation of an educational program *Cordées de la Réussite*. She leads as Polytech’Nice Sophia member, and in connection with MASTIC activities at Sophia-Antipolis, two *Cordées pour la Science à Sophia* groups (one with Lycée Vinci Antibes, the other with Lycée Tocqueville Grasse). Since 2013, she is in charge of the AVOSTII-STI2D program implementation for Polytech’Nice Sophia, requiring adverstising high-school pupils for engineering studies, in particular in ICT.

  As ICT Labs Master School University of Nice Sophia-Antipolis coordinator, she is in charge of publicizing, orchestrating, and reporting about all the ICT Labs master programmes involvement of UNS (Internet and Technology Architecture, and Data Science programmes); and as part of the *Paris Node comité de pilotage*, to follow French activities within ICT Labs, and report about UNS activities within this KIC. Overall this mission required about two full-time months of work in 2014.

- Fabrice Huet is in charge, since 2011, of the *Informatiques et Sciences du Numérique* courses for the academia of Nice. These courses are offered to high school teachers who volunteer to offer a Computer Science option to their students. He has also given seminars in high school during the *Fête de la Science*.

10. Bibliography

Major publications by the team in recent years

[1] Q. ZAGARESE, G. CANFORA, E. ZIMEO, I. ALSHABANI, L. PELLEGRINO, F. BAUDE. *Efficient Data-Intensive Event-Driven Interaction in SOA*, in "SAC 2013, 28th ACM Symposium On Applied Computing", March 2013

Publications of the year

Doctoral Dissertations and Habilitation Theses

[2] L. PELLEGRINO. *Pushing dynamic and ubiquitous event-based interactions in the Internet of services : a middleware for event clouds*, Université Nice Sophia Antipolis, April 2014, https://tel.archives-ouvertes.fr/tel-00984262

Articles in International Peer-Reviewed Journals

[3] F. BAUDE, L. HENRIO, C. RUZ. *Programming distributed and adaptable autonomous components–the GCM/ProActive framework*, in "Software: Practice and Experience", May 2014, https://hal.inria.fr/hal-01001043

[4] Y. CHEN, Y. CHEN, E. MADELAINE. *Timed-pNets: a communication behavioural semantic model for distributed systems*, in "Frontiers of Computer Science", November 2014, vol. 8, 24 p. [*DOI :* 10.1007/S11704-014-4096-4], https://hal.inria.fr/hal-01086091

[5] C. DUPONT, F. HERMENIER, T. SCHULZE, R. BASMADJIAN, A. SOMOV, G. GIULIANI. *Plug4Green: A flexible energy-aware VM manager to fit data centre particularities*, in "Ad Hoc Networks", November 2014, pp. 1–16 [*DOI :* 10.1016/J.ADHOC.2014.11.003], https://hal.archives-ouvertes.fr/hal-01086893

[6] Q. ZAGARESE, G. CANFORA, E. ZIMEO, I. ALSHABANI, L. PELLEGRINO, A. ALSHABANI, F. BAUDE. *Improving data-intensive EDA performance with annotation-driven laziness*, in "Science of Computer Programming", January 2015, vol. 97, n^{o} Special Issue on Service-Oriented Architecture and Programming (SOAP), pp. 266-279 [*DOI :* 10.1016/J.SCICO.2014.03.007], https://hal.inria.fr/hal-01094590

International Conferences with Proceedings

[7] M. ANTOINE, F. HUET. *Dealing with Skewed Data in Structured Overlays using Variable Hash Functions*, in "The 15th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)", Hong Kong, Hong Kong SAR China, IEEE Computer Society, December 2014, pp. 42-48 [*DOI :* 10.1109/PDCAT.2014.15], https://hal.inria.fr/hal-01101678

[8] M. ANTOINE, L. PELLEGRINO, F. HUET, F. BAUDE. *A Generic API for Load Balancing in Structured P2P Systems*, in "26th International Symposium on Computer Architecture and High Performance Computing", Paris, France, Computer Architecture and High Performance Computing Workshop (SBAC-PADW), 2014 International Symposium on, IEEE Computer Society, October 2014, pp. 138 - 143 [*DOI :* 10.1109/SBAC-PADW.2014.17], https://hal.inria.fr/hal-01101688

[9] M. BENGUIGUI, F. BAUDE. *Fast American Basket Option Pricing on a multi-GPU Cluster*, in "22nd High Performance Computing Symposium", Tampa, FL, United States, April 2014, pp. 1-8, 8 pages, https://hal.archives-ouvertes.fr/hal-00927482

[10] Y. CHEN, Y. CHEN, E. MADELAINE. *Investigation on Time Properties of Timed-pNets*, in "National Software Application Conference", Guilin, China, November 2014, https://hal.inria.fr/hal-01097783

[11] L. HENRIO, O. KULANKHINA, D. LIU, E. MADELAINE. *Verifying the correct composition of distributed components: Formalisation and Tool*, in "FOCLASA", Rome, Italy, September 2014, https://hal.inria.fr/hal-01055370

[12] L. HENRIO, J. ROCHAS. *Declarative Scheduling for Active Objects*, in "SAC - 29th Symposium On Applied Computing", Gyeongju, South Korea, S. Y. SHIN (editor), ACM, March 2014, pp. 1-6, https://hal.inria.fr/hal-00916293

[13] V. KHERBACHE, E. MADELAINE, F. HERMENIER. *Planning Live-Migrations to Prepare Servers for Maintenance*, in "Euro-Par : Parallel Processing Workshops", Porto, Portugal, August 2014, vol. 8806, n° 0302-9743, pp. 498 - 507 [*DOI :* 10.1007/978-3-319-14313-2_42], https://hal.inria.fr/hal-01096040

[14] G. PABON, L. HENRIO. *Self-Configuration and Self-Optimization Autonomic Skeletons using Events*, in "Programming Models and Applications for Multicores and Manycores", Orlando, United States, Proceedings of Programming Models and Applications on Multicores and Manycores, ACM, February 2014 [*DOI :* 10.1145/2560683.2560699], https://hal.inria.fr/hal-00944294

[15] G. SONG, J. ROCHAS, F. HUET, F. MAGOULÈS. *Solutions for Processing K Nearest Neighbor Joins for Massive Data on MapReduce*, in "23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing", Turku, France, March 2015, https://hal.inria.fr/hal-01097337

[16] D. VICINO, O. DALLE, G. WAINER. *A data type for discretized time representation in DEVS*, in "SIMUTOOLS - 7th International Conference on Simulation Tools and Techniques", Lisbon, Portugal, K. PERUMALLA, R. EWALD (editors), ICST, March 2014, https://hal.inria.fr/hal-01055555

[17] D. VICINO, C.-H. LUNG, G. WAINER, O. DALLE. *Evaluating the impact of Software-Defined Networks' Reactive Routing on BitTorrent performance*, in "FNC - 9th International Conference on Future Networks and Communications", Niagara Falls, Canada, E. M. SHAKSHUKI (editor), Elsevier, August 2014, https://hal.inria.fr/hal-01055556

[18] M. J. DE SOUZA, F. BAUDE. *Distributed Snapshot algorithm for multi-active object-based applications*, in "XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos XV (SBRC) Workshop de Testes e Tolerância a Falhas (WTF)", Florianópolis, Brasil, France, L. ARANTES (editor), May 2014, https://hal.inria.fr/hal-01087336

Conferences without Proceedings

[19] F. HERMENIER, S. DEMASSEY. *BtrPlace: Flexible VM Management in Data Centers*, in "PGMO-COPI'14 Conference on Optimization & Practices in Industry", Paris, France, October 2014, https://hal.inria.fr/hal-01095958

Scientific Books (or Scientific Book chapters)

[20] O. DALLE. *Reuse-centric simulation software architectures*, in "Modeling and Simulation-Based Systems Engineering Handbook", D. GIANNI, A. D'AMBROGIO, A. TOLK (editors), CRC Press, December 2014, pp. 263-292 [*DOI :* 10.1201/B17902-12], https://hal.inria.fr/hal-01111447

[21] S. DEMASSEY, F. HERMENIER, V. KHERBACHE. *Dynamic Packing with Side Constraints for Datacenter Resource Management*, in "Optimized Packings and Their Applications", Springer, January 2015, https://hal.inria.fr/hal-01095949

Research Reports

[22] M. ANTOINE, L. PELLEGRINO, F. HUET, F. BAUDE. *Towards a Generic API for Data Load Balancing in Structured P2P Systems*, July 2014, n° RR-8564, 18 p. , https://hal.inria.fr/hal-01022722

[23] Y. CHEN, Y. CHEN, E. MADELAINE. *Timed-pNets: A Communication Behavioural Semantic Model for Distributed Systems (extended version)*, May 2014, n° RR-8526, 35 p. , https://hal.inria.fr/hal-00988010

[24] L. HENRIO, E. MADELAINE, M. ZHANG. *pNets: an Expressive Model for Parameterised Networks of Processes (Extended Version)*, Inria, November 2014, n° RR-8579, 23 p. , https://hal.inria.fr/hal-01055091

[25] J. ROCHAS, L. HENRIO. *A ProActive Backend for ABS: from Modelling to Deployment*, September 2014, n° RR-8596, https://hal.inria.fr/hal-01065072