IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2014

Project-Team ALGORILLE

Algorithms for the Grid

IN COLLABORATION WITH: Laboratoire lorrain de recherche en informatique et ses applications (LORIA)

RESEARCH CENTER
Nancy - Grand Est

THEME
Distributed and High Performance Computing

Table of contents

Project-Team ALGORILLE

Keywords: Distributed System, Parallel Algorithms, Performance, Experimentation, High Performance Computing, Simulation

Creation of the Project-Team: 2007 January 01*, updated into Team:* 2015 January 01.

1. Members

Research Scientist
Jens Gustedt [Inria, Senior Researcher, HdR]

Faculty Members
Martin Quinson [Team leader, Univ. Lorraine, Associate Professor, HdR]
Sylvain Contassot-Vivier [Univ. Lorraine, Professor, HdR]
Lucas Nussbaum [Univ. Lorraine, Associate Professor, on partial leave to Inria since September 2013]

Engineers
Paul Bédaride [Univ. Lorraine, until Sep 2014]
Gabriel Corona [Univ. Lorraine]
Emmanuel Jeanvoine [Inria, permanent SED engineer delegated to AlGorille]
Jérémie Gaidamour [Inria, from Oct 2014]
Phillippe Kalitine [Univ. Lorraine, from May 2014 until June 2015]
Stéphane Martin [Inria, until Oct 2014]
Émile Morel [Inria, until Sep 2014]
Matthieu Nicolas [Inria, from Apr 2014]
Cristian Ruiz [Inria, from Nov 2014]
Luc Sarzyniec [Inria, until Jun 2014]
Arthur Garnier [Inria, TELECOM Nancy, élève-ingénieur en apprentissage]

PhD Students
Marion Guthmuller [Univ. Lorraine, from Oct 2011 to March 2015]
Tomasz Buchert [Inria, Inria grant from Oct 2011 to Aug 2014, Univ. Lorraine (ATER), since Sep 2014]
Mariem Saied [Inria, until Oct 2016]

Post-Doctoral Fellows
Rajni Aron [Inria, until Jun 2014]
Joseph Emeras [Inria, until Feb 2014]

Administrative Assistants
Delphine Hubert [Univ. Lorraine]
Martine Kuhlmann [CNRS]
Céline Simon [Inria]

Other
Raphael Rieu Helft [Inria, Student, internship from Jun 2014 until Aug 2014]

2. Overall Objectives

2.1. Introduction

The possible access to computing resources over the Internet allows a new type of applications that use the power of the machines and the network. The transparent and efficient access to these parallel and distributed resources is one of the major challenges of information technology. It needs the implementation of specific techniques and algorithms to make heterogeneous processing elements communicate with each other, let applications work together, allocate resources and improve the quality of service and the security of the transactions.

Figure 1. Layered model of a grid architecture

Given the complex nature of these platforms, software systems have to rely on a layered model. Here, as a specific point of view for our project we will distinguish four layers as they are illustrated in Figure 1. The *infrastructure* encompasses both hardware and operating systems. *Services* abstract infrastructure into *functional units* (such as resource and data management, or authentication) and thus allow to cope with the heterogeneity and distribution of the infrastructure.

Services form grounding bricks that are aggregated into *middlewares*. Typically one particular service will be used by different middlewares, thus such a service must be sufficiently robust and generic, and the access to it should be standardized. Middlewares then offer a software infrastructure and programming model (data-parallel, client/server, peer-to-peer, *etc.*) to the user *applications*. Middlewares may be themselves generic (*e.g.,*Globus), specialized to specific programming models (*e.g.,*message passing libraries) or specific to certain types of applications.

2.2. Challenges

To our opinion the algorithmic challenges of such systems are located at the *application* and *service* layers. In addition to these two types of challenges, we identify a third one which consists in the evaluation of models, algorithms and implementations. To summarize, the three research areas that we address are:

applications: We have to organize the application and its access to the middleware in a way that is convenient for both. The application should restrict itself to a sensible usage of the middleware and make the least assumptions about the other underlying (and hidden) layers.

services: The service layer has to organize the infrastructure in a convenient way such that resources are used efficiently and such that the applications show a good performance.

performance (and correctness) evaluation: To assert the quality of computational models and algorithms that we develop within such a paradigm, we have to compare algorithms and program executions amongst each other. A lot of challenges remain in the reproducibility of experiments and in the extrapolation to new scales in the number of processors or the input data size. Traditionally, the application performance was the main concern in our domain while application correctness was seen as a simpler issue, to be solved through testing. This is not true anymore because of the scale reached by modern applications, mandating formal assessment methods of the application correction.

2.3. Approach

So, our approach emphasizes on **algorithmic** and **engineering aspects** of such computations on all scales. In particular it addresses the problems of organizing the computation **efficiently**, be it on the side of a service provider or within an application program.

To assert the quality and validity of our results, the inherent complexity of the interplay of platforms, algorithms and programs imposes a strong emphasis on **experimental methodology**. Our research is structured in three different themes:

- *Structuring of applications for scalability*: modeling of size, locality and granularity of computation and data.
- *Transparent resource management*: sequential and parallel task scheduling, migration of computations, data exchange, distribution and redistribution of data.
- *Experimental validation and methodology*: reproducibility, extendability and applicability of formal assessments, simulations, emulations and *in situ* experiments.

An important goal of the project is to increase the cross-fertility between these different themes and their respective communities and thus to allow the scaling of computations for new forms of applications, reorganize platforms and services for economic use of resources, and to endow the scientific community with foundations, software and hardware for conclusive and reproducible experiments.

3. Research Program

3.1. Structuring Applications

Computing on different scales is a challenge under constant development that, almost by definition, will always try to reach the edge of what is possible at any given moment in time: in terms of the scale of the applications under consideration, in terms of the efficiency of implementations and in what concerns the optimized utilization of the resources that modern platforms provide or require. The complexity of all these aspects is currently increasing rapidly.

3.1.1. Diversity of platforms

Design of processing hardware is diverging in many different directions. Nowadays we have SIMD registers inside processors, on-chip or off-chip accelerators (many-core boards, GPU, FPGA, vector-units), multi-cores and hyperthreading, multi-socket architectures, clusters, grids, clouds... The classical monolithic architecture of one-algorithm/one-implementation that solves a problem is obsolete in many cases. Algorithms (and the software that implements them) must deal with this variety of execution platforms robustly.

As we know, the "*free lunch*" for sequential algorithms provided by the increase of processor frequencies is over, we have to go parallel. But the "*free lunch*" is also over for many automatic or implicit adaptation strategies between codes and platforms: e.g the best cache strategies can't help applications that access memory randomly, or algorithms written for "simple" CPU (von Neumann model) have to be adapted substantially to run efficiently on vector units.

3.1.2. The communication bottleneck

Communication and processing capacities evolve at a different pace, thus the *communication bottleneck* is always narrowing. An efficient data management is becoming more and more crucial.

Not many implicit data models have yet found their place in the HPC domain, because of a simple observation: latency issues easily kill the performance of such tools. In the best case, they will be able to hide latency by doing some intelligent caching and delayed updating. But they can never hide the bottleneck for bandwidth. An efficient solution to this problem is the use of asynchronism in the algorithms. However, until now its application has been limited to iterative processes with specific constraints over the computational scheme.

HPC was previously able to cope with the communication bottleneck by using an explicit model of communication, namely MPI. It has the advantage of imposing explicit points in code where some guarantees about the state of data can be given. It has the clear disadvantage that coherence of data between different participants is difficult to manage and is completely left to the programmer.

Here, our approach is and will be to timely request explicit actions (like MPI) that mark the availability of (or need for) data. Such explicit actions ease the coordination between tasks (coherence management) and allow the platform underneath the program to perform a pro-active resource management.

3.1.3. Models of interdependence and consistency

Interdependence of data between different tasks of an application and components of hardware will be crucial to ensure that developments will possibly scale on the ever diverging architectures. We have up to now presented such models (PRO, DHO, ORWL) and their implementations, and proved their validity for the context of SPMD-type algorithms.

Over the next years we will have to enlarge the spectrum of their application. On the algorithm side we will have to move to heterogeneous computations combining different types of tasks in one application. Concerning the architectures, we will have to take into account the fact of increased heterogeneity, processors of different speeds, multi-cores, accelerators (FPU, GPU, vector units), communication links of different bandwidth and latency, memory and generally storage capacity of different size, speed and access characteristics. First implementations using ORWL in that context look particularly promising.

The models themselves will have to evolve to be better suited for more types of applications, such that they allow for a more fine-grained partial locking and access of objects. They should handle *e.g.* collaborative editing or the modification of just some fields in a data structure. This work has already started with DHO which allows the locking of *data ranges* inside an object. But a more structured approach would certainly be necessary here to be usable more comfortably in most applications.

3.1.4. Frequent I/O

A complete parallel application includes I/O of massive data, at an increasing frequency. In addition to applicative input and output data flows, I/O are used for checkpointing or to store traces of execution. These then can be used to restart in case of failure (hardware or software) or for a post-mortem analysis of a chain of computations that led to catastrophic actions (for example in finance or in industrial system control). The difficulty of frequent I/O is more pronounced on hierarchical parallel architectures that include accelerators with local memory.

I/O have to be included in the design of parallel programming models and tools. The ORWL library (Ordered Read-Write Lock) should be enriched with such tools and functionalities, in order to ease the modeling and development of parallel applications that include data IO, and to exploit most of the performance potential of parallel and distributed architectures.

3.1.5. Algorithmic paradigms

Concerning asynchronous algorithms, we have studied different variants of asynchronous models and developed several versions of implementations, allowing us to precisely study the impact of our design choices. However, we are still convinced that improvements are possible in order to extend the applicability of asynchronism, especially concerning the control of its behavior and the termination detection (global convergence of iterative algorithms). We have proposed some generic and non-intrusive way of implementing such a procedure in any parallel iterative algorithm.

3.1.6. Cost models and accelerators

We have already designed some models that relate computation power and energy consumption. Our present works in this topic concern the design and implementation of an auto-tuning system that controls the application according to user defined optimization criteria (computation and/or energy performance). This implies the insertion of multi-schemes and/or multi-kernels into the application such that it will be able to adapt its behavior to the requirements.

3.1.7. Design of dynamical systems for computational tasks

In the context of a collaboration with Nazim Fatès over dynamical systems, and especially cellular automata, we address a new way to study dynamical systems, that is more development oriented than analysis oriented. In fact, until now, most of the studies related to dynamical systems consisted in analyzing the dynamical properties (convergence, fixed points, cycles, initialization,...) of some given systems, and in describing the emergence of complex behaviors. Here, we focus on the dual approach that consists in designing dynamical systems in order to fulfill some given tasks. In this approach, we consider both theoretical and practical aspects.

3.2. Transparent Resource Management for Clouds

Given the extremely large offer of resources by public or private clouds, users need software assistance to make provisioning decisions. Our goal is to design a **cloud resource broker** which handles the workload of a user or

of a community of users as a multi-criteria optimization problem. The notions of resource usage, scheduling, provisioning and task management have been adapted to this new context. For example, to minimize the makespan of a DAG of tasks, usually a fixed number of resources is assumed. On IaaS clouds, the amount of resources can be provisioned at any time, and hence the scheduling problem must be redefined using one new prevalent optimization criterion: the financial cost of the computation.

3.2.1. Provisioning strategies

The provisioning strategies are hence central to the broker. They are designed after heuristics which aim to fit execution constraints and satisfy user preferences. For instance, lowering the costs can be achieved with strategies aiming at reusing already leased resources, or switch to less powerful and cheaper resources. However, some economic models proposed by cloud providers involve a complex cost-benefit analysis which we plan to address. Moreover, these economic models incur additional costs, *e.g.* for data storage or transfer, which have to be taken into account to design a comprehensive broker.

3.2.2. User workload analysis

Another possible extension of the capability of such a broker is the analysis of user workloads. Characterizing the workload might help to anticipate the behavior of each alternative provisioning strategy. The objective is to allow the user to select the suitable provisioning solution thanks to concrete information, such as completion time and financial cost.

3.2.3. Simulation of cloud platforms

Providing concrete information about provisioning solutions can also be achieved through simulation. Although predicting the behavior of applicative cases in real grid environment is made very difficult by the shared (*e.g.* multi-tenant), heterogeneous and dynamic nature of the resources, cloud resources (i.e. VMs) are perceived as reserved and homogeneous and stable by the end-user. Therefore, proposing an accurate prediction of the different strategies through an accurate simulation process would be a strong decision support for the user.

3.3. Experimental Methodologies for the Evaluation of Distributed Systems

Distributed systems are very challenging to study, test, and evaluate. Computer scientists traditionally prefer to study their systems *a priori* by reasoning theoretically on the constituents and their interactions. But the complexity of large-scale distributed systems makes this methodology near to impossible, explaining that most of the studies are done *a posteriori* through experiments.

In AlGorille, we strive at designing a comprehensive set of solutions for experimentation on distributed systems by working on several methodologies (formal assessment, simulation, use of experimental facilities, emulation) and by leveraging the convergence opportunities between methodologies (co-development, shared interfaces, validation combining several methodologies).

3.3.1. Simulation and Dynamic Verification

Our team plays a key role in the SimGrid project, a mature simulation toolkit widely used in the distributed computing community. Since more than ten years, we work on the validity, scalability and robustness of our tool.

Our current medium term goal is to extend the tool applicability to **Clouds and Exascale systems**. In the last years, we therefore worked toward disk and memory models in addition to the previously existing network and CPU models. The tool's scalability and efficiency also constitutes a permanent concern to us. **Interfaces** constitute another important work axis, with the addition of specific APIs on top of our simulation kernel. They provide the "syntactic sugar" needed to express algorithms of these communities. For example, virtual machines are handled explicitly in the interface provided for Cloud studies. Similarly, we pursue our work on an implementation of the full MPI standard allowing to study real applications using that interface. This work may also be extended in the future to other interfaces such as OpenMP or OpenCL.

We integrated a model checking kernel in SimGrid to enable **formal correctness studies** in addition to the practical performance studies enabled by simulation. Being able to study these two fundamental aspects of distributed applications within the same tool constitutes a major advantage for our users. In the future, we will enforce this capacity for the study of correctness and performance such that we hope to tackle their usage on real applications.

3.3.2. Experimentation on testbeds and production facilities, emulation

Our work in this research axis is meant to bring major contributions to the **industrialization of experimentation** on parallel and distributed systems. It is structured through multiple layers that range from the design of a testbed supporting high-quality experimentation, to the study of how stringent experimental methodology could be applied to our field, as depicted in Figure 2.

During the last years, we have played a **key role in the design and development of Grid'5000** by leading the design and technical developments, and by managing several engineers working on the platform. We pursue our involvement in the design of the testbed with a focus on ensuring that the testbed provides all the features needed for high-quality experimentation. We also collaborate with other testbeds sharing similar goals in order to exchange ideas and views. We now work on **basic services supporting experimentation** such as resources verification, management of experimental environments, control of nodes, management of data, etc. Appropriate collaborations will ensure that existing solutions are adopted to the platform and improved as much as possible.

One key service for experimentation is the ability to alter experimental conditions using emulation. We work on the **Distem emulator**, focusing on its validation and on adding features (such as the ability to emulate faults, varying availability, churn, load injection, etc) and investigate if altering memory and disk performance is possible. Other goals are to scale the tool up to 20000 virtual nodes while improving the tool usability and documentation.

We work on **orchestration of experiments** in order to combine all the basic services mentioned previously in an efficient and scalable manner, with the design of a workflow-based experiment control engine named **XPFlow**.

3.3.3. Convergence and co-design of experimental methodologies

We see the experimental methodologies we work on as steps of a common experimental staircase: ideally, **one could and should leverage the various methodologies to address different facets of the same problem**. To facilitate that, we must co-design common or compatible formalisms, semantics and data formats.

Other experimental sciences such as biology and physics have paved the way in terms of scientific methodology. We **should learn from other experimental sciences, adopt good practices and adapt them** to Computer Science's specificities.

But Computer Science also has specific features that make it the ideal field to **create a truly Open Science**: provide infrastructure and tools for publishing and reproducing experiments and results, linked with our own methodologies and tools.

Finally, one important part of our work is to maintain a deep understanding of systems and their environments, in order to properly model them and experiment on them. Similarly, we need to understand the emerging scientific challenges in our field in order to improve adequately our experimental tools.

4. Application Domains

4.1. Promoting parallelism in applications

In addition to direct contributions within our own scientific domain, numerous collaborations have permitted us to test our algorithmic ideas in connection with academics of different application domains and through our association with SUPELEC with some industrial partners: physics, geology, biology, medicine, machine learning or finance.

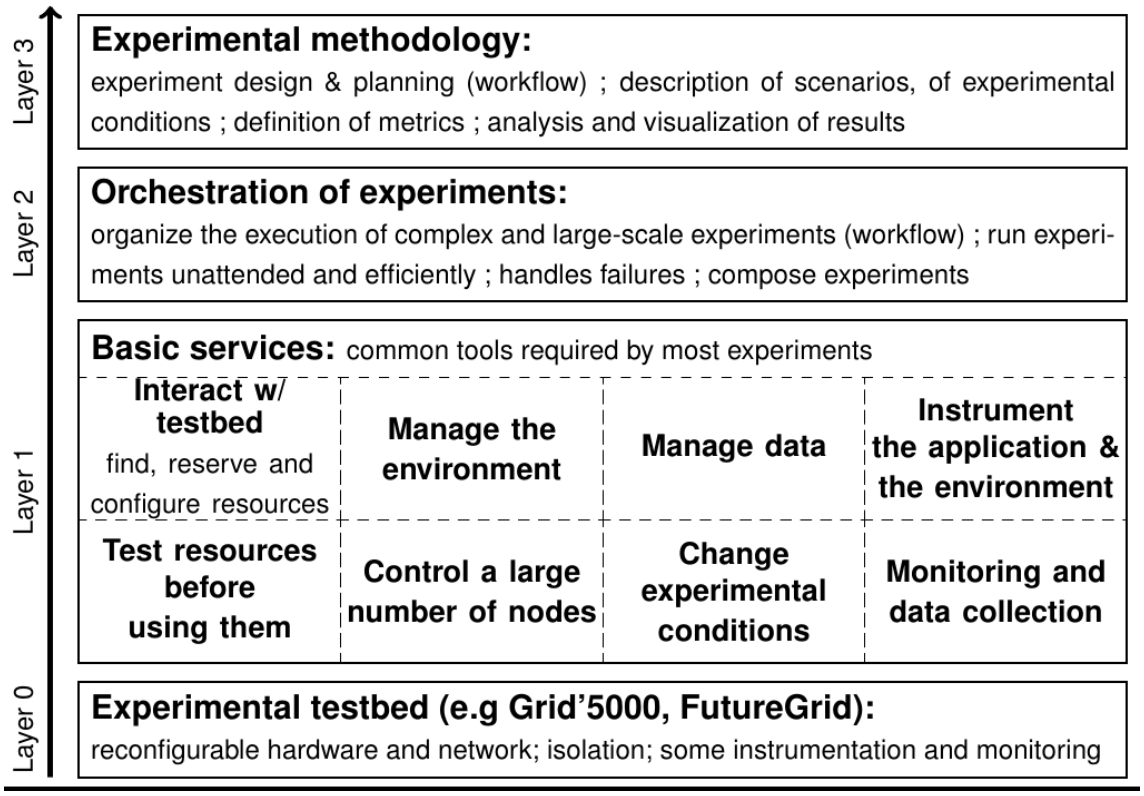

Figure 2. General structure of our project: We plan to address all layers of the experimentation stack.

4.2. Experimental methodologies for the evaluation of distributed systems

Our experimental research axis has a *meta* positioning, targeting all large-scale distributed systems. This versatility allows us to factorize the efforts and maximize our efficiency. The resulting findings are typically used by researchers and developers of systems in the following domains:

- High Performance Computing systems (in particular MPI applications on high-end platforms)
- Cloud environments (in particular virtualized environments)
- Grids (in particular high throughput computing systems)
- Peer-to-peer systems

5. New Software and Platforms

5.1. Introduction

Software is a central part of our output. In the following we present the main tools to which we contribute. We use the Inria software self-assessment catalog for a classification.

5.2. Implementing parallel models

Several software platforms have served us to implement and promote our ideas in the domain of coarse grained computation and application structuring.

5.2.1. ORWL and P99

Participants: Jens Gustedt, Stéphane Vialle [External collaborator, SUPELEC], Mariem Saied.

ORWL is a reference implementation of the Ordered Read-Write Lock tools as described in [4]. The macro definitions and tools for programming in C99 that have been implemented for ORWL have been separated out into a toolbox called P99. ORWL is intended to become opensource, once it will be in a publishable state. P99 is available under a QPL at http://p99.gforge.inria.fr/.
Software classification: A-3-up, SO-4, SM-3, EM-3, SDL (P99: 4, ORWL: 2-up), DA-4, CD-4, MS-3, TPM-4

5.2.2. parXXL

Participants: Jens Gustedt, Stéphane Vialle [External collaborator, SUPELEC].

ParXXL is a library for large scale computation and communication that executes fine grained algorithms on coarse grained architectures (clusters, grids, mainframes). It has been one of the software bases of the InterCell project and has been proven to be a stable support, there. It is available under a GPLv2 at http://parxxl.gforge.inria.fr/. ParXXL is not under active development anymore, but still maintained in the case of bugs or portability problems.
Software classification: A-3, SO-4, SM-3, EM-2, SDL-4, DA-4, CD-4, MS-2, TPM-2

5.2.3. musl

Participant: Jens Gustedt.

musl is a re-implementation of the C library as it is described by the C and POSIX standards. It is *lightweight*, *fast*, *simple*, *free*, and strives to be correct in the sense of standards-conformance and safety. Musl is production quality code that is mainly used in the area of embedded device. It gains more market share also in other area, *e.g.* there are now Linux distributions that are based on musl instead of Gnu LibC.

In 2014, we have added an implementation of the new thread interface that had been defined in the recent C11 standard.

5.3. Parallel developments for numerical scientific application

Participant: Sylvain Contassot-Vivier.

The RAD2D/RAD3D software are co-developed with Fatmir Asllanaj, full researcher in physics at the LEMTA Laboratory, in the context of an inter-disciplinary collaboration. The object of those software is to solve and compute the radiative-transfer equation by using the finite volume method. As the amount of computations induced is very large, the resort to parallelism is mandatory [9], [15]. By its complexity and similarity with a large proportion of scientific applications, this real case application is a fully pertinent test-case for the parallel techniques and schemes we have designed in our team. Those software are not open-source and, by the way, are still in development state.

5.4. Distem

Participants: Tomasz Buchert, Emmanuel Jeanvoine, Lucas Nussbaum, Luc Sarzyniec.

Wrekavoc and Distem are distributed system emulators. They enable researchers to evaluate unmodified distributed applications on heterogeneous distributed platforms created from an homogeneous cluster: CPU performance and network characteristics are altered by the emulator.
Wrekavoc was developed until 2010, and we then focused our efforts on **Distem**, that shares the same goals with a different design. Distem is available from http://distem.gforge.inria.fr/ under GPLv3.
Software classification: A-3-up, SO-4, SM-3-up, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4.

5.5. SimGrid

SimGrid is a toolkit for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning.

5.5.1. Core distribution

Participants: Martin Quinson, Marion Guthmuller, Paul Bédaride, Gabriel Corona, Lucas Nussbaum.

SimGrid has an active user community of more than one hundred members, and is available under GPLv3 from http://simgrid.gforge.inria.fr/. One third of the source code is devoted to about 12000 unit tests and 500 full integration tests. These tests are run for each commit for 4 package configurations and on 4 operating systems thanks to the Inria continuous integration platform.
Software classification: A-5, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-4, TPM-4.

5.5.2. SimGridMC

Participants: Martin Quinson, Marion Guthmuller, Gabriel Corona.

SimGridMC is a module of SimGrid that can be used to formally assess any distributed system that can be simulated within SimGrid. It explores all possible message interleavings searching for states violating the provided properties. We recently added the ability to assess liveness properties over arbitrary C codes, thanks to a system-level introspection tool that provides a finely detailed view of the running application to the model checker. This can for example be leveraged to verify arbitrary MPI code written in C.
Software classification: A-3-up, SO-4, SM-3-up, EM-3-up, SDL-5, DA-4, CD-4, MS-4, TPM-4.

5.5.3. SCHIaaS

Participants: Julien Gossa [External collaborator, SUPELEC], Stéphane Genaud [External collaborator, SUPELEC], Rajni Aron.

The *Simulation of Clouds, Hypervisor and IaaS* (SCHIaaS) is an extension of SimGrid that can be used to comprehensively simulate clouds, from the hypervisor/system level, to the IaaS/administrator level. The hypervisor level includes models about virtualization overhead and VMs operations like boot, start, suspend, migrate, and network capping. The IaaS level includes models about instances management like image storage and deployment and VM scheduling. This extension allows to fully simulate any cloud infrastructure, whatever the hypervisor or the IaaS manager. This can be used by both cloud administrators to dimension and tune clouds, and cloud users to simulate cloud applications and assess provisioning strategies in term of performances and cost.
Software classification: A-3-up, SO-3, SM-2-up, EM-2-up, SDL-2, DA-4, CD-4, MS-4, TPM-4.

5.6. Kadeploy

Participants: Luc Sarzyniec, Stéphane Martin, Emmanuel Jeanvoine, Lucas Nussbaum [correspondant].

Kadeploy is a scalable, efficient and reliable deployment (provisioning) system for clusters and grids. It provides a set of tools for cloning, configuring (post installation) and managing cluster nodes. It can deploy a 300-nodes cluster in a few minutes, without intervention from the system administrator. It plays a key role on the Grid'5000 testbed, where it allows users to reconfigure the software environment on the nodes, and is also used on a dozen of production clusters both inside and outside INRIA. It is available from http://kadeploy3.gforge.inria.fr/ under the Cecill license.
Software classification: A-4-up, SO-3, SM-4, EM-4, SDL-4-up, DA-4, CD-4, MS-4, TPM-4.

5.7. XPFlow

Participants: Tomasz Buchert, Lucas Nussbaum [correspondant].

XPFlow is an implementation of a new, workflow-inspired approach to control experiments involving large-scale computer installations. Such systems pose many difficult problems to researchers due to their complexity, their numerous constituents and scalability problems. The main idea of the approach consists in describing the experiment as a workflow and execute it using achievements of Business Process Management (BPM), workflow management techniques and scientific workflows. The website of XPFlow is http://xpflow.gforge.inria.fr/. XPFlow was featured in a tutorial during Grid'5000 Spring School 2014.
Software classification: A-2-up, SO-3-up, SM-2-up, EM-3-up, SDL-2-up, DA-4, CD-4, MS-4, TPM-4.

5.8. Grid'5000 testbed

Participants: Luc Sarzyniec, Jérémie Gaidamour, Arthur Garnier, Clément Parisot, Emmanuel Jeanvoine, Émile Morel, Lucas Nussbaum [correspondant].

Grid'5000 (http://www.grid5000.fr) is a scientific instrument designed to support experiment-driven research in all areas of computer science related to parallel, large-scale or distributed computing and networking. It gathers 10 sites, 25 clusters, 1200 nodes, for a total of 8000 cores. It provides its users with a fully reconfigurable environment (bare metal OS deployment with Kadeploy, network isolation with KaVLAN) and a strong focus on enabling high-quality, reproducible experiments.
The AlGorille team contributes to the design of Grid'5000, to the administration of the local Grid'5000 site in Nancy, and to the design and development of Kadeploy (in close cooperation with the Grid'5000 technical team). The AlGorille engineers also administer *Inria Nancy – Grand Est*'s local production cluster, named *Talc*, leveraging the experience and tools from Grid'5000.

Software classification: A-5, SO-4, SM-4, EM-4, SDL-N/A, DA-4, CD-4, MS-4, TPM-4.

6. New Results

6.1. Structuring applications for scalability

6.1.1. Combining locking and data management interfaces

Participants: Jens Gustedt, Mariem Saied.

Handling data consistency in parallel and distributed settings is a challenging task, in particular if we want to allow for an easy to handle asynchronism between tasks. Our publication [4] shows how to produce deadlock-free iterative programs that implement strong overlapping between communication, IO and computation.

A new implementation (ORWL) of our ideas of combining control and data management in C has been undertaken, see 5.2.1. In 2014, work has demonstrated its efficiency for a large variety of platforms, see [20]. By using the example of dense matrix multiplication, we show that ORWL permits to reuse existing code for the target architecture, namely open source library ATLAS, Intel's compiler specific MKL library or NVidia's CUBLAS library for GPUs. ORWL assembles local calls into these libraries into efficient functional code, that combines computation on distributed nodes with efficient multi-core and accelerator parallelism.

Our next efforts will concentrate on the continuation of an implementation of a complete application (an American Option Pricer) that was chosen because it presents a non-trivial data transfer and control between different compute nodes and their GPU. ORWL is able to handle such an application seamlessly and efficiently, a real alternative to home made interactions between MPI and CUDA.

6.2. Experimental methodologies for the evaluation of distributed systems

6.2.1. Simulation and dynamic verification

6.2.1.1. SimGrid framework improvement

Participants: Paul Bédaride, Martin Quinson, Gabriel Corona.

On the technical side, we kept up with our regular releases of the SimGrid framework, integrating the work of our partners in the SONGS ANR project. This year, we reimplemented the simulation kernel in C++. This modularity improvement will ease the addition of performance models by external contributors. This work thus contributes to our overall goal of constituting a user community focused on this first-class tool.

[11] is a long awaited paper describing the current state of the project and its future roadmap. This constitutes the new reference paper on the SimGrid project (the previous article, a short paper from 2008, was cited over 350 times since its publication). We show that despite the common beliefs, the tool specialization is not necessarily a warrant for performance and correctness.

We also continued our animation of our scientific community, for example through our participation to the Joint Laboratory for Petascale Computing (Inria/ANL/UIUC/BSC). We co-organized a summer school on Performance Metrics, Modeling and Simulation of Large HPC Systems in June, to push our tools toward PhD students that need to assess their HPC applications.

6.2.1.2. Dynamic verification and SimGrid

Participants: Marion Guthmuller, Martin Quinson, Gabriel Corona.

This year, the PhD thesis of M. Guthmuller went into its third year. The proposed methodology maturated into a usable tool: we can now verify small-size real HPC applications using MPI in C/C++/Fortran. This relies on a heuristic exploration of the applicative state at the system level that was presented in [21], [22].

Also, we finally added the ability to dynamically verify some CTL properties over MPI implementations. SimGrid was one of the rare framework able to verify LTL liveness properties over real implementations. To the best of our knowledge, it becomes the very first tool verifying CTL properties on real C/C++/Fortran applications. The targeted properties quantify the stability of the applicative communication pattern. The applications that respect these properties can benefit from specific, more efficient, fault tolerance algorithms. Verifying these properties is thus of a major practical interest. A publication is in preparation, as well as the PhD manuscript of M. Guthmuller who will defend by 2015 Q1.

6.2.2. Experimentation on testbeds and production facilities, emulation

6.2.2.1. Evaluating load balancing and fault tolerance strategies on Distem

Participants: Joseph Emeras, Emmanuel Jeanvoine, Lucas Nussbaum.

(For context, see sections 3.3 and 5.4.)

We extended our work [27] to enable the study of load balancing and fault tolerance strategies on Distem. Distem now supports the introduction of changing heterogeneity and imbalance among virtual nodes, as well as the introduction of failures. Two HPC runtimes targeting Exascale (Charm++ and OpenMPI) were used as target applications. This work was presented at the Joint Laboratory for Extreme-Scale Computing in June, and at the Grid'5000 Spring School. However, those results still have to be properly published.

6.2.2.2. Distem improvements: VXLAN, release and tutorial

Participants: Emmanuel Jeanvoine, Tomasz Buchert, Lucas Nussbaum.

(For context, see sections 3.3 and 5.4.)

The scalability of Distem's networking layer was improved by adding support for VXLAN networks. This enabled experiments with up to 40,000 virtual nodes, presented at the CCGrid'2014 SCALE challenge (where we were selected as finalist) [17]. Version 1.0 of Distem was also released in March 2014, and featured in a tutorial at the Grid'5000 Spring School.

6.2.2.3. Kadeploy improvements: REST API, new image broadcast mechanism

Participants: Luc Sarzyniec, Stéphane Martin, Emmanuel Jeanvoine, Lucas Nussbaum.

(For context, see sections 3.3 and 5.4.)

Kadeploy 3.2 was released in March 2014. Among many other changes, that release included a new REST API to interact with Kadeploy, replacing the old Ruby-specific RPC mechanism, and easing the automation of experiments by providing a way to call Kadeploy from scripts.

Kadeploy 3.3 was released in November 2014. This release is mostly a bug-fix release, with many bug fixes in the internal cache system, the shell runner, and others.

We also implemented an improved mechanism to broadcast machine images to nodes. The new tool, called Kascade, is fault tolerant, and its performance has been thoroughly tested. It was described in a publication accepted at HPDIC'2014 [24], included in Kadeploy 3.2, and used as the default method for environment broadcast since Kadeploy 3.3.

6.2.2.4. XPFlow

Participants: Tomasz Buchert, Stéphane Martin, Emmanuel Jeanvoine, Lucas Nussbaum, Jens Gustedt.

(For context, see sections 3.3 and 5.7.)

A publication focusing on XPFlow was accepted at CCGrid'2014 [18], and XPFlow was also featured in a tutorial at Grid'5000 Spring School. Our ongoing work focuses on improved support for collecting provenance in XPFlow.

6.2.2.5. Survey of Experiment Management tools

Participants: Tomasz Buchert, Cristian Ruiz, Lucas Nussbaum.

We produced a survey of Experiment Management tools for distributed systems, published in Future Generation Computer Systems [10]. This survey provides an extensive list of features offered by general-purpose experiment management tools dedicated to distributed systems research on real platforms. It then uses it to assess existing solutions and compare them, outlining possible future paths for improvements.

6.2.2.6. Grid'5000

Participants: Émile Morel, Luc Sarzyniec, Lucas Nussbaum.

(For context, see sections 3.3 and 5.8.)

The work on resources description, selection, reservation and verification was wrapped-up in a TridentCom'2014 paper [23].

As a member of the Grid'5000 architects committee, Lucas Nussbaum was involved in the submission (and acceptance) of ADT Laplace.

Lucas Nussbaum also presented a talk [12] on Reproducible Research and Grid'5000 at the Grid'5000 evaluation by the Scientific Committee, during the Spring School.

6.2.3. Convergence and co-design of experimental methodologies

6.2.3.1. Realis'2014

Participant: Lucas Nussbaum.

Lucas Nussbaum organized (with Olivier Richard) the second edition of the Realis event [14]. Associated to the Compas'14 conference, this workshop aimed at providing a place to discuss the reproducibility of the experiments underlying the publications submitted to the main conference. We hope that this kind of venue will motivate the researchers to further detail their experimental methodology, ultimately allowing others to reproduce their experiments.

6.2.3.2. Reproducible Research working group at Inria Nancy – Grand Est

Participant: Lucas Nussbaum.

Lucas Nussbaum is organizing a working group on Reproducible Research at Inria Nancy – Grand Est since May 2014. Meetings involve a dozen of members from many different teams, and discussion topics have so far covered online platforms to test algorithms and applications, and evaluation contests organized together with conferences and workshops.

Lucas Nussbaum has also been invited to participate in the Inria national initiative on reproducible research.

6.2.3.3. Organization of Reppar

Participant: Lucas Nussbaum.

Lucas Nussbaum co-organized the first edition of the Reppar workshop, held during Europar'2014, with a focus on experimental practices in parallel computing research.

6.3. Algorithmic schemes for efficient use of parallel devices in clusters

Participants: Sylvain Contassot-Vivier, Stéphane Vialle [External collaborator, SUPELEC].

During the year 2014, we have continued our studies about the design and implementation of efficient algorithmic schemes to fully exploit all the available computational resources inside a parallel system. In particular, we have proposed general schemes that optimize the use of GPUs in clusters [26]. This is achieved by performing two kinds of overlappings. The former corresponds to computation/communication overlappings, either for the communications between machines but also for the data transfers between central RAM and GPUs inside each machine. The latter is the computation/computation overlapping that consists in executing computations on the GPUs in parallel of some computations on the central CPUs. Moreover, in this work we have paid a particular attention to some important aspects of software engineering that are the development and maintenance costs. Those aspects are essential as they directly determine the practical usability of the schemes, especially in the industry where there is a permanent vigilance to minimize the associated costs.

6.4. Parallel schemes for the resolution of the RTE with finite volumes method

Participant: Sylvain Contassot-Vivier.

In the context of our collaboration with the Lemta laboratory (Fatmir Asllanaj), about the design and implementation of an efficient and high accuracy algorithm for solving the Radiative Transfer Equation (RTE), we have reached our second objective that consisted in the realization of a multi-threaded parallel version of the software. That new version is based on the optimized sequential version produced as a first objective. It makes use of the OpenMP library to exploit all the cores inside one machine. The results are very satisfying as our algorithm obtains very good speed up and efficiency (around 90% and above) in realistic contexts. Moreover, besides this work over performance, we focus also on the high quality (accuracy) of the results of our software by making a permanent effort to track any possible enhancement of our numerical scheme. Then, the actual implementation of each of these possible enhancements is considered according to its potential costs, either in performance degradation as well as in additional resource consumptions (CPUs, GPUs and RAM). Confrontations to other existing computational schemes to solve the RTE are regularly realized to corroborate the validity preservation of our software [9], [15].

6.5. Study of binary multiplication and dynamical approaches to the integer factorization

Participants: Sylvain Contassot-Vivier, Nazim Fatès.

In the context of a collaboration with Nazim Fatès over dynamical systems we have co-supervised the internship of Raphaël Rieu-Helft (student at the ENS Paris), during June and July 2014. The goal of this internship was to study the relevance of the dynamical systems formalism as an efficient way to express and solve two specific problems. The former one was the queens problem on chessboards of arbitrary size. This goal was to express a solving algorithm of the queens problem under the form of a cellular automaton. The second step was to extend the results obtained for the queens problem to a more complex and computationally expensive problem that is the integer factorization. Two dynamical systems (cellular automata) have been obtained for both problems and their respective efficiencies, either in terms of convergence speed or speed of solution reaching, have been experimentally evaluated.

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. ANR

Plate-form(E)^3 (2012-2015, 87k€) has been accepted in 2012 in the ANR SEED program. It deals with the design and implementation of a multi-scale computing and optimization platform for energetic efficiency in industrial environment. It gathers 7 partners either academic (LEMTA, Fédération Charles Hermite (including AlGorille), Mines Paris, INDEED) or industrial (IFPEN, EDF, IDEEL). We will contribute to the design and development of the platform. The engineer P. Kalitine has been recruited to work on this project from May 2014 to June 2015.

ANR SONGS (2012–2015, 1800k€) Martin Quinson is also the principal investigator of this project, funded by the ANR INFRA program. SONGS (Simulation Of Next Generation Systems) aims at increasing the target community of SimGrid to two new research domains, namely Clouds (restricted to the *Infrastructure as a Service* context) and High Performance Computing. We develop new models and interfaces to enable the use of SimGrid for generic and specialized researches in these domains.

As project leading team, we are involved in most parts of this project, which allows the improvement of our tool even further and sets it as the reference in its domain (see Section 6.2.1).

7.1.2. Inria financed projects and clusters

AEN Hemera (2010-2014, 2k€) aims at demonstrating ambitious up-scaling techniques for large scale distributed computing by carrying out several dimensioning experiments on the Grid'5000 infrastructure, and at animating and enlarging the scientific community around the testbed. M. Quinson, L. Nussbaum and S. Genaud lead three working groups, respectively on *simulating large-scale facilities*, on *conducting large and complex experimentations on real platforms*, and on *designing scientific applications for scalability*.
Other partners: 20 research teams in France, see https://www.grid5000.fr/mediawiki/index.php/Hemera for details.

ADT Aladdin-G5K (2007-2014, 200k€ locally) aims at the construction of a scientific instrument for experiments on large-scale parallel and distributed systems, building on the Grid'5000 testbed (http://www.grid5000.fr/). It structures INRIA's leadership role (8 of the 9 Grid'5000 sites) concerning this platform. The technical team is now composed of 10 engineers, of which 2 are currently hosted in the AlGorille team. As a member of the executive committee, L. Nussbaum is in charge of following the work of the technical team, together with the Grid'5000 technical director.
Other partners: EPI DOLPHIN, GRAAL, MESCAL, MYRIADS, OASIS, REGAL, RESO, RUNTIME, IRIT (Toulouse), Université de Reims - Champagne Ardennes

ADT LAPLACE (2014-2016, AlGorille is major partner, 100k€) builds upon the foundations of the Grid'5000 testbed to reinforce and extend it towards new use cases and scientific challenges. Several directions are being explored: networks and Software Defined Networking, Big Data, HPC, and production computation needs. Already developed prototypes are also being consolidated, and the necessary improvements to user management and tracking are also being performed.

ADT Cosette (2013-2016, AlGorille is the only partner, 120k€), for *COherent SET of Tools for Experimentation* aims at developing or improving a tool suite for experimentation at large scale on testbeds such as Grid'5000. Specifically, we will work on (1) the development of Ruby-CUTE, a library gathering features useful when performing such experiments; (2) the porting of Kadeploy, Distem and XPFlow on top of Ruby-CUTE; (3) the release of XPFlow, developed in the context of Tomasz Buchert's PhD; (4) the improvement of the Distem emulator to address new scientific challenges in Cloud and HPC. E. Jeanvoine (SED) is delegated in the AlGorille team for the duration of this project.

INRIA Project Lab MultiCore (2013-) Supporting multicore processors in an efficient way is still a scientific challenge. This project introduces a novel approach based on virtualization and dynamicity, in order to mask hardware heterogeneity, and to let performance scale with the number and nature of cores. Our main partner within this project is the Camus team on the Strasbourg site. The move of J. Gustedt there, has strengthened the collaboration within this project.

ADT PLM (2014-2016, Martin Quinson is leading this project in collaboration with G. Oster from the Coast project-team, 100k€) This project is not directly in line with the goal of the AlGorille project-team, as its goal is to establish an experimental platform to study of the didactic of informatics, specifically centered on introductory programming courses.

The project builds upon a pedagogical programming exerciser developed for our own teaching, and improves this base in several ways. We want to provide more adapted feedback to the learners, and gather more data to better understand how beginners learn programming.

7.2. European Initiatives

7.2.1. FP7 Projects

7.2.1.1. FED4FIRE

Participant: Lucas Nussbaum.

Title: Federation for Future Internet Research and Experimentation

Type: ICT

Instrument: Integrated Project

Duration: October 2012 - September 2016

Coordinator: iMinds

Other partners: IT Innovation, UPMC, Fraunhofer, TUB, UEDIN, Inria, NICTA, ATOS, UTH, NTUA, UNIVBRIS, i2CAT, EUR, DANTE Limited, UC, NIA.

See also: http://www.fed4fire.eu

Abstract: The key outcome of Fed4FIRE will be an open federation solution supporting all stakeholders of FIRE. Fed4FIRE is bringing together key players in Europe in the field of experimentation facilities and tool development who play a major role in the European testbeds of the FIRE initiative projects.

Lucas Nussbaum started participating in the project in September 2013, mainly with an expert role.

7.3. International Research Visitors

7.3.1. Visits of International Scientists

7.3.1.1. Internships

Ezequiel Torti Lopez

Subject: Parallel and Distributed Simulation of Large-Scale Distributed Applications

Date: from May 2014 until October 2014

Institution: Universidad National de Rosario (Argentina)

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific events organisation

8.1.1.1. Organizing committee membership

Lucas Nussbaum was a member of the organizing committee for Reppar (1st International Workshop on Reproducibility in Parallel Computing, held together with Euro-Par 2014).

Martin Quinson was a member of the organizing committee for the Fourth SimGrid Users' Days (Le Bono, Brittany, France), and of the PUF/JLPC Summer school on Performance Metrics, Modeling and Simulation of Large HPC Systems (Sophia Antipolis, France).

8.1.2. Scientific events selection

8.1.2.1. Conference program committee membership

Lucas Nussbaum was a member of the PC for Reppar (1st International Workshop on Reproducibility in Parallel Computing, held together with Euro-Par 2014), CloudCom'2014 (HPC on Cloud track), WETICE'2014 (Convergence of Distributed Grid, Cloud and their Management track), Grid'5000 Spring School 2014, and ComPAS'2014.

Martin Quinson was a member of the PC for IPDPS'14 (ACM/IEEE International Parallel and Distributed Processing Symposium), SimulTech'14 (In cooperation with ACM SIGSIM) and PADS'14 (ACM SIGSIM Workshop on Principles of Advanced Discrete Simulation).

8.1.3. Journal

8.1.3.1. Editorial board membership

Since October 2001, J. Gustedt is Editor-in-Chief of the journal *Discrete Mathematics and Theoretical Computer Science* (DMTCS).

In 2014, the episcience platform for open access journals has been created in a joint effort by Inria and CNRS. DMTCS will be one of the first journals passing to this platform and we have been a driving force in the definition and debugging of the new platform, see [16].

Since 2013, M. Quinson is a member of the editorial board of the Interstices journal (edited by Inria in collaboration), aiming at increasing the scientific outreach of informatic.

8.1.3.2. Reviewing Activities

Jens Gustedt has served as a reviewer for *Social Network Analysis and Mining* and *IEEE Transactions on Parallel and Distributed Systems*.

Lucas Nussbaum was a reviewer for *Security and Communication Networks*, and *IEEE Transactions on Emerging Topics in Computing*.

Sylvain Contassot-Vivier is a regular reviewer for the *Engineering Applications of Artificial Intelligence* journal.

8.1.4. Standardization

Since Nov. 2014, Jens Gustedt is a member of the ISO working group SC22-WG14 for the standardization of the C programming language.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

IUT Nancy-Charlemagne, Université de Lorraine: Lucas Nussbaum, Installation of Linux, 20 ETD, Licence pro ASRALL (L3), Université de Lorraine, France.

IUT Nancy-Charlemagne, Université de Lorraine: Lucas Nussbaum, Outils Libres, 16 ETD, Licence pro ASRALL (L3), Université de Lorraine, France.

IUT Nancy-Charlemagne, Université de Lorraine: Lucas Nussbaum, Administration des infrastructures avancées, 12 ETD, Licence pro ASRALL (L3), Université de Lorraine, France.

École CNRS "Informatique pour le calcul scientifique": Lucas Nusbaum, Gestion de versions avec Git, 3h.

TELECOM Nancy, Université de Lorraine: Lucas Nussbaum, Introduction au Logiciel Libre et au projet Debian, 2h.

TELECOM Physique Strasbourg: Jens Gustedt, parallélisme, 10h ETD.

ENSIIE Strasbourg: Jens Gustedt, programmation avancée, 20h ETD.

Département Informatique, Université de Strasbourg: Jens Gustedt, systèmes concurrents, 20h ETD.

TELECOM Nancy, Université de Lorraine: Martin Quinson, “Algorithmique et Programmation,”, 48 ETD, 1ere année (L3), Université de Lorraine, France.

TELECOM Nancy, Université de Lorraine: Martin Quinson, “Langage C et programmation shell,”, 48 ETD, 1ere année (L3), Université de Lorraine, France.

TELECOM Nancy, Université de Lorraine: Martin Quinson, “Programmation Système,”, 24 ETD, 2ème année (M1), Université de Lorraine, France.

Faculté des Sciences et Technologies, Université de Lorraine: Sylvain Contassot-Vivier, “Architectures logicielles avancées”, 19 ETD, M2, Université de Lorraine, France.

Faculté des Sciences et Technologies, Université de Lorraine: Sylvain Contassot-Vivier, “Informatique graphique”, 63.5 ETD, L1, Université de Lorraine, France.

Faculté des Sciences et Technologies, Université de Lorraine: Sylvain Contassot-Vivier, “Sécurité”, 10 ETD, L3, Université de Lorraine, France.

TELECOM Nancy, Université de Lorraine: Sylvain Contassot-Vivier, “Algorithmique Parallèle”, 35 ETD, 2ème année (M1), Université de Lorraine, France.

Faculté des Sciences et Technologies, Université de Lorraine: Sylvain Contassot-Vivier, “Algorithmique et Programmation”, 16 ETD, L1, Université de Lorraine, France.

Faculté des Sciences et Technologies, Université de Lorraine: Sylvain Contassot-Vivier, “Algorithmique et Programmation”, 12 ETD, L2, Université de Lorraine, France.

Faculté des Sciences et Technologies, Université de Lorraine: Sylvain Contassot-Vivier, “Algorithmique et Programmation”, 32.5 ETD, L3, Université de Lorraine, France.

8.2.2. Supervision

PhD in progress: Tomasz Buchert, *Orchestration of experiments on distributed systems*, since Oct 2011, Jens Gustedt & Lucas Nussbaum.

PhD in progress: Marion Guthmuller, *Dynamic verification of distributed applications, using a model-checking approach*, since Oct 2011, Sylvain Contassot-Vivier & Martin Quinson.

PhD in progress: Mariem Saied, *Ordered Read-Write Locks for Multicores and Accelerators*, since Nov 2013, Jens Gustedt & Gilles Muller.

Internship: Raphaël Rieu-Helft, *Étude de la multiplication binaire et approches dynamiques pour la factorisation de nombres*, June and July 2014, Sylvain Contassot-Vivier & Nazim Fatès (MAIA team).

8.2.3. Juries

Lucas Nussbaum was a member of the recruitment committee for an associate professor (MCF) in computer science at ENS Lyon.

Sylvain Contassot-Vivier was member and president of the PhD committee of Carlos Carvajal. He is also the Loria referent of several PhD students (referents have to follow and check the regular progress of the PhD).

8.3. Popularization

Jens Gustedt is regularly blogging about efficient programming, in particular about the C programming language. He also is an active member of the stack**overflow** community a technical Q&A site for programming and related subjects. A book about modern C is in preparation.

In collaboration with G. Oster, Coast team of Inria Nancy Grand-Est, M. Quinson develops a pedagogic platform. This tool aims at providing an environment that is both appealing for the student, easy to use for the teacher, and efficient for the learning process. Since 2014, an Inria ADT project aims at changing this practical tool into an experimental platform to study the didactic of programming.

M. Quinson is co-leading a working group on the teaching of computer science in the LORIA laboratory. He served both as a program chair and a local chair for a nation-wide two-days workshop gathering about hundred people involved in the introduction of computer science in the French secondary education: university lecturers in charge of teaching to the prospective CS teachers, regional heads of the Education minister accompanying this reform and producer of teaching resources. He also served both as a program chair and local chair for a regional gathering of CS teachers of the secondary wanting to exchange their good practices. This initiative, initiated in Nancy, will spread in several other French cities in 2015.

Either as a speaker or as a co-organizer, M. Quinson participated in several events that aim the popularization of computer science. These targeted a large variety of public, kids and pupils (Telecom Nancy in November), university students (Inria in March), mathematics teachers (APMEP Lorraine in March), CS teachers (SIF-ISN day in June), or the general public (Fête de la science in November).

S. Contassot-Vivier has participated to Loria animation days towards high-school visitors. In this context he has used the material and games co-designed by M. Quinson and others in the context of unplugged computer science teaching.

9. Bibliography

Major publications by the team in recent years

[1] T. BUCHERT, L. NUSSBAUM, J. GUSTEDT. *A workflow-inspired, modular and robust approach to experiments in distributed systems*, in "CCGRID - 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing", Chicago, United States, May 2014, https://hal.inria.fr/hal-00909347

[2] T. BUCHERT, C. RUIZ, L. NUSSBAUM, O. RICHARD. *A survey of general-purpose experiment management tools for distributed systems*, in "Future Generation Computer Systems", 2015, vol. 45, pp. 1 - 12 [*DOI :* 10.1016/J.FUTURE.2014.10.007], https://hal.inria.fr/hal-01087519

[3] H. CASANOVA, A. GIERSCH, A. LEGRAND, M. QUINSON, F. SUTER. *Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms*, in "Journal of Parallel and Distributed Computing", June 2014, vol. 74, n° 10, pp. 2899-2917 [*DOI :* 10.1016/J.JPDC.2014.06.008], https://hal.inria.fr/hal-01017319

[4] P.-N. CLAUSS, J. GUSTEDT. *Iterative Computations with Ordered Read-Write Locks*, in "Journal of Parallel and Distributed Computing", 2010, vol. 70, n° 5, pp. 496-504 [*DOI :* 10.1016/J.JPDC.2009.09.002], http://hal.inria.fr/inria-00330024/en

[5] P.-N. CLAUSS, M. STILLWELL, S. GENAUD, F. SUTER, H. CASANOVA, M. QUINSON. *Single Node On-Line Simulation of MPI Applications with SMPI*, in "International Parallel & Distributed Processing Symposium", Anchorange (AK), États-Unis, IEEE, May 2011, http://hal.inria.fr/inria-00527150/en/

[6] J. GUSTEDT, E. JEANNOT, M. QUINSON. *Experimental Validation in Large-Scale Systems: a Survey of Methodologies*, in "Parallel Processing Letters", 2009, vol. 19, n° 3, pp. 399-418, RR-6859, http://hal.inria.fr/inria-00364180/en/

[7] T. JOST, S. CONTASSOT-VIVIER, S. VIALLE. *An efficient multi-algorithms sparse linear solver for GPUs*, in "Parallel Computing: From Multicores and GPU's to Petascale (Volume 19)", B. CHAPMAN, F. DESPREZ, G. R. JOUBERT, A. LICHNEWSKY, F. PETERS, T. PRIOL (editors), Advances in Parallel Computing, IOS Press, 2010, vol. 19, pp. 546-553, Extended version of EuroGPU symposium article, in the International Conference on Parallel Computing (Parco) 2009 [*DOI :* 10.3233/978-1-60750-530-3-546], http://hal.inria.fr/hal-00485963/en

Publications of the year

Articles in International Peer-Reviewed Journals

[8] L. A. ABBAS-TURKI, S. VIALLE, B. LAPEYRE, P. MERCIER. *Pricing derivatives on graphics processing units using Monte Carlo simulation*, in "Concurrency and Computation: Practice and Experience", June 2014, vol. 26, n° 9, pp. 1679-1697 [*DOI :* 10.1002/CPE.2862], https://hal-supelec.archives-ouvertes.fr/hal-00773740

[9] F. ASLLANAJ, S. CONTASSOT-VIVIER. *Radiative transfer equation for predicting light propagation in biological media: comparison of a modified finite volume method, the monte carlo technique, and an exact analytical solution*, in "Journal of Biomedical Optics", 2014, vol. 19, n° 1, 10 p. [*DOI :* 10.1117/1.JBO.19.1.015002], https://hal.inria.fr/hal-01101240

[10] T. BUCHERT, C. RUIZ, L. NUSSBAUM, O. RICHARD. *A survey of general-purpose experiment management tools for distributed systems*, in "Future Generation Computer Systems", 2015, vol. 45, pp. 1 - 12 [*DOI :* 10.1016/J.FUTURE.2014.10.007], https://hal.inria.fr/hal-01087519

[11] H. CASANOVA, A. GIERSCH, A. LEGRAND, M. QUINSON, F. SUTER. *Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms*, in "Journal of Parallel and Distributed Computing", June 2014, vol. 74, n° 10, pp. 2899-2917 [*DOI :* 10.1016/J.JPDC.2014.06.008], https://hal.inria.fr/hal-01017319

Invited Conferences

[12] L. NUSSBAUM. *Grid'5000 for high-quality reproducible research*, in "Grid'5000 Spring School 2014", Lyon, France, June 2014, https://hal.inria.fr/hal-01011403

[13] L. NUSSBAUM. *Reproducible Research in Computer Science*, in "Séminaire du laboratoire ICube", Strasbourg, France, January 2015, https://hal.inria.fr/hal-01110206

[14] L. NUSSBAUM, O. RICHARD. *Realis'2014: Reproductibilité expérimentale pour l'informatique en parallélisme, architecture et système*, in "ComPAS : Conférence d'informatique en Parallélisme, Architecture et Système", Neuchatel, Switzerland, April 2014, https://hal.inria.fr/hal-01011401

International Conferences with Proceedings

[15] F. ASLLANAJ, S. CONTASSOT-VIVIER. *Radiative transfer equation for predicting light propagation in biological media: comparison of a modified finite volume method, the Monte Carlo technique, and an exact analytical solution* , in "International Conferences on Laser Applications in Life Sciences (LALS)", ULM, Germany, June 2014, https://hal.inria.fr/hal-01101260

[16] C. BERTHAUD, L. CAPELLI, J. GUSTEDT, C. KIRCHNER, K. LOISEAU, A. MAGRON, M. MEDVES, A. MONTEIL, G. RIVERIEUX, L. ROMARY. *EPISCIENCES - an overlay publication platform*, in "ELPUB2014 - International Conference on Electronic Publishing", Thessalonique, Greece, D. P. POLYDORATOU (editor), IOS Press, June 2014, pp. 78-87 [*DOI :* 10.3233/978-1-61499-409-1-78], https://hal.inria.fr/hal-01002815

[17] T. BUCHERT, E. JEANVOINE, L. NUSSBAUM. *Emulation at Very Large Scale with Distem*, in "SCALE Challenge, held in conjunction with the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)", Chicago, United States, May 2014, https://hal.inria.fr/hal-00959616

[18] T. BUCHERT, L. NUSSBAUM, J. GUSTEDT. *A workflow-inspired, modular and robust approach to experiments in distributed systems*, in "CCGRID - 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing", Chicago, United States, May 2014, https://hal.inria.fr/hal-00909347

[19] B. GUILLAUME, K. FORT, G. PERRIER, P. BEDARIDE. *Mapping the Lexique des Verbes du Français (Lexicon of French Verbs) to a NLP Lexicon using Examples*, in "International Conference on Language Resources and Evaluation (LREC)", Reykjavik, Iceland, May 2014, https://hal.inria.fr/hal-00969184

[20] J. GUSTEDT, S. VIALLE, P. MERCIER. *Resource Centered Computing delivering high parallel performance*, in "Heterogeneity in Computing Workshop (HCW 2014), 28th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2014)", Phenix, AZ, United States, IEEE, May 2014, https://hal.inria.fr/hal-00921128

[21] M. GUTHMULLER, M. QUINSON, G. CORONA. *System-level State Equality Detection for the Formal Dynamic Verification of Legacy Distributed Applications*, in "Formal Approaches to Parallel and Distributed Systems (4PAD) - Special Session of Parallel, Distributed and network-based Processing (PDP)", Turku, Finland, March 2015, https://hal.archives-ouvertes.fr/hal-01097204

[22] M. GUTHMULLER, M. QUINSON. *System-level State Equality Detection for the Dynamic Verification of Distributed Applications*, in "EuroSys - 9th European Conference on Computer Systems", Amsterdam, Netherlands, ACM, April 2014, https://hal.inria.fr/hal-00997941

[23] D. MARGERY, E. MOREL, L. NUSSBAUM, O. RICHARD, C. ROHR. *Resources Description, Selection, Reservation and Verification on a Large-scale Testbed*, in "TRIDENTCOM - 9th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities", Guangzhou, China, May 2014, https://hal.inria.fr/hal-00965708

[24] S. MARTIN, T. BUCHERT, P. WILLEMET, O. RICHARD, E. JEANVOINE, L. NUSSBAUM. *Scalable and Reliable Data Broadcast with Kascade*, in "HPDIC - International Workshop on High Performance Data Intensive Computing, in conjunction with IEEE IPDPS 2014", Phoenix, United States, May 2014, https://hal.inria.fr/hal-00957671

[25] C. C. RUIZ SANABRIA, O. RICHARD, J. EMERAS. *Reproducible Software Appliances for Experimentation*, in "TRIDENTCOM - 9th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (2014)", Guangzhou, China, May 2014, https://hal.inria.fr/hal-01064825

Scientific Books (or Scientific Book chapters)

[26] S. VIALLE, S. CONTASSOT-VIVIER. *Optimization methodology for Parallel Programming of Homogeneous or Hybrid Clusters*, in "Patterns for parallel programming on GPUs", F. MAGOULES (editor), Saxe-Coburg Publications, 2014, https://hal.inria.fr/hal-01101225

Research Reports

[27] J. EMERAS, E. JEANVOINE, L. NUSSBAUM. *An Experimental Environment for the Evaluation of Exascale HPC Runtimes*, February 2014, n^{o} RR-8482, https://hal.inria.fr/hal-00949762