IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2013

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.

1. Members

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

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

External Collaborators
Stéphane Genaud [Univ. Strasbourg, Associate Professor, HdR]
Julien Gossa [Univ. Strasbourg, Associate Professor]
Stéphane Vialle [SUPELEC, Professor, HdR]

Engineers
Sébastien Badia [Inria, granted by Caisse des Dépôts et Consignations, until September 2013]
Emmanuel Jeanvoine [Inria, permanent SED engineer delegated to AlGorille]
Paul Bédaride [Univ. Lorraine, granted by ANR SONGS]
Gabriel Corona [Univ. Lorraine, granted by ANR SONGS, since December 2013]
Stéphane Martin [Inria, granted by ANR SONGS, since October 2013]
Émile Morel [Inria]
Luc Sarzyniec [Inria, granted by ANR CATREL]

PhD Students
Tomasz Buchert [Inria, CORDI-S]
Marion Guthmuller [Univ. Lorraine]
Thomas Jost [Inria, granted by FP7 Goodshape, until March 2013]
Mariem Saied [Inria, CORDI-S, since November 2013]
Soumeya Leila Hernane [UST Oran, Algeria, teaching assistant, until June 2013]

Post-Doctoral Fellows
Rajni Aron [Inria, since November 2013]
Joseph Emeras [Inria, since October 2013]

Administrative Assistants
Isabelle Herlich [Inria]
Delphine Hubert [Univ. Lorraine]

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. Layer 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 programing 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 a system are located at the *application* and *service* layers. In addition to these two types of challenges, we identify a third 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 utilization 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 (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 adaption strategies between codes and platforms: e.g the best cache strategies can't help applications that accesses 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.

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. For the architectures we will have to take into account the fact of increased heterogeneity, processors of different speed, 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 applications.

3.1.4. Frequent IO.

A complete parallel application includes I/O of massive data, at an increasing frequency. In addition to applicative input and output data flow, I/O is 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 has to be included in the design of parallel programming models and tools. ORWL will 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 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 its application domain, especially concerning the detection of global convergence and the control of asynchronism. We are currently working on the design of a generic and non-intrusive way of implementing such a procedure in any parallel iterative algorithm.

Also, we would like to compare other variants of asynchronous algorithms, such as waveform relaxations. Here, computations are not performed for each time step of the simulation but for an entire time interval. Then, the evolution of the elements at the frontiers between the domain that are associated to the processors are exchanged asynchronously. Although we have already studied such schemes in the past, we would like to see how they will behave on recent architectures, and how the models and software for data consistency mentioned above can be helpful in that context.

3.1.6. Cost models and accelerators.

We have already designed some models that relate computation power and energy consumption. Our next goal is to design and implement 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 able to adapt its behavior to the requirements.

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.

We are currently extending the applicability to **Clouds and Exascale systems**. Therefore, we work toward disk and memory models in addition to the already 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 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.

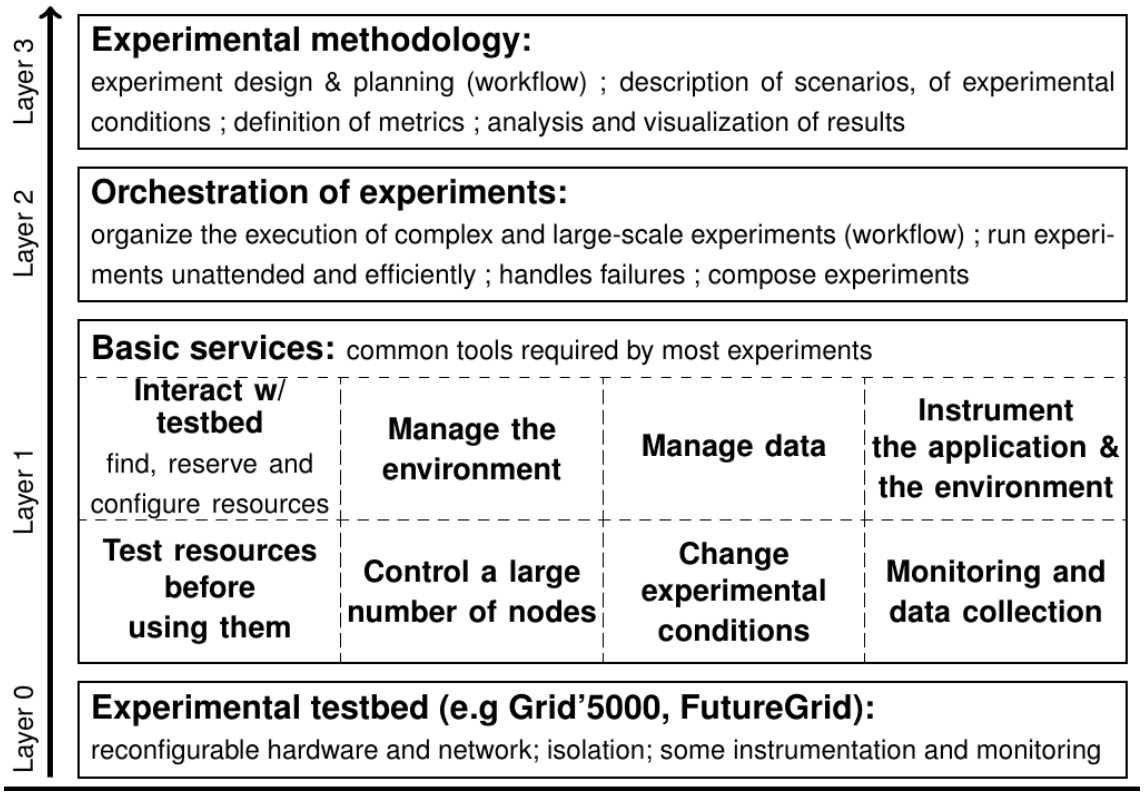

Figure 2. General structure of our project: We plan to address all layers of the experimentation stack.

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. 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, Rodrigo Campos-Catelin, Stéphane Vialle, 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.

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

5.3. 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.

5.4. 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.

5.4.1. 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-4-up, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-3, TPM-4.

5.4.2. 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.

5.4.3. SCHIaaS

Participants: Julien Gossa, Stéphane Genaud, Luke Bertot, Rajni Aron, Étienne Michon.

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-4-up, SO-4, SM-2-up, EM-2-up, SDL-2, DA-4, CD-4, MS-4, TPM-4.

5.5. Kadeploy

Participants: Luc Sarzyniec, Emmanuel Jeanvoine, Lucas Nussbaum.

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.

5.6. XPFlow

Participants: Tomasz Buchert, Lucas Nussbaum.

XPFlow is an implementation of a new, workflow-inspired approach to control of 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 Workflow Management (BPM), workflow management techniques and scientific workflows. The website of XPFlow is http://xpflow.gforge.inria.fr/. The software itself is not released to the general public yet, a fact that will be addressed during the year 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.

5.7. Grid'5000 testbed

Participants: Luc Sarzyniec, Emmanuel Jeanvoine, Émile Morel, Lucas Nussbaum.

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

In this domain we have been active on several research subjects: efficient locking interfaces, data management, asynchronism, algorithms for large scale discrete structures and the use of accelerators, namely GPU.

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

6.1.1. Efficient linear algebra on accelerators.

Participants: Sylvain Contassot-Vivier, Thomas Jost.

The PhD thesis of Thomas Jost, co-supervised by S. Contassot-Vivier and Bruno Lévy (Alice INRIA team) since January 2010, dealt with specific algorithms for GPUs, in particular linear solvers [32]. He also worked on the use of GPUs within clusters of workstations via the study of a solver of non-linear problems [30], [33], [29]. The defense of this thesis was initially planned in January 2013 but Thomas decided at the end of 2012 to stop his PhD and to leave for industry.

6.1.2. Development methodologies for parallel programming of clusters.

Participants: Sylvain Contassot-Vivier, Jens Gustedt, Stéphane Vialle.

We have conducted a particular effort in merging and synthesizing our respective experiences of parallel programming of clusters (homogeneous, heterogeneous, hybrid). This has lead to two book chapters [19] and [34] (to appear).

6.1.3. Combining locking and data management interfaces.

Participants: Jens Gustedt, Stéphane Vialle, Soumeya Leila Hernane, Rodrigo Campos-Catelin.

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. The thesis of Soumeya Hernane [12] has been defended in 2013. It extends distributed lock mechanisms and combines them with implicit data management.

A new implementation (ORWL) of our ideas of combining control and data management in C has been undertaken, see 5.2.1. In 2013, work has demonstrated its efficiency for a large variety of platforms, see [22]. 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.

Additionally, during the internship of Rodrigo Campos-Catelin, a detailed instrumentation of the ORWL library has been undertaken, and a new, less expensive strategy for cyclic FIFOs has been tested. This work will be continued with a master thesis at the university of Buenos Aires that will summarize and extend the results that were achieved during the internship.

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.

6.1.4. Discrete and continuous dynamical systems.

Participants: Sylvain Contassot-Vivier, Marion Guthmuller.

The continuous aspect of dynamical systems has been intensively studied through the development of asynchronous algorithms for solving PDE problems. In past years, we have focused our studies on the interest of GPUs in asynchronous algorithms [29]. Also, we have investigated the possibility to insert periodic synchronous iterations inside the asynchronous scheme in order to improve the convergence detection delay. This is especially interesting on small/middle sized clusters with efficient networks. The SimGrid environment has been used to validate and evaluate load balancing strategies in parallel iterative algorithms on large scale systems [28].

In 2011, the PhD thesis of Marion Guthmuller, supervised by M. Quinson and S. Contassot-Vivier, has started on the subject of model-checking distributed applications inside the SimGrid simulator [31]. The expected results of that work may provide a very interesting tool for studying dynamical systems expressed under the form of a distributed application.

6.2. Transparent Resource Management for Clouds

Participants: Julien Gossa, Rajni Aron, Stéphane Genaud, Étienne Michon, Marc-Eduard Frîncu.

6.2.1. Provisioning strategies.

Our main achievement was the design of one comprehensive provisioning meta-strategy. This meta-strategy only use one parameter as a deadline given by the user. Contrary to other deadline-based provisioning strategies, our meta-strategy is able to combine any provisioning strategy in order to optimize the cost while meeting the deadline. This is achieved through simulation of cost and makespan of every available strategy thanks to SCHIaaS5.4.3. It allows to apply the most inexpensive strategy as long as possible, before progressively switching to more expensive strategy when the deadline becomes closer.

The next step is to asses this meta-strategy among an important set of applications and platforms, both in real environments and simulation. The data are currently gathered and analyzed, and we should be able to draw conclusions soon.

6.2.2. User workload analysis.

We have conducted one broad study about workflows execution on the cloud, from both the theoretical and experimental point of view. In this study, we tried to discover causalities between the characteristics of workflows and the performances of provisioning strategies. We concluded that, except very peculiar cases, no causality can be identified. That is why we decided to make use of simulation to predict the strategies performances.

This predictive process is now integrated as a module of our cloud broker. It can be invoked by a user to help him decide which strategy should be used before any actual resource leasing.

We are now convinced that workload analysis is not a suitable approach because of its lack of generality.

6.2.3. Experimentations.

Given the very large consumption of CPU hours, the above work was supported mostly by simulation. We have assessed the gap between the performances of real executions on a private cloud and simulation. The latter proved to be very accurate, predicting almost perfectly the cost and makespan of every strategy on a wide range of workloads.

However, we have also shown that the simulation can be very sensitive to user defined input parameters (such as task runtimes) and may be mislead in borderline cases. Identifying the pitfalls and limitations of the simulation is very important and should end up in recommendations for a wise interpretation of simulation results.

We have also extended the range of experimentations to assess our simulator. First, we have extended the set of simulations with new applications, mostly workflows that are both generated and real applications (i.e. Montage). Second, we have conducted intensive experimentations on new platforms (i.e. Bonfire). The experimental data we have recently gathered in both cases is to be analyzed to further validate our approach.

6.3. Experimental methodologies for the evaluation of distributed systems

This year, M. Quinson defended his Habilitation on the experimental methodologies of distributed systems [13]. This concludes 10 years of research on this topic (including the elements presented in this section), and paves the road of future research.

6.3.1. Simulation and dynamic verification

6.3.1.1. MPI simulation

Participants: Martin Quinson, Paul Bédaride, Marion Guthmuller.

We continued our long-term effort toward the simulation of HPC application within SimGrid. We slightly increased the API coverage of our reimplementation of MPI on top of SimGrid, and proposed a new model of the network performance for MPI applications on top of Ethernet TCP networks. This model combines the advantages of flow-based networks for large data transfers as previous SimGrid network models, but also leverage algorithmic performance models extending the classical LogP models. As shown in [16], these models greatly improve the realism of MPI simulations, enabling the prediction of the performance of a non-trivial application in great details.

6.3.1.2. Dynamic verification and SimGrid

Participants: Marion Guthmuller, Martin Quinson, Gabriel Corona.

This year, our work toward the verification of liveness properties within SimGrid became fully functional thanks to the PhD work of M. Guthmuller. This relies on a system-level introspection mechanism allowing the model checker to finely explore the state of the verified programs. This is mandatory to detect the execution cycles that constitute the counter examples to liveness properties. This introspection mechanism is also used to implement a new reduction mechanism that can mitigate the state space explosion problem. A publication presenting these results is currently under review.

6.3.1.3. SimGrid framework improvement

Participants: Paul Bédaride, Martin Quinson, Gabriel Corona.

We rolled out a new major version of the SimGrid framework to our users. It contains both the HPC network models used to improve the prediction of MPI applications and all of our developments toward the dynamic verification of distributed applications. We also improved further the usability of our framework, that is now properly integrated within the Debian Linux distribution.

The next release is already underway, with a proper integration of the work from our partners on virtual machines and with a full reimplementation of the simulation kernel in C++ for a better modularity.

6.3.1.4. Formal Verification of Distributed Algorithms

Participants: Esteban Campostrini, Martin Quinson, Stephan Merz.

M. Quinson co-advised an internship with S. Merz (project-team Veridis) on the formal verification of distributed algorithm. The goal was to push further the PlusCal algorithmic language and its compiler to TLA^+ on which we are working since several years within the Veridis team.

We wanted to explore some hard problem raised by the verification of distributed protocol, such as how to represent timeout errors in verification settings where the time is not present. We think that this could be modeled somehow similarly to fairness properties, but more work is needed in this topic for a definitive answer.

6.3.2. Experimentation on testbeds and production facilities, emulation

6.3.2.1. Distem improvements: scalability and matrix-based inter-nodes latencies

Participants: Ahmed Bessifi, Emmanuel Jeanvoine, Lucas Nussbaum.

(For context, see sections 3.3 and 5.3.)

Following our PDP'13 publication[18], we focused on improving Distem's scalability. First, on the Distem engine side, we parallelized the startup of physical nodes and virtual nodes, and added support for BTRFS snapshots to enable starting a very large number of virtual nodes with their own filesystems. Second, during the internship of Ahmed Bessifi we investigated several networking issues causing problems with large-scale experiments (over 4000 virtual nodes). The resulting improvements to ARP parameters tunings were integrated in Distem 0.8, and enabled network-intensive experiments with up to 8000 virtual nodes. We plan to publish those results in early 2014.

In the context of the AEN HEMERA project, we worked with Trong-Tuan Vu (EPI DOLPHIN, Inria Lille Nord Europe) to add support for specifying inter-nodes latencies using a matrix. This is especially useful for experiments on load-balancing and locality.

6.3.2.2. Evaluating load balancing HPC runtimes with Distem

Participants: Joseph Emeras, Emmanuel Jeanvoine, Lucas Nussbaum.

(For context, see sections 3.3 and 5.3.)

We aim at demonstrating the suitability of Distem to evaluate Exascale and Cloud runtime environments providing load balancing and fault tolerance features. In that context, we reproduced some experiments published at CCGrid'2013 on Charm++ load balancers. Preliminary results are promising, and we hope that this will lead to collaborations with runtime developers.

A publication presenting how Distem to test HPC runtimes (scalability, fault tolerance and load balancing capabilities) is in the works.

6.3.2.3. Further improvements to XPFlow

Participants: Tomasz Buchert, Lucas Nussbaum, Jens Gustedt.

(For context, see sections 3.3 and 5.6.)

We strengthened our XPFlow experiment control system using several sets of experiments, including experiments on the OpenStack IaaS Cloud stack on hundreds of Grid'5000 nodes.

A publication describing XPFlow was submitted to CCGrid'2014[21].

6.3.2.4. Further improvements to Kadeploy

Participants: Luc Sarzyniec, Emmanuel Jeanvoine, Lucas Nussbaum.

(For context, see sections 3.3 and 5.5.)

We continued the development of Kadeploy:

- The support for multi-partition images was added;
- The communication interface between the Kadeploy server and the Kadeploy client was completely rewritten to use a REST API;
- A test framework, integrated with Inria's Continuous Integration facility, was added.

Two new Kadeploy releases were published during 2013, including those changes.

6.3.2.5. Grid'5000

Participants: Sébastien Badia, Luc Sarzyniec, Émile Morel, Lucas Nussbaum.

(For context, see sections 3.3 and 5.7.)

The team continued to support Grid'5000. Highlights of 2013 include:

- Lucas Nussbaum is now a member of the *Bureau* and *Comité d'Architectes* of GIS Grid'5000. In the context of the *Comité d'Architectes*, he led the writing on several internal documents (on possible evolutions of the testbed).
- An article describing Grid'5000's support for experiments on IaaS Clouds[15] was published at the *Testing The Cloud* workshop.
- A new cluster, *graphite*, was installed in Nancy.

6.3.3. Convergence and co-design of experimental methodologies

6.3.3.1. Practical study on combining experimental methodologies

Participants: Maximiliano Geier, Lucas Nussbaum, Martin Quinson.

During an internship, we explored how simulation, emulation and experimentation on Grid'5000 could be combined in practice. Starting with a simple question on a particular system, we used a representative framework for each methodology: SimGrid for simulation, Distem for emulation and Grid'5000 for experimentation, and described our experiments using the workflow logic provided by the XPFlow tool. We identified a set of pitfalls in each paradigm that experimenters may encounter regarding models, platform descriptions and others. We proposed a set of general guidelines to avoid these pitfalls. We showed these guidelines may lead to accurate simulation results. Finally, we provided some insight to framework developers in order to improve the tools and thus facilitate this convergence.

The results of this work were published at the *WATERS* workshop[17].

6.3.3.2. Organization of an event on reproducible research

Participant: Lucas Nussbaum.

We organized Realis, an event aiming at testing the experimental reproducibility of papers submitted to Compas'2013. Associated to the Compas'13 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.

7. Partnerships and Cooperations

7.1. Regional Initiatives

CPER MISN, EDGE project (2010-2013, 518k€). M. Quinson and L. Nussbaum are leading a project of the regional CPER contract, called *Expérimentations et calculs distribués à grande échelle* (EDGE). It focuses on maintaining and improving the local Grid'5000 infrastructure, and animating both the research on experimental grids and the research community using these facilities. More information is available at http://misn.loria.fr/spip.php?rubrique8.
Other partners: EPI CARAMEL, VERIDIS

Lorraine Region (2011-2013, 30k€). The project *"Systèmes dynamiques : étude théorique et application à l'algorithmique parallèle pour la résolution d'équation aux dérivées partielles"* lead by S. Contassot-Vivier is the sequel of his research on dynamical systems and consists in designing more efficient algorithmic schemes for parallel iterative solvers. This project is closely linked to the collaboration with the Lemta as the real case application provided by F. Asllanaj will be the target of our future developments in this field.

7.2. National Initiatives

7.2.1. 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 (IFP, EDF, CETIAT). We will contribute to the design and development of the platform.

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 projects, which allows the improvement of our tool even further and sets it as the reference in its domain (see Section 6.3.1).

7.2.2. Inria financed projects and clusters

AEN Hemera (2010-2013, 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 (2007-2015, 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 Kadeploy (2011-2013, AlGorille is the only partner, 90k€) focuses on the Kadeploy software, a tool for efficient, scalable and reliable cluster deployment. It is used on several clusters at INRIA and playing a key role on the Grid'5000 testbed. This ADT allows the continuation of the development to improve its performance, reliability and security, and aims at a larger distribution to industry and other INRIA platforms with similar needs.

ADT Solfége (2011-2013, AlGorille is the only partner, 100k€), for *Services et Outils Logiciels Facilitant l'Experimentation à Grande Échelle* aims at developing or improving a tool suite for experimentation at large scale on testbeds such as Grid'5000. Specifically, we will work on control of a large number of nodes, on data management, and on changing experimental conditions with emulation. E. Jeanvoine (SED) is delegated in the AlGorille team for the duration of this project.

ADT Cosette (2013-2015, 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 MC (2012-) 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, will strengthen the collaboration within this project.

7.3. European Initiatives

7.3.1. FP7 Projects

7.3.1.1. 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.

7.4. International Research Visitors

7.4.1. Visits of International Scientists

7.4.1.1. Internships

Maximiliano Geier

Subject: Leveraging multiple experimentation methodologies to study P2P broadcast

Date: from Sep 2012 until Mar 2013

Institution: University of Buenos Aires (Argentina)

Ahmed Bessifi

Subject: Reliability and Scalability improvements in Kadeploy

Date: from Mar 2013 until Aug 2013

Institution: Université de Tunis El Manar - Faculté des Sciences (Tunisia)

Luis Esteban Campostrini

Subject: Formal Verification of Distributed Algorithms

Date: from May 2013 to Oct 2013

Institution: Universidad National de Rosario (Argentina)

Rodrigo Campos

Subject: Ordered Read-Write Locks on Multicore Architectures

Date: from Mar 2013 until Aug 2013

Institution: University of Buenos Aires (Argentina)

8. Dissemination

8.1. Scientific Animation

Since October 2001, J. Gustedt is Editor-in-Chief of the journal *Discrete Mathematics and Theoretical Computer Science* (DMTCS). He is member of the recruiting committee for PhDs and postdocs of Inria research center and has been member in a recruitment committee for a position at university of Franche Comté in 2013. For the French evaluation organization AERES, J. Gustedt has been one of the evaluators of the LRI Lab near Paris.

For SUPÉLEC, S. Vialle is now leader of the IDMaD research group (focussed on distributed computing and big data) inside the IMS team.

Since 2010, J. Gossa serves as expert for the French ministry of science and education and is in charge of reviewing industrial R&D expenses and *Credit Impôt Recherche* reports. He is also a member of the following boards: IUT Robert Schuman, ICube laboratory computer science department, and University of Strasbourg.

Since march 2013, S. Genaud is dean of the ENSIIE engineering school in Computer Science.

Since 2013, S. Contassot-Vivier is director of the Computer Science Department of the Faculty of Science at Université Lorraine. Since 2011, he also serves as an expert for the French ministry of education and research in the DGRI/MEI mission and is in charge of reviewing academic projects between French and foreign teams.

L. Nussbaum is appointed as an expert on production facilities by the direction of the Inria Nancy–Grand Est center.

M. Quinson has served as program committee member of the 28th ACM/IEEE International Parallel & Distributed Processing Symposium (IPDPS'14), of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH, co-organized with ACM SIGSIM), of the 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS'13), and of the 4th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS'13).

M. Quinson served as a member of the recruitment committees for associate professor in computer science at University of Lorraine and for junior researcher at Inria Nancy.

M. Quinson is elected of the AM2I board, in charge of voting the budget for 6 laboratories at Université de Lorraine (450 researchers in computer science, mathematics and control theory).

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

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

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

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

Licence: Sylvain Contassot-Vivier, "Informatique graphique", 42 ETD, Licence informatique (L1), Université de Lorraine, France

Licence: Sylvain Contassot-Vivier, "Algorithmique 3", 12 ETD, Licence informatique (L2), Université de Lorraine, France

Licence: Sylvain Contassot-Vivier, "Algorithmique 4", 32 ETD, Licence informatique (L3), Université de Lorraine, France

Licence, Stephane Vialle, "modèle de programmation", 21 TD, L1, SUPELEC, France

MIAGE Nancy: Marion Guthmuller, "Réseaux", 50 ETD, M1, Université de Lorraine, France

Master: Sylvain Contassot-Vivier, "Algorithmique répartie et systèmes distribués", 45 ETD, M1, Université de Lorraine, France

Telecom Nancy: Sylvain Contassot-Vivier, "Algorithmique des systèmes parallèles et distribués", 32 ETD, 2ème année (M1), université de Lorraine, France

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

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

ESSTIN Nancy: Sylvain Contassot-Vivier, "Calcul haute performance", 42 ETD, 4ème année (M1), université de Lorraine, France

Master, Stephane Vialle, "systèmes d'information", 39 TD, M1, SUPELEC, France

Master, Stephane Vialle, "calcul haute performance", 62 TD, M2, SUPELEC, France

Engineering School, Stéphane Genaud, "systèmes informatiques et réseaux", 72 TD, niveau L3, ENSIIE, France

Engineering School, Stéphane Genaud, "middleware", 32 TD, niveau M1, ENSIIE, France

Master, Stéphane Genaud, “parallélisme, systèmes distribués et grilles”, 21 TD, M2, University of Strasbourg, France

Master, Stephane Vialle, “parallélisme, systèmes distribués et grilles”, 21 TD, M2, University of Strasbourg, France

Master, Stephane Vialle, “systèmes distribués et grilles”, 28 TD, M2, University of Strasbourg, 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 stopped: Thomas Jost, *Solveurs linéaires creux sur GPU*, started in Jan 2010, Bruno Lévy & Sylvain Contassot-Vivier

PhD: Soumeya Hernane, *Models and algorithms for consistent data sharing in high performance parallel and distributed computing*, defense Jul 2013, Jens Gustedt & Mohamad Benyettou

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

8.2.3. Juries

Jens Gustedt was reviewer and member of the jury of the thesis of Lilia Ziane Khodja at Franche-Comté university.

Martin Quinson was a member of the jury of the thesis of Rafael Ferreira Da Silva at INSA-Lyon.

Stéphane Genaud was a member of the jury of thesis of Imen Chakroun at Lille 1 university.

8.3. Popularization

Jens Gustedt is regularly blogging about efficient programming in particular 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.

L. Nussbaum currently serves as the (elected) Debian Project Leader since April 2013.

M. Quinson develops a pedagogic platform in collaboration with G. Oster (Score team of Inria Nancy Grand Est). 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. It is available from its page.

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 2014.

M. Quinson participated to several events toward the popularization of computer science (either as a speaker or as a co-organizer), targeting either kids and pupils (Telecom Nancy in November), students (Inria in March), maths teachers (APMEP Lorraine in March), CS teachers (SIF-ISN day in June), or all public (Fête de la science in November).

M. Quinson serves on the editorial board of the Interstices website of Inria for the popularization of computer science.

9. Bibliography

Major publications by the team in recent years

[1] T. BUCHERT, L. NUSSBAUM, J. GUSTEDT. *Methods for Emulation of Multi-Core CPU Performance*, in "13th IEEE International Conference on High Performance Computing and Communications (HPCC-2011)", Banff, Canada, IEEE, September 2011, pp. 288 - 295 [*DOI :* 10.1109/HPCC.2011.45], http://hal.inria.fr/inria-00535534/en

[2] L.-C. CANON, O. DUBUISSON, J. GUSTEDT, E. JEANNOT. *Defining and Controlling the Heterogeneity of a Cluster: the Wrekavoc Tool*, in "Journal of Systems and Software", 2010, vol. 83, n^{o} 5, pp. 786-802 [*DOI :* 10.1016/J.JSS.2009.11.734], http://hal.inria.fr/inria-00438616/en

[3] H. CASANOVA, A. LEGRAND, M. QUINSON. *SimGrid: a Generic Framework for Large-Scale Distributed Experiments*, in "10th IEEE International Conference on Computer Modeling and Simulation - EUROSIM / UKSIM 2008", Royaume-Uni Cambrige, IEEE, 2008, http://hal.inria.fr/inria-00260697/en/

[4] P.-N. CLAUSS, J. GUSTEDT. *Iterative Computations with Ordered Read-Write Locks*, in "Journal of Parallel and Distributed Computing", 2010, vol. 70, n^{o} 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] A. H. GEBREMEDHIN, J. GUSTEDT, M. ESSAÏDI, I. GUÉRIN LASSOUS, J. A. TELLE. *PRO: A Model for the Design and Analysis of Efficient and Scalable Parallel Algorithms*, in "Nordic Journal of Computing", 2006, vol. 13, pp. 215-239, http://hal.inria.fr/inria-00000899/en/

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

[8] 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

[9] T. KLEINJUNG, L. NUSSBAUM, E. THOMÉ. *Using a grid platform for solving large sparse linear systems over GF(2)*, in "11th ACM/IEEE International Conference on Grid Computing (Grid 2010)", Brussels, Belgium, October 2010, http://hal.inria.fr/inria-00502899/en

[10] C. MAKASSIKIS, V. GALTIER, S. VIALLE. *A Skeletal-Based Approach for the Development of Fault-Tolerant SPMD Applications*, in "The 11th International Conference on Parallel and Distributed Computing, Applications and Technologies - PDCAT 2010", Wuhan, China, December 2010 [*DOI :* 10.1109/PDCAT.2010.89], http://hal.inria.fr/inria-00548953/en

[11] M. QUINSON, G. OSTER. , *The Java Learning Machine: A Learning Management System Dedicated To Computer Science Education*, Inria, February 2011, n^{o} RR-7537, http://hal.inria.fr/inria-00565344/en

Publications of the year

Doctoral Dissertations and Habilitation Theses

[12] S. HERNANE. , *Modèles et algorithmes de partage de données cohérents pour le calcul parallèle et distribué à haut débit.*, Université de Lorraine and Université des Sciences et de la Technologie d'Oran, June 2013, http://hal.inria.fr/tel-00919272

[13] M. QUINSON. , *Méthodologies d'expérimentation pour l'informatique distribuée à large échelle*, Université de Lorraine, March 2013, Habilitation à Diriger des Recherches, http://hal.inria.fr/tel-00927316

Articles in International Peer-Reviewed Journals

[14] E. JEANVOINE, L. SARZYNIEC, L. NUSSBAUM. *Kadeploy3: Efficient and Scalable Operating System Provisioning for Clusters*, in "USENIX ;login:", February 2013, vol. 38, n^{o} 1, pp. 38-44, http://hal.inria.fr/hal-00909111

International Conferences with Proceedings

[15] S. BADIA, A. CARPEN-AMARIE, A. LÈBRE, L. NUSSBAUM. *Enabling Large-Scale Testing of IaaS Cloud Platforms on the Grid'5000 Testbed*, in "TTC - 1st International Workshop on Testing The Cloud, co-located with ISSTA 2013", Lugano, Switzerland, ACM, July 2013, pp. 7-12 [*DOI :* 10.1145/2489295.2489298], http://hal.inria.fr/hal-00907888

[16] P. BEDARIDE, A. DEGOMME, S. GENAUD, A. LEGRAND, G. MARKOMANOLIS, M. QUINSON, M. STILLWELL, F. SUTER, B. VIDEAU. *Toward Better Simulation of MPI Applications on Ethernet/TCP Networks*, in "PMBS13 - 4th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems", Denver, United States, November 2013, http://hal.inria.fr/hal-00919507

[17] M. GEIER, L. NUSSBAUM, M. QUINSON. *On the Convergence of Experimental Methodologies for Distributed Systems: Where do we stand?*, in "WATERS - 4th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems", Paris, France, July 2013, http://hal.inria.fr/hal-00907887

[18] L. SARZYNIEC, T. BUCHERT, E. JEANVOINE, L. NUSSBAUM. *Design and Evaluation of a Virtual Experimental Environment for Distributed Systems*, in "PDP2013 - 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing", Belfast, United Kingdom, IEEE, February 2013, pp. 172 - 179 [*DOI :* 10.1109/PDP.2013.32], http://hal.inria.fr/hal-00724308

Scientific Books (or Scientific Book chapters)

[19] S. CONTASSOT-VIVIER, S. VIALLE, J. GUSTEDT. *Development methodologies for GPU and cluster of GPUs*, in "Development methodologies for GPU and cluster of GPUs", R. COUTURIER (editor), Chapman & Hall/CRC, 2013, http://hal.inria.fr/hal-00905790

Research Reports

[20] P. BEDARIDE, S. GENAUD, A. DEGOMME, A. LEGRAND, G. MARKOMANOLIS, M. QUINSON, M. STILLWELL, F. SUTER, B. VIDEAU. , *Improving Simulations of MPI Applications Using A Hybrid Network Model with Topology and Contention Support*, Inria, May 2013, n° RR-8300, 22 p. , http://hal.inria.fr/hal-00821446

[21] T. BUCHERT, L. NUSSBAUM, J. GUSTEDT. , *A workflow-inspired, modular and robust approach to experiments in distributed systems*, Inria, November 2013, n° RR-8404, http://hal.inria.fr/hal-00909347

[22] J. GUSTEDT, S. VIALLE, P. MERCIER. , *Resource Centered Computing delivering high parallel performance*, Inria, December 2013, n° RR-8433, http://hal.inria.fr/hal-00921128

[23] A. ROUSSEAU, A. DARNAUD, B. GOGLIN, C. ACHARIAN, C. LEININGER, C. GODIN, C. HOLIK, C. KIRCHNER, D. RIVES, E. DARQUIE, E. KERRIEN, F. NEYRET, F. MASSEGLIA, F. DUFOUR, G. BERRY, G. DOWEK, H. ROBAK, H. XYPAS, I. ILLINA, I. GNAEDIG, J. JONGWANE, J. EHREL, L. VIENNOT, L. GUION, L. CALDERAN, L. KOVACIC, M. COLLIN, M.-A. ENARD, M.-H. COMTE, M. QUINSON, M. OLIVI, M. GIRAUD, M. DORÉMUS, M. OGOUCHI, M. DROIN, N. LACAUX, N. ROUGIER, N. ROUSSEL, P. GUITTON, P. PETERLONGO, R.-M. CORNUS, S. VANDERMEERSCH, S. MAHEO, S. LEFEBVRE, S. BOLDO, T. VIÉVILLE, V. POIREL, A. CHABREUIL, A. FISCHER, C. FARGE, C. VADEL, I. ASTIC, J.-P. DUMONT, L. FÉJOZ, P. RAMBERT, P. PARADINAS, S. DE QUATREBARBES, S. LAURENT. , *Médiation Scientifique : une facette de nos métiers de la recherche*, March 2013, 34 p. , http://hal.inria.fr/hal-00804915

Scientific Popularization

[24] M. QUINSON, J.-C. BACH. *L'informatique nomade, c'est la liberté !*, in "Interstices", February 2013, http://hal.inria.fr/hal-00794187

Other Publications

[25] H. CASANOVA, A. GIERSCH, A. LEGRAND, M. QUINSON, F. SUTER. , *SimGrid: a Sustained Effort for the Versatile Simulation of Large Scale Distributed Systems*, 2013, 4 p. , submission to WSSSPE'13, http://hal.inria.fr/hal-00926437

[26] E. JEANVOINE, L. SARZYNIEC, L. NUSSBAUM. *Efficient and Scalable OS Provisioning with Kadeploy 3*, in "JRES - Journées Réseaux - 2013", Montpellier, France, December 2013, JRES - Journées Réseaux - 2013, http://hal.inria.fr/hal-00920358

[27] L. NUSSBAUM, P. NEYRON, O. RICHARD, E. JEANVOINE. *Grid'5000: A Production-grade Testbed for Experiment-driven Computer Science on HPC and Clouds*, in "Inria Booth at SC'13", Denver, United States, November 2013, Inria Booth at SC'13, http://hal.inria.fr/hal-00920389

References in notes

[28] J. M. BAHI, S. CONTASSOT-VIVIER, A. GIERSCH. *Load balancing in dynamic networks by bounded delays asynchronous diffusion*, in "10th International Meeting on High Performance Computing for Computational Science", Berkeley, États-Unis, 2010, 31 p. , An extended version is to appear in LNCS, http://hal.archives-ouvertes.fr/hal-00547300/en/

[29] S. CONTASSOT-VIVIER, T. JOST, S. VIALLE. *Impact of Asynchronism on GPU Accelerated Parallel Iterative Computations*, in "Applied Parallel and Scientific Computing", Reykjavík, Islande, K. JÓNASSON

(editor), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, vol. 7133, pp. 43-53 [*DOI :* 10.1007/978-3-642-28151-8_5], http://hal-supelec.archives-ouvertes.fr/hal-00685153

[30] S. CONTASSOT-VIVIER, S. VIALLE, T. JOST. *Optimizing computing and energy performances on GPU clusters: experimentation on a PDE solver*, in "Proceedings of the COST Action IC0804 on Energy Efficiency in Large Scale Distributed Systems 1st Year", Passau, Allemagne, J.-M. PIERSON, H. HLAVACS (editors), IRIT, 2010, pp. 50-54, http://hal-supelec.archives-ouvertes.fr/hal-00517374/en/

[31] M. GUTHMULLER. *State equality detection for implementation-level model-checking of distributed applications*, in "18th International Symposium on Formal Methods - Doctoral Symposium", Paris, France, August 2012, http://hal.inria.fr/hal-00758351

[32] T. JOST, S. CONTASSOT-VIVIER, S. VIALLE. *Solveur linéaire sur GPU*, in "Journée jeunes chercheurs sur les Multiprocesseurs et Multicoeurs", Paris, France, June 2009, http://hal.inria.fr/inria-00430529

[33] S. VIALLE, S. CONTASSOT-VIVIER, T. JOST. *Optimizing computing and energy performances in heterogeneous clusters of CPUs and GPUs*, in "Handbook of Energy-Aware and Green Computing", I. AHMAD, S. RANKA (editors), Chapman & Hall/CRC, 2011, http://hal.inria.fr/hal-00643938

[34] S. VIALLE, S. CONTASSOT-VIVIER. *Optimization methodology for Parallel Programming of Homogeneous or Hybrid Clusters*, in "Patterns for parallel programming on GPUs", F. MAGOULÉS (editor), Saxe-Coburg Publications, 2014, to appear