Activity Report 2015

Team SCALE

Safe Composition of Autonomous applications with Large-SCALE Execution environment

Inria teams are typically groups of researchers working on the definition of a common project, and objectives, with the goal to arrive at the creation of a project-team. Such project-teams may include other partners (universities or research institutions).

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Distributed Systems and middleware

Table of contents

Team SCALE

Creation of the Team: 2014 January 01, end of the Team: 2015 December 31

Keywords:

Computer Science and Digital Science:

1.1.6. - Cloud
1.1.7. - Peer to peer
1.3. - Distributed Systems
1.6. - Green Computing
2.1.1. - Semantics of programming languages
2.1.7. - Distributed programming
2.4.2. - Verification
2.6.2. - Middleware
2.6.3. - Virtual machines
3.1.3. - Distributed data
3.3.3. - Big data analysis
6. - Modeling, simulation and control
7.1. - Parallel and distributed algorithms
7.12. - Computer arithmetic

Other Research Topics and Application Domains:

4.4.1. - Green computing
6.1.1. - Software engineering

1. Members

Research Scientists

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

Marco Biazzini [Inria, until Apr 2015]
Nassim Jibai [Inria, until Feb 2015]

PhD Students

Maeva Antoine [Univ. Nice, until Sep 2015]
Michael Benguigui [Cifre, until Aug 2015]
Lea El Beze [Univ. Nice]
Vincent Kherbache [Inria]
Oleksandra Kulankhina [Inria]
Vincenzo Mastandrea [Univ. Nice]
Justine Rochas [Univ. Nice]

Ge Song [Centrale Paris]
Damián Vicino [Univ. Nice]

Post-Doctoral Fellow
Pedro Velho [Inria]

Visiting Scientists
Florian Kammuller [Middlesex Univ.]
Yu Feng

Administrative Assistant
Christel Boulard Kozinski [Inria]

Others
Pavlo Khvorostov [Inria, Internship, from Mar 2015 until Aug 2015]
Miguel Oliva [Inria, Internship, from Mar 2015 until Aug 2015]
Mario Taddei [Inria, Internship, from Mar 2015 until Aug 2015]
Alexandros Tsantilas [Inria, Internship, from Mar 2015 until Aug 2015]
Yuanmin Xu [Inria, Internship, from Sep 2015]

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.

In 2014, we were informed that the common team SCALE between Inria and I3S would not be continued. As a consequence, SCALE will be terminated as an Inria team at the end of 2015, and will continue as a standard I3S project.

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.

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

[1] http://akka.io

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

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.

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

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. Highlights of the Year

5.1. Highlights of the Year

Workshops and Conférence organization

- Organisation of a workshop on active object languages in September 2015

6. New Software and Platforms

6.1. BigGrph

- Participants: Eric Madelaine, Fabrice Huet
- Contact: Luc Hogie
- The objective of “biggrph” is to provide a distributed platform for very large graphs processing. A typical data set for testing purpose is a sample of the Twitter graph : 240GB on disk, 398M vertices, 23G edges, average degree of 58 and max degree of 24,635,412.

  We started the project in 2014 with the evaluation of existing middleware (GraphX / Spark and Giraph / Hadoop). After having tested some useful algorithms (written according to the BSP model) we decided to develop our own platform.

The development of the "biggrph" platform is now at the stage where we focus on the quality and the improvement of the code.

In particular we have designed strong test suites and some non trivial bugs have been fixed. We have also solved problems of scalability, in particular concerning the communication layer with billions of messages exchanged between BSP steps. Moreover, we have implemented specific data structures for BSP and support for distributed debugging. This comes along with the implementation of algorithms such as BFS or strongly connected components that are run on the NEF cluster (a facility maintained at Inria Sophia Antipolis).

- This project is a joint work of the three EPs Coati, Diana and Scale and is supported by an ADT grant.
- URL : http://www.i3s.unice.fr/~hogie/software/?name=biggrph

6.2. BtrPlace

FUNCTIONAL DESCRIPTION

BtrPlace dynamically adapts the VM placement depending on pluggable expectations using a network and memory-aware migration scheduler. It currently addresses affinity constraints, resource booking, node state manipulation and hosting restrictions. BtrPlace is a complete rewrite of the reconfiguration algorithm that was inside OW2 project Entropy with a clear focus on extensibility. It embeds the constraint programming library Choco to compute solutions.

BtrPlace has been released 5 times this year. The current code amounts for 44000 lines of Java (production and test code, comments) and the online documentation amounts for around 1500 lines. BtrPlace is available from the Maven central repository. In 2015, it has been downloaded around 480 times from unique IPs and 2400 unique visitors accessed the Website.

- Contact: Fabien Hermenier
- Participants: Fabien Hermenier, Vincent Kherbache
- URL: http://www.btrplace.org/
- APP registration code: IDDN.FR.001.330025.000.S.C.2014.000.31235

6.3. EventCloud

SCIENTIFIC DESCRIPTION

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.

FUNCTIONAL DESCRIPTION

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.

- Participants: Laurent Pellegrino, Fabrice Huet, Françoise Baude, Maeva Antoine and Iyad Alshabani
- Partner: CNRS
- Contact: Françoise Baude

6.4. OSA

Open Simulation Architecture
FUNCTIONAL DESCRIPTION

OSA on 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.

- Participant: Olivier Dalle
- Contact: Olivier Dalle
- URL: http://osa.inria.fr/

6.5. Vercors Component Editor (VCE)

VERification of models for distributed communicating COmponants, with safety and Security
FUNCTIONAL DESCRIPTION

The Vercors tools 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.

- Participants: Eric Madelaine, Antonio Cansado, Ludovic Henrio, Marcela Rivera, Oleksandra Kulankhina, Bartlomiej Szejna, Nassim Jibai and Siqi Li
- Contact: Eric Madelaine
- URL: http://team.inria.fr/scale/software/vercors/

7. New Results

7.1. Programming Languages for Distributed Systems

7.1.1. Multi-active Objects

Participants: Ludovic Henrio, 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 proved the correctness of our 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, and is under submission to a conference. The proof brought us very deep and interesting understanding on the differences between the languages.

- 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). An article is currently submitted to a conference on this subject.
- We are formalising in Isabelle/HOL a first version of the semantics of multiactive objects. This work was done in collaboration with Florian Kammuller
- We organised a workshop on active object languages with the main teams in Europe involved in the development of active-object languages. A journal survey paper on the subject is currently being written.
- We implemented a debugger for multi active object programs.

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

7.1.2. Behavioural Semantics

Participants: Ludovic Henrio, Eric Madelaine, Min Zhang, Siqi Li.

We are conducting a large study on Parameterised Networks of Automata (pNets) from a theoretical perspective. We started last year 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. After publishing those results [13], we focused on open systems and our formalism is able to represent operators of composition of processes, they are represented as hierarchically composed automata with holes and parameters. We defined a semantics for open pNets and a bisimulation theory for them. This study was driven by several usecase examples including a hierarchical broadcast algorithm and several operators of concurrent processes. A short presentation is accepted for publication in the journal "Science China: Information Sciences -". A full paper on the subject of the semantics and bisimulation for open pNets is under submission to a conference.

In parallel we have started the study of a denotational semantics for open pNets, based on the Universal Theory of Processes (UTP). The idea in the long term would be to draw links between the operational, denotational, and algebraic models of the pNet formalism. A short presentation of our preliminary results will be presented at the conference PDP'16 (work in progress session).

7.1.3. GPU-based High Performance Computing for finance

Participants: Michael Benguigui, Françoise Baude.

We have pursued our work on pricing American multi-dimensional (so very computation intensive) options in finance and we hav been able to extend this to the computation of Value At Risk (consists in repeating the American option pricing, but we have found a financial grounded optimization that avoids us to replicate the most time consuming phase).

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. As our scheduling solution is capable to get a reasonable prediction of the workload of each slave computation, we have leveraged this to run the whole pricing and VaR computations onhybrid and heterogeneous clusters. These last results have been incorporated in the PhD thesis of M. Benguigui.

7.1.4. Scalable and robust Middleware for distributed event based computing

Participants: Maeva Antoine, Fabrice Huet, Françoise Baude.

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.

As a distributed system handling huge amount of information, this middleware can suffer from data imbalances. In a journal extension of a previous workshop paper [6], we have enlarged our litterature review of structured peer to peer systems regarding the way they handle load imbalance to the case of distributed big data systems. 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.

7.1.5. Vercors: Integrated environment for verifying and running distributed components

Participants: Ludovic Henrio, Oleksandra Kulankhina, Eric Madelaine.

It is the general prurpose of the Vercors platform to target the generation of distributed applications with safety guarantees. In Vercors, the approach starts from graphical specification formalisms allowing the architectural and behavioral description of component systems. From this point, the user can automatically verify application properties using model-checking techniques. Finally, the specified and verified component model can be translated into executable Java code. The Vercors tool suite is distributed as an Eclipse plugin. This year

- we implemented a first reliable version of the whole tool chain including generation of verifiable models and executable Java code.
- We applied the approach to several examples including Peterson's leader election algorithm, a workflow executor, and the control and management of service composition [7].
- A paper accepted at FASE'2016 presents an overview of this work; a research report provides the full version of the paper [20]. The theoretical background was published as a research report and an improve version is being submitted as a journal paper.

The practical implementation allowed us to improve the presentation of the theory and better evaluate it.

7.2. Run-time/middle-ware level

7.2.1. Virtual Machines Scheduling

Participants: Fabien Hermenier, Vincent Kherbache.

In [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.

We also continued our work on scheduling VM migrations. In [14], [17], we propose a model for VM migration that consider their memory workload and the network topology. This model was then implemented in place of the previous migration scheduler in BtrPlace. Experiments on a real testbed show the new scheduler outperforms state-of-the-art approaches that cap the migration parallelism by a constant to reduce the completion time. Besides an optimal capping, it reduces the migration duration by 20.4% on average and the completion time by 28.1%. In a maintenance operation involving 96 VMs to migrate between 72 servers, it saves 21.5% Joules against the native BtrPlace. Finally, its current library of 6 constraints allows administrators to address temporal and energy concerns, for example to adapt the schedule and fit a power budget.

Finally, in [10] we transfer the principles of using Constraint Programming to propose a multi-objective job placement algorithm devoted to High Performance Computing (HPC). One of the key decisions made by both MapReduce and HPC cluster management frameworks is the placement of jobs within a cluster. To make this decision, they consider factors like resource constraints within a node or the proximity of data to a process. However, they fail to account for the degree of collocation on the cluster's nodes. A tight process placement can create contention for the intra-node shared resources, such as shared caches, memory, disk, or network bandwidth. A loose placement would create less contention, but exacerbate network delays and increase cluster-wide power consumption. Finding the best job placement is challenging, because among many possible placements, we need to find one that gives us an acceptable trade-off between performance and power consumption. We then propose to tackle the problem via multi-objective optimization. Our solution is able to balance conflicting objectives specified by the user and efficiently find a suitable job placement.

7.3. Application level

7.3.1. 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 [15] we propose simulation algorithms, based on the Discrete Event System Specification (DEVS) formalism, that can be used to simulate and obtain every possible output and state trajecto-ries of simulations that receive input values with uncertainty quantification. Then, we present a subclass of DEVS models, called Finite Forkable DEVS (FF-DEVS), that can be simulated by the proposed algorithms. This subclass ensures that the simulation is forking only a finite number of processes for each simulation step. Finally, we discuss the simulation of a traffic light model and show the trajectories obtained when it is subject to input uncertainty.

We have also worked on improving the simulation of DEVS models in some particular situations[16]. Parallel Discrete Event System Specification (PDEVS), for example, is a well-known formalism used to model and simulate Discrete Event Systems. This formalism uses an abstract simulator that defines a set of abstract algorithms that are parallel by nature. To implement simulators using these abstract algorithms , several architectures were proposed. Most of these architectures follow distributed approaches that may not be appropriate for single core processors or microcontrollers. In order to reuse efficiently PDEVS models in this type of systems, we define a new architecture that provides a single threaded execution by passing messages in a call/return fashion to simplify the execution time analysis.

This work has also been presented and defended in the PhD Thesis of D. Vicino[5].

7.3.2. 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 [8], 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.

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 - September 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. 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 Digital

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

Project title: EIT Digital Data Science Master

Duration: submitted in 2014, funded from 2014 onwards

Coordinator: Farideh Heidari, 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 has successfully launched a new major for the EIT Digital KIC 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 (now terminated) SCADA associated team with the Universidad de Chile (UdC). We have some research collaboration with our chilean colleagues, in particular on new usages of the GCM component model for cloud management[7].

8.3.1.2. LIAMA Shanghai

Liama project: CASCADES

Inria principal investigator: Vania Joloboff

SCALE researchers involved: Eric Madelaine, Ludovic Henrio,

AOSTE researchers: Robert de Simone, Julien DeAntoni, Frederic Mallet

International Partner (Institution - Laboratory - Researcher):

East China Normal University (ECNU) Shanghai - Software Engineering Institue - MOE International Lab of Trustworthy Software : Jifeng HE, Changbo WANG, Huibiao ZHU, Min ZHANG, Yixiang CHEN.

Duration: 2016 - 2017

The SACCADES project aims at improving the development of reliable cyber physical systems and more generally of distributed systems combining asynchronous with synchronous aspects, with different but complementary angles:

Develop the theoretical support for Models of Computations and Communications (MoCCs) that are the fundamentals basis of the tools. Develop software tools (a) to enable the development and verification of executable models of the application software, which may be local or distributed and (b) to define and optimize the mapping of software components over the available resources. Develop virtual prototyping technology enabling the validation of the application software on the target hardware platform.

The Scale team is involved in particular, with our chinese partners, on studies of semantics and compositional properties, and on the development of software tools supporting the Model-Driven Engineering approaches.

This LIAMA project is tighly linked with our FM4CPS Associated team (Inria principal investigator: Robert de Simone, SCALE participants: Eric Madelaine, Ludovic Henrio, Oleksandra Kulankhina).

8.3.2. Inria International Partners

8.3.2.1. Informal International Partners

- Advanced Real-Time Simulation Laboratory, Carleton University, Ottawa Canada: collaboration on simulation methodology, the DEVS formalism, and SDN Networks [16], [8], [15]

8.4. International Research Visitors

8.4.1. Visits of International Scientists

8.4.1.1. Internships

- Alexandros Tsantilas. *Fuzzing a VM Scheduler*. Co-supervised by Ludovic Henrio. Master 2 Ubinet 2014-15
- Wafa Khlif. *How sustainable data centres can be ?* Co-supervised by Fabien Hermenier. Master 2 IFI 2014-15
- Mario Taddei. *An integrated system for building and running reproducible research*. Supervised by Olivier Dalle. Master 2 Ubinet, 2014-2015;

8.4.1.2. Research stays abroad

- Olivier Dalle visited Carleton University (Ottawa, Canada) for one month (Dec 2015 - Jan 2016)
- Eric Madelaine visited East China Normal University in Shanghai for 3 weeks (July and November)

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. general chair, scientific chair

- Ludovic Henrio, member of the chairing committee of ICE'2016; organisation of a workshop on Active object languages in September 2015.
- Fabien Hermenier co-chaired of the 4th Workshop Energy-Efficiency in Data Centres (E2DC)

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 Comittee;

9.1.2.2. member of the conference program committee

- Françoise Baude: program committee member of HPCS 2015, Compas2015
- Olivier Dalle: program committee member of DEVS/TMS'15, ACM SIGSIM PADS'15, WSC'15
- Ludovic Henrio: program committee member of ICE'15, PDP4PAD 2016, AGERE!2015
- Fabien Hermenier: program committee member of Compas2015, PDP, VTDC
- Fabrice Huet : program committee member of Europar 2014, IEEE Big Data 2014, BDCloud 2014.

9.1.3. Journal

9.1.3.1. reviewer

- Françoise Baude was a reviewer for International Journal of Computer Mathematics, and TSI special issue of Compas 2014;
- Olivier Dalle was a reviewer for journals ACM TOMACS, SCS SIMULATION, Elsevier SIMPRA;
- Fabien Hermenier was a reviewer for journals of Systems and Software, IEEE Transactions on Cloud Computing, Journal of Grid Computing;
- Ludovic Henrio was reviewer for the journal SCP (Science of Computer Programming);
- Eric Madelaine was reviewer for the journals SCP (Science of Computer Programming) and JLAMP (Journal of Logical and Algebraic Methods in Programming).

9.1.4. Invited talks

- Eric Madelaine presented a talk titled “Semantics and Equivalences for Parameterized Networks of Synchronised Automata (pNets)”, at the “Advanced Software Technology Forum” of the Chinese Academy of Science, Shanghai, July 23rd.

9.1.5. Research administration

- Eric Madelaine is council member of the International Joint Research Laboratory of Trustworthy Software, Ministery of Education, China.

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, 30 H eqTD, niveau L2, Polytech'Nice Sophia, UNS, France

Licence : Olivier Dalle, Systèmes Informatiques, 72 H eqTD, niveau L1, UFR Sciences, UNS, France

Licence : Olivier Dalle, Outils Formels Informatique, 36 H eqTD, niveau L2, 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, 72 H eqTD, niveau L2, 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, 36 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 : 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 : Fabien Hermenier, Software Architecture for Cloud Computing, 17h eqTD, niveau M2, Polytech'Nice Sophia, UNS, France

Master : Fabien Hermenier, Virtualised Architecture for Cloud Computing, 17h eqTD, niveau M2, Polytech'Nice Sophia, 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.

9.2.2. Supervision

PhD : Michaël Benguigui, "Valorisation d'options américaines et Value at Risk de portefeuille sur cluster de GPUs/CPUs hétérogène", UNS, 27/8/2015 advisor Françoise Baude.

PhD : Maeva Antoine, "Improving skewed data dissemination in structured overlays", UNS, 23/3/2015, advisors Eric Madelaine and Fabrice Huet

PhD: Damian Vicino, "Improved Time Representation in Discrete-Event Simulation", UNS-Carleton, 23/11/2015, advisors Olivier Dalle, Françoise Baude, Gabriel Wainer (Carleton University, Canada).

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

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.

PhD in progress: Léa El Bèze, “Predictive Big Data Analytics: continuous and dynamically adaptable data request and processing”, since Nov. 2015, advisors Françoise Baude and Didier Parigot (Inria).

9.2.3. Juries

- Françoise Baude was jury president for the PhD of Ivan Paez (Université de Rennes 1.
- Ludovic Henrio was reviewer and jury member for the PhD of Sylvain Dailler (Université d’Orléans).
- Fabrice Huet reviewed the PhD thesis of Roelof Kemp (Vrije Universiteit, Amsterdam).

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*, which in 2015 has launched a robotic project around the humanoid robot named Poppy. 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 EIT Digital Master School University of Nice Sophia-Antipolis coordinator, she is in charge of publicizing, orchestrating, and reporting about all the EIT Digital 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 EIT Digital, and report about UNS activities within this KIC. Overall this mission required about four full-time months of work in 2015. She presented these two master programs at the Telecom Valley Innovation meeting on March 27th and June 18 2015, and in more details at the Kick-Off meeting of the Master School organized in Eindhoven for master students, on October 23th 2015.

  She has been invited at the round table “Mystérieuses Big Data” on november 230th that was organized for the 50th anniversary of UNS.
- 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 *FÃªte de la Science*.

10. Bibliography

Major publications by the team in recent years

[1] B. Sergey, A. Fedorova, V. Evgeny, D. Tyler, F. Hermenier. *Multi-objective job placement in clusters*, in "International Conference for High Performance Computing, Networking, Storage and Analysis", Austin, United States, November 2015 [*DOI :* 10.1145/2807591.2807636], https://hal.inria.fr/hal-01279724

[2] 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

[3] M. Antoine. *Improving skewed data dissemination in structured overlays*, Université Nice Sophia Antipolis, September 2015, https://tel.archives-ouvertes.fr/tel-01245077

[4] M. BENGUIGUI. *American option pricing and computation of the portfolio Value at risk on heterogeneous GPU-CPU cluster*, Université Nice Sophia Antipolis, August 2015, https://tel.archives-ouvertes.fr/tel-01204580

[5] D. VICINO. *Improved Time Representation in Discrete-Event Simulation*, Université Nice Sophia Antipolis ; Carleton University (CA), November 2015, https://tel.archives-ouvertes.fr/tel-01276128

Articles in International Peer-Reviewed Journals

[6] M. ANTOINE, L. PELLEGRINO, F. HUET, F. BAUDE. *A generic API for load balancing in distributed systems for big data management*, in "Concurrency and Computation: Practice and Experience", August 2015 [*DOI :* 10.1002/CPE.3646], https://hal.archives-ouvertes.fr/hal-01273083

[7] T. AUBONNET, L. HENRIO, S. KESSAL, O. KULANKHINA, F. LEMOINE, E. MADELAINE, C. RUZ, N. SIMONI. *Management of service composition based on self-controlled components*, in "Journal of Internet Services and Applications", 2015, vol. 6, n^{o} 15, 17 p. [*DOI :* 10.1186/S13174-015-0031-7], https://hal.inria.fr/hal-01180627

[8] D. VICINO, C.-H. LUNG, G. WAINER, O. DALLE. *An Investigation on Software-Defined Networks' Reactive Routing against BitTorrent*, in "IET Networks", 2015, vol. 4, n^{o} 5, pp. 249-254 [*DOI :* 10.1049/IET-NET.2014.0105], https://hal.inria.fr/hal-01274250

[9] 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

[10] S. BLAGODUROV, A. FEDOROVA, V. EVGENY, D. TYLER, F. HERMENIER. *Multi-objective job placement in clusters*, in "International Conference for High Performance Computing, Networking, Storage and Analysis", Austin, United States, ACM (editor), November 2015 [*DOI :* 10.1145/2807591.2807636], https://hal.inria.fr/hal-01279724

[11] C. DUPONT, M. SHEIKHALISHAHI, F. HERMENIER, F. FACCA. *An energy aware application controller for optimizing renewable energy consumption in Cloud computing*, in "8th IEEE/ACM International Conference on Utility and Cloud Computing", Limassol, Cyprus, December 2015, https://hal.inria.fr/hal-01282260

[12] N. GASPAR, L. HENRIO, E. MADELAINE. *Painless support for static and runtime verification of component-based applications*, in "Fundamentals of Software Engineering (FSEN'2015)", Teheran, Iran, April 2015, 15 p. , https://hal.inria.fr/hal-01168757

[13] L. HENRIO, E. MADELAINE, M. ZHANG. *pNets: an Expressive Model for Parameterised Networks of Processes*, in "Formal Approaches to Parallel and Distributed Systems (4PAD)-Special Session of Parallel, Distributed and network-based Processing (PDP)", Turku, Finland, 2015, https://hal.inria.fr/hal-01139432

[14] V. KHERBACHE, E. MADELAINE, F. HERMENIER. *Scheduling Live-Migrations for Fast, Adaptable and Energy-Efficient Relocation Operations*, in "8th IEEE/ACM International Conference on Utility and Cloud Computing", Limassol, Cyprus, December 2015 [*DOI :* 10.13140/RG.2.1.1626.8563], https://hal.inria.fr/hal-01271684

[15] D. VICINO, O. DALLE, G. WAINER. *Using Finite Forkable DEVS for Decision-Making Based on Time Measured with Uncertainty*, in "Eighth EAI International Conference on Simulation Tools and Techniques (SIMUTOOLS 2015)", Athènes, Greece, August 2015 [*DOI :* 10.4108/EAI.24-8-2015.2261152], https://hal.inria.fr/hal-01274200

[16] D. VICINO, D. NIYONKURU, G. WAINER, O. DALLE. *Sequential PDEVS Architecture*, in "Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium", Alexandria, Virginia, United States, 2015, https://hal.inria.fr/hal-01274279

National Conferences with Proceedings

[17] V. KHERBACHE, E. MADELAINE, F. HERMENIER. *Ordonnancement contrôlé de migrations à chaud*, in "Compas'15", Lille, France, July 2015 [*DOI :* 10.13140/RG.2.1.5100.2649], https://hal.inria.fr/hal-01271676

Conferences without Proceedings

[18] 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, Finland, March 2015, https://hal.inria.fr/hal-01097337

Scientific Books (or Scientific Book chapters)

[19] S. DEMASSEY, F. HERMENIER, V. KHERBACHE. *Dynamic Packing with Side Constraints for Datacenter Resource Management*, in "Optimized Packings with Applications", G. FASANO, J. D. PINTÉR (editors), Springer Optimization and Its Applications, Springer International Publishing, 2015, vol. 105, pp. 19-35, Chapter 2 [*DOI :* 10.1007/978-3-319-18899-7_2], https://hal-mines-paristech.archives-ouvertes.fr/hal-01224870

Research Reports

[20] L. HENRIO, O. KULANKHINA, S. LI, E. MADELAINE. *Integrated environment for verifying and running distributed components -Extended version*, Inria Sophia-Antipolis, December 2015, n^{o} RR-8841, 24 p. , https://hal.inria.fr/hal-01252323

Other Publications

[21] M. K. GURAYA, R. SINGH BAJWA, D. VICINO, C.-H. LUNG. *The assessment of BitTorrent's performance using SDN in a Mesh topology*, September 2015, The 6th International Conference On Network of the Future, Poster [*DOI :* 10.1109/NOF.2015.7333294], https://hal.inria.fr/hal-01274346

[22] V. KHERBACHE, E. MADELAINE, F. HERMENIER. *Memory and Network Aware Scheduling of Virtual Machine Migrations*, April 2015, EuroSys'15, Poster [*DOI :* 10.13140/RG.2.1.2740.9681], https://hal.inria.fr/hal-01271665

[23] P. KHVOROSTOV. *A viewer tool for multi-active object*, Universite Nice Sophia Antipolis, August 2015, https://hal.inria.fr/hal-01212170