IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2012

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: January 01, 2007 .

1. Members

Research Scientist

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

Faculty Members

Sylvain Contassot-Vivier [Professor, Université de Lorraine, HdR]
Lucas Nussbaum [Associate Professor, Université de Lorraine]
Martin Quinson [Associate Professor, Université de Lorraine; temporary assignment as Inria junior researcher since Oct 2011]
Constantinos Makassikis [Temporary Assistant Professor, Université de Lorraine, until Sep 2012]

External Collaborators

Stéphane Genaud [Associate Professor, Univ. Strasbourg, HdR]
Julien Gossa [Associate Professor, Univ. Strasbourg]
Stéphane Vialle [Professor, SUPÉLEC Metz Campus, HdR]

Engineers

Emmanuel Jeanvoine [Engineer, SED INRIA Nancy – Grand Est, ADT Solfége]
Sébastien Badia [engineer, INRIA, CPER MISN, thème EDGE]
Paul Bédaride [engineer, ANR SONGS since Oct 2012]
Émile Morel [engineer, INRIA ADT ALADDIN-G5K since Oct 2012]
Tinaherinantenaina Rakotoarivelo [engineer, INRIA ADT Aladdin-G5K, until Nov 2012]
Luc Sarzyniec [engineer, INRIA ADT Kadeploy since Oct 2011]

PhD Students

Soumeya Leila Hernane [teaching assistant UST Oran, Algeria, since Oct 2007]
Thomas Jost [since Oct 2009]
Wilfried Kirschenmann [EDF R&D (Clamart, France) and SUPÉLEC, since Jan 2009]
Marion Guthmuller [since Oct 2011]
Tomasz Buchert [since Oct 2011]

Post-Doctoral Fellow

Christophe Thiéry [Post-Doc ANR USS-SimGrid until Feb 2012]

Administrative Assistant

Isabelle Herlich [INRIA]

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.

applications ⇐
middleware
services ⇐
infrastructure

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.

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 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.

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 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.

2.2. Highlights of the Year

- Our team (composed of Luc Sarzyniec, Sébastien Badia, Emmanuel Jeanvoine and Lucas Nussbaum) won the **best challenge entry award during the Grid'5000 winter school**. We successfully demonstrated the deployment of 4500 virtual machines using Kadeploy3 in less than an hour. An earlier iteration of this work was selected as a **finalist of the SCALE challenge, held with CCGrid'2013**.

3. Scientific Foundations

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

During the next years, we will continue to design resource provisioning strategies for cloud clients. Given the extremely large offer of resources by public or private clouds, users need software assistance to make provisioning decisions. Our goal is to gather our strategies into a **cloud resource broker** which will handle the workload of a user or of a community of users as a multi-criteria optimization problems. The notions of resource usage, scheduling, provisioning and task management have to be 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: the new prevalent optimization criterion is the financial cost of the computation.

3.2.1. Provisioning strategies

Future work includes the design of new strategies to reuse already leased resources, or switch to less powerful and cheaper resources. On one hand, some economic models proposed by cloud providers may involve a complex cost-benefit analysis for the client which we want to address. On the other hand, these economic models incur additional costs, e.g for data storage or transfer, which must 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 user workload analysis. Characterizing the workload may help to anticipate the resource provisioning, and hence improve the scheduling.

3.2.3. Experimentations

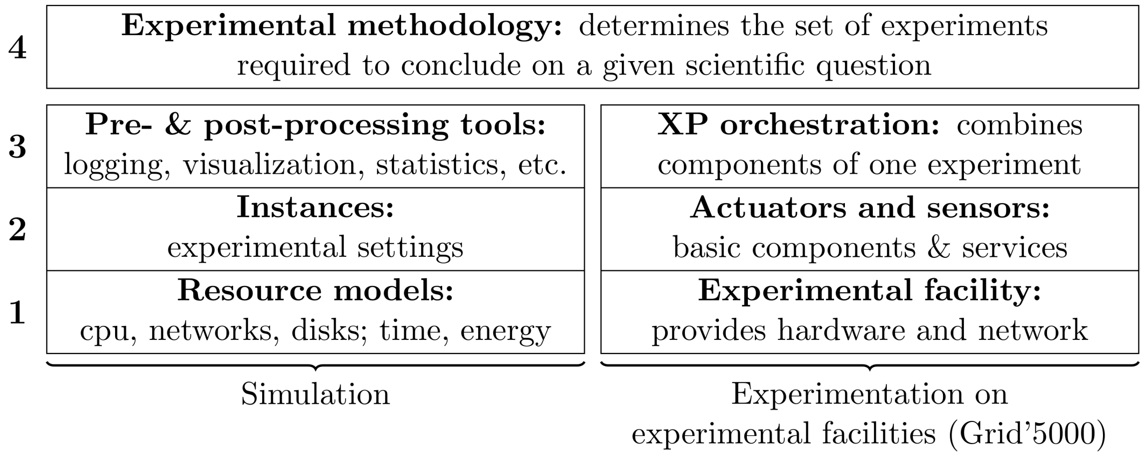
Given the very large consumption of CPU hours, the above strategies will first be tested mostly through simulation. Therefore, we will closely work with the members of the Experimental methodologies axis to co-design the cloud interface and the underlying models. Furthermore, we will assess the gap between the performances on simulation and both public and private cloud. This work will take place inside the Cloud work package of the SONGS ANR project.

3.2.4. HPC on clouds

Clouds are not suitable to run massive HPC applications. However, it might be interesting to use them as cheap HPC platform for occasional or one shot executions. This will be investigated with the Structuring Applications axis and in collaboration with the LabEx IRMIA and the CALVI team.

3.3. Experimental methodologies for the evaluation of distributed systems

We strive at designing a comprehensive set of solutions for experimentation on distributed systems by working on several methodologies (simulation, direct execution on experimental facilities, emulation) and by leveraging the convergence opportunities between methodologies (shared interfaces, validation combining several methodologies).



4	**Experimental methodology:** determines the set of experiments required to conclude on a given scientific question	
3	**Pre- & post-processing tools:** logging, visualization, statistics, etc.	**XP orchestration:** combines components of one experiment
2	**Instances:** experimental settings	**Actuators and sensors:** basic components & services
1	**Resource models:** cpu, networks, disks; time, energy	**Experimental facility:** provides hardware and network
	Simulation	Experimentation on experimental facilities (Grid'5000)

Figure 2. Our experimentation methodology, encompassing both simulation and experimental facilities.

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. Recent, we increased its audience to target the P2P research community in addition to the one on grid scheduling. It now allows **precise simulations of millions of nodes** using a single computer.

In the future, we aim at extending further 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 the ones developed in our team to structure applications, in particular ORWL. In the near future, we also consider using our toolbox to give **online performance predictions to the runtimes**. It would allow these systems to improve their adaptability to the changing performance conditions experienced on the platform.

We recently integrated a model checking kernel in our tool 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 using direct execution on testbeds and production facilities.

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, see Figure 3,

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, ...and investigate if altering memory and disk performance is possible. Other goals are to scale the tool up to 20000 virtual nodes and to improve its 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. Our approach is based on the reuse of lessons learned in the field of Business Process Management (BPM), with the design of a workflow-based experiment control engine. This is part of an ongoing collaboration with EPI SCORE (INRIA Nancy Grand Est), which has already yield promising preliminary results [15], [28].

3.3.3. Exploring new scientific objects.

We aim at addressing different kinds of distributed systems (HPC, Cloud, P2P, Grid) using the same experimental approaches. Thus a key requirement for our success is to build sufficient knowledge on target distributed systems to discover and understand the final research questions that our solutions should target. In

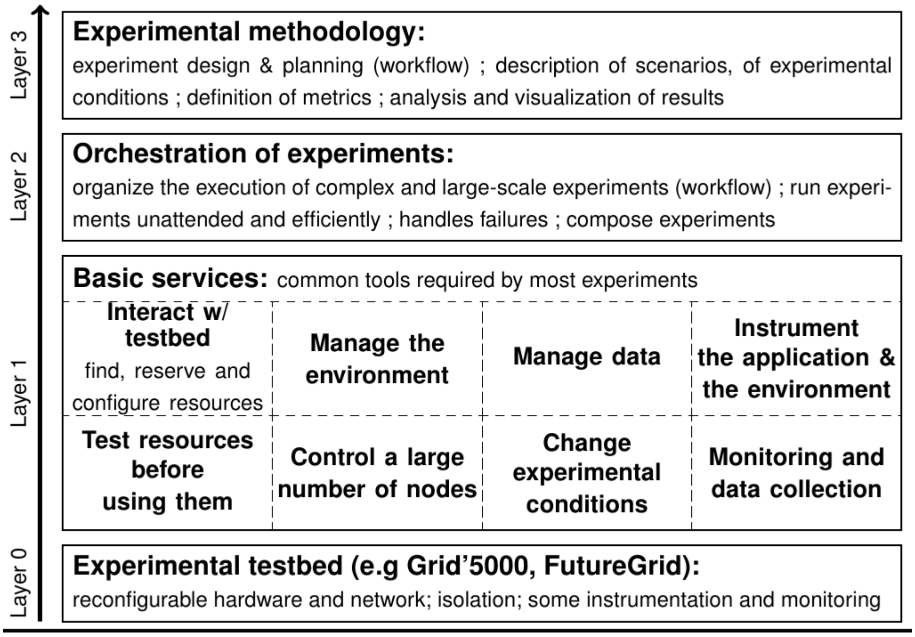

Layer 3
Experimental methodology:
experiment design & planning (workflow) ; description of scenarios, of experimental conditions ; definition of metrics ; analysis and visualization of results
Layer 2
Orchestration of experiments:
organize the execution of complex and large-scale experiments (workflow) ; run experiments unattended and efficiently ; handles failures ; compose experiments
Layer 1
Basic services: common tools required by most experiments
Interact w/ testbed
find, reserve and configure resources
Manage the environment
Manage data
Instrument the application & the environment
Test resources before using them
Control a large number of nodes
Change experimental conditions
Monitoring and data collection
Layer 0
Experimental testbed (e.g Grid'5000, FutureGrid):
reconfigurable hardware and network; isolation; some instrumentation and monitoring

Figure 3. General structure of our project: We plan to address all layers of the experimentation stack.

the framework of ANR SONGS (2012-2016), we are working closely with experts from HPC, Cloud, P2P and Grid. We are also collaborating with the production grids community, e.g. on using Grid'5000 to evaluate the gLite middleware, and with Cloud experts in the context of the OpenCloudWare project.

4. Application Domains

4.1. Scientific computation

In the context of studying GPU cluster programming as well as asynchronous algorithms, we have developed a parallel code to solve a PDE problem that is representative of scientific computations. This is a 3D version of the *advection-diffusion-reaction* problem modeling the interactions of two chemical species in shallow waters. Different versions of the application have been implemented. A first version has been the subject of extensive studies about computing and energy performance. Other versions, especially the last one using MPI+OpenMP+CUDA, are still the subject of studies.

Also in a close collaboration with Fatmir Asllanaj from the LEMTA laboratory, Nancy, France, we launched the development of a code for Radiative Transport Equations, ETR. As a first step to higher efficiency the code (originally in FORTRAN) was completely rewritten in C. Already this sequential rewrite has largely improved the time and memory efficiency of the code. The next step will be a parallelization for multi-core machines and clusters that will make this a unique tool to tackle ETRs in 2 and 3 dimensions.

4.2. Financial computation

AmerMal application is a parallel *American Option Pricer* developed in collaboration with Lokman Abbas-Turki during his PhD thesis at University Paris-Est. This pricing mathematic algorithm has been designed in order to be parallelized on clusters of GPUs. It is based on a Maillavin calculus and on Monte Carlo stochastic computations. However, the Monte Carlo trajectories are coupled, and our final parallel algorithm includes many communications between the computing nodes. Nevertheless, we achieved good scalability on our 16-nodes GPU clusters.

The initial version of AmerMal has been implemented using MPI+OpenMP+CUDA in 2010 and 2011. In 2012, we have ported this application on ORWL+CUDA, both in order to test and improve ORWL semantic and ORWL performances.

4.3. Signal processing

ParSONS is a parallel *sound sources classifier* developed at SUPELEC by Stephane Rossignol. It allows to identify different kinds of sound sources in an audio signal file (like a radio or TV archive file), and to tag different part of the file. Then it is possible to know which part of the archive file include human voice, or music, or more accurately jazz or rock music. From an algorithmic point of view, this application performs a lot of Fourrier Transform computations and a lot of IO operations.

We have developed an MPI+OpenMP version, optimizing input data file reading and overlapping these IO operations with computations (signal processing). We run some experiments on a 256-nodes dual-core PC cluster and on a 16-nodes 4-hyperthreaded-cores experimentation PC cluster, both located at SUPELEC. We identified the most efficient configurations, considering the number of input file reading processes, the number of threads per process, the number of processes per node... Finally, we have designed and implemented a deployment tool, in order to hide these configuration issues to signal processing researchers and users.

In a near future, we aim to port this application to ORWL, to experiment, improve and validate ORWL on distributed applications achieving a high ratio of IO compared to computations and classic communications.

5. Software

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. 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 is 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

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

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. SimGrid has an active user community of more than one hundred members, and is available under GPLv3 from http://simgrid.gforge.inria.fr/.
Software classification: A-4-up, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-3, TPM-4.

5.5. ORWL and P99

Participant: Jens Gustedt.

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

5.6. 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.

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.

Graphics Processing Units have evolved to fully programmable parallel vector-processor sub-systems. We have designed several parallel algorithms on GPUs, and integrated that level of parallelism into larger applications including several other levels of parallelism (multi-core, multi-node,...). In this context, we also have studied energy issues and designed some energy performance models for GPU clusters, in order to model and predict energy consumption of GPU clusters.

The PhD thesis of Wilfried Kirschenmann, has been a collaboration with EDF R&D and was co-supervised by S. Vialle and Laurent Plagne (EDF SINETICS). It has given rise to a DSEL based on C++ and to a unified generic library that adapts to multi-core CPUs, multi-core CPUs with vector units (SSE or AVX), and GPUs. This framework allows to implement linear algebra operations originating from neutronic computations, see [22].

The PhD thesis of Thomas Jost, co-supervised by S. Contassot-Vivier and Bruno Lévy (Alice INRIA team) deals with specific algorithms for GPUs, in particular linear solvers. He has also worked on the use of GPUs within clusters of workstations via the study of a solver of non-linear problems [17]. The defense of this thesis is planned in January 2013.

6.1.2. Combining locking and data management interfaces.

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; [21] 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 5.5. A first work has demonstrated its efficiency for a benchmark application [18]. Our current efforts concentrate on the 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 now able to handle such an application seamlessly and efficiently, a real alternative to home made interactions between MPI and CUDA.

6.1.3. Discrete and continuous dynamical systems.

The continuous aspect of dynamical systems has been intensively studied through the development of asynchronous algorithms for solving PDE problems. We have focused our studies on the interest of GPUs in asynchronous algorithms [17]. Also, we investigate 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. Finally, we investigate other optimizations like load balancing. For this last subject, the SimGrid environment has revealed itself to be a precious tool to perform feasibility tests and benchmarks for this kind of algorithms on large scale systems. It has been successfully used to evaluate an asynchronous load balancing algorithm [37].

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 [20]. This is also the opportunity of designing new tools to study more precisely the dynamics of discrete or continuous systems. See the simulation part in Section 6.3.2 for more details on this PhD.

6.2. Transparent resource management

6.2.1. Client-side cloud broker.

Integrating the ‘pay-as-you-go‘ pricing model commonly used in IaaS clouds is an important question which profoundly changes the assumptions for job scheduling. From the observation that in most commercial solutions the price of a CPU cycle is identical, be the CPU a fast or slow one, several schedulings may be derived for a same price but with different makespans. Hence, in a context where resources can be started on-demand, scheduling strategies must include a decision process regarding the scaling (number of resources used) of the platform and the types of resources rented over time. In [24], we have studied the impact of these two factors on classic job scheduling strategies applied to bag-of-tasks workloads. The results show that shorter makespans can be achieved through scaling at no extra cost, while using quicker CPUs largely increases the price of the computations. More importantly, we show the difficulty to predict the outcomes of such decisions, which requires to design new provisioning approaches.

6.3. Experimental Methodologies

6.3.1. Overall improvement of SimGrid

2012 was the last year of the USS-SimGrid project granted by the ANR. We thus capitalized the results of the first project by properly releasing them in the public releases. Parallel simulation is now stable enough to be used in practice by our users. In addition, the framework is now able to simulate millions of processes without any particular settings in C. The java bindings were also improved to simulate several hundred thousand processes out of the box [25].

This year was also the first year of the SONGS project, also funded by the ANR. This project is much larger that the previous one, both in funding and targets. In surface, SONGS aims at increasing the scope of the SimGrid simulation framework by enabling the Cloud and HPC scenarios in addition to the existing Grid and P2P ones. Under the hood, it aims at providing new models specifically designed for these use cases, and also provide the necessary internal hooks so that users can modify the used models by themselves.

This project is well started, with three plenary meetings and a user conference organized over the year, but no new publication resulted of this work yet. The first work toward increasing the simulation versatility, initiated last year, was published this year[14]

6.3.2. Dynamic verification of liveness properties in SimGrid

A full featured model-checker is integrated to SimGrid since a few years, but it was limited to the verification of safety properties. We worked toward the verification of liveness properties in this framework. The key challenge is to quantify the state equality at state level, adding and leveraging introspection abilities to arbitrary C programs.

This constitutes the core of the PhD thesis of M. Guthmuller, started last year. A working prototype was developed during this year, described in an initial publication [20].

6.3.3. Grid'5000 and related projects

We continued to play a key role in the Grid'5000 testbed in 2012. Lucas Nussbaum, being delegated by the executive committee to follow the work of the technical team, was heavily involved in the recent evolutions of the testbed (network weathermaps, storage management, etc.) and in other activities such as the preparation of the Grid'5000 winter school. We were also involved in a publication [33] which is a follow-up to the workshop on *Supporting Experimental Computer Science* held during SC'11, and in another publication [32] describing the recent advances on the Grid'5000 testbed in order to support experiments involving virtualization at large scale.

More specifically, our involvement in the *OpenCloudWare* project led us to design several tools that ease the deployment of Cloud stacks on Grid'5000 for experimental purposes. Those tools were also used during an internship that was co-advised with the *Harmonic Pharma* start-up, exploring how complex bio-informatics workflows could be ported to the Cloud.

On the institutional side, we will also play a central role in the *Groupement d'Intérêt Scientifique* that is currently being set up, since Lucas Nussbaum is a member of both the *bureau* and of the *comité d'architectes*.

6.3.4. Distem – DISTributed systems EMulator

In the context of ADT Solfége, we continued our work on Distem. Three releases were made over the year, with several improvements and bug fixes, including support for variable CPU and network emulation parameters during an experiment. See http://distem.gforge.inria.fr/ for more information, or our paper accepted at PDP'2013 [26].

6.3.5. Kadeploy3 – scalable cluster deployment solution

Thanks to the support of ADT Kadeploy3, many efforts were carried out on Kadeploy3. Two releases were made, including many new features (many improvements to the handling of parallel commands and to the inner automaton for more fault-tolerant deployments; use of Kexec for faster deployments) as well as bug fixes.

Kadeploy3 was featured during several events (*journée 2RCE*, *SuperComputing 2012*), and in two publications: one unsuccessfully submitted to LISA'2012 [35], one accepted in USENIX *;login:* [13].

Finally, Kadeploy3 was also the basis of submissions to the *SCALE challenge held with CCGrid'2012*, of which we were finalists, and of the winner challenge entry at *Grid'5000 winter school 2012*.

6.3.6. Business workflows for the description and control of experiments

We are exploring the use of Business Process Modelling and Management for the description and the control of complex experiments. In [28], we outlined the required features for an experiment control framework, and described how business workflows could be used to address this issue. In [27] and [15], we described our early implementation of XPFlow, a experiment control engine relying on business workflows paradigms.

6.3.7. Towards Open Science for distributed systems research

One of our long term goal on experimental methodologies would be the advance of an Open Science in the research domain of Distributed Systems. Scientific tools would be sufficiently assessed and easily combined when necessary, and scientific experiments would be perfectly reproducible. These objectives are still very ambitious for the researches targeting distributed systems.

In order to precisely evaluate the path remaining toward these goals, and try addressing some of the challenges that they pose, we currently host Maximiliano Geier as an Inria intern. While most researchers try to answer brilliant scientific questions with simple scientific methodologies, he is asked to answer a simple question (on the adaptation of the BitTorrent protocol to high bandwidth networks) using an advanced scientific methodology. We are also surveying the experimental methodology used in top tier conferences to gain further insight on this topic.

In addition, we are organizing Realis, an event aiming at testing the experimental reproducibility of papers submitted to Compas'2013. Associated to the Compas'13 conference, this workshop aims 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. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

- In 2012, SUPÉLEC had 2 contracts with Quartet Financial System about parallel and distributed applications processing flows of financial data (the first one on PC clusters, and the second on NUMA computing nodes). This industrial research collaboration is continuing in 2013.
- In 2012 SUPÉLEC has achieved an industrial contract with Thales Underwater Systems about parallelisation on GPU of sonar signal processing algorithms.
- In 2012 SUPÉLEC has achieved an industrial contract with CGGVeritas about the parallelization on GPU of seismic data decompression.
- In 2012 SUPÉLEC has started 2 contracts with EDF R&D about the development of co-simulators for electrical smart Grids, including control parallelism issues.

8. Partnerships and Cooperations

8.1. Regional Initiatives

CPER MISN, EDGE project (2010-2013, 468k€). 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.

8.2. National Initiatives

8.2.1. 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.

USS-SimGrid (2009–2012, 840k€) Martin Quinson is the principal investigator, funded by the ANR ARPEGE program. USS-SimGrid (Ultra Scalable Simulation with SimGrid) aims at improving the scalability of the SimGrid simulator to allow its use in Peer-to-Peer research in addition of Grid Computing research. The challenges to tackle included models being more scalable at the eventual price of slightly reduced accuracy, automatic instantiation of these models, tools to conduct experiments campaigns, as well as a partial parallelization of the simulator tool. This project was successfuly completed this year.

ANR SONGS (2012–2015, 1800k€) Martin Quinson is also the principal investigator of a 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 set it as the reference in its domain (see Sections 6.3.1 and 6.3.2).

8.2.2. 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.

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.

8.3. International Research Visitors

8.3.1. Visits of International Scientists

8.3.1.1. Internships

Maximiliano GEIER (09/2012 - 03/2013)

Subject: Leveraging multiple experimentation methodologies to study P2P broadcast

Institution: University of Buenos Aires (Argentina)

8.3.2. Visits to International Teams

Martin Quinson was hosted as a visiting professor at university of Hawai'i at Manoa for one month in April 2012. He was invited by Prof. Casanova to pursue the collaboration on SimGrid, originally started by Prof. Casanova.

9. Dissemination

9.1. 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 that center and has been member in three recruitment committees for university positions in 2012.

In 2011 and 2012, J. Gustedt also served as head of the *Networks, Systems and Services* department of the LORIA computer science lab and through that as a member of the directory of LORIA and of several recruiting commissions of Université de Lorraine.

Since 2005, S. Vialle is head of the RGE action ("Réseau Grand Est") of the GDR ASR of CNRS. For SUPÉLEC, S. Vialle served as head of the IMS research team (2010–2012), and 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.

Since 2011, S. Contassot-Vivier 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 was member of the organization and program committee for the Grid'5000 Winter School 2012.

M. Quinson has served as program committee member of the 27th ACM/IEEE International Parallel & Distributed Processing Symposium (IPDPS'13).

We organized the first Realis event this year. Associated to the Compas'13 conference, this workshop aims at providing a place to discuss the reproducibility of the experiments basing the publications submitted to the main conference. L. Nussbaum was PC chair (with O. Richard) while M. Quinson was PC member.

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

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

IUT Charlemagne: Lucas Nussbaum, Systems, 40 ETD, 1ère année (L1), Université de Lorraine, France

IUT Charlemagne: Lucas Nussbaum, Networks, 70 ETD, année spéciale (L1/2), Université de Lorraine, France

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

IUT Charlemagne: Lucas Nussbaum, Administration of applications, 24 ETD, Licence pro ASRALL (L3), Université de Lorraine, France

IUT Charlemagne: Sylvain Contassot-Vivier, “Algorithmique”, 42 ETD, année spéciale (L1), université de Lorraine, France

Licence: Jens Gustedt, “algorithmique et programmation”, 20 TD, niveau (L1), 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

Master: Sylvain Contassot-Vivier, “Systèmes communicants”, 24 ETD, M2, 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: Lucas Nussbaum, Networks and systems, Systems part (40h), 2éme 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

In addition, Martin Quinson authored a pedagogic platform in collaboration with Gérald 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. Its advantages for the learning process are described in [11].

9.2.2. Supervision

PhD: Wilfried Kirschenmann, *Towards sustainable intensive computing kernels*, University of Lorraine, October 17, 2012, Stephane Vialle & Laurent Plagne (EDF)

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: Thomas Jost, *Solveurs linéaires creux sur GPU*, since Jan 2010, Bruno Lèvy & Sylvain Contassot-Vivier

PhD in progress: Soumeya Hernane, *Models and algorithms for consistent data sharing in high performance parallel and distributed computing*, since Oct 2007, Jens Gustedt & Mohamad Benyettou

9.2.3. Juries

In 2012, our team members participated to following thesis and habilitation committees:

Stéphane Genaud, rapporteur, Adrian Muresan, University of Lyon - École Normale Supérieure, France.

Jens Gustedt, member, Jean-Loup Guillaume (HDR), University Paris 6.

Martin Quinson, rapporteur, Silas De Munck, University of Antwerp, Belgium.

Martin Quinson, member, Sabina Akhtar, University de Lorraine, France.

Martin Quinson, member, Bogdan Cornea, University de Franche-Comté, France.

Stephane Vialle, member, Lokman Abbas-Turki, University of Paris-Est, France.

Stephane Vialle, rapporteur, Jonathan Caux, University Blaise Pascal of Clermont-Ferrand, France.

Sylvain Contassot-Vivier, member, Karim Dahman, université de Lorraine, France

Sylvain Contassot-Vivier, member, Tomás Navarrete Gutiérrez, université de Lorraine, France

Sylvain Contassot-Vivier, member, Wahiba Taouali, université de Lorraine, France

9.3. 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.

Martin Quinson acts as a member to the Inria-Nancy Grand Est committee for Scientific Mediation. He works on several activities to demonstrate the *algorithmic thinking* at the core of the Computer Science without requiring any computer or even electric devices. An intern was funded for two months by the national Inria fund toward science popularization on this topic. These activities were demonstrated several times this year by Thomas Jost, Marion Guthmuller, Sébastien Badia and Martin Quinson to during popularization events welcoming the public within our laboratory. They were also shown during the national APMEP days, a nation-wide gathering of maths teachers at the secondary level. Martin Quinson was involved in popularization activities with Interstice [1] by writing short debunking articles (“Idées reçues”) for non computer scientists about the Church thesis and Turing’s work.

[1] http://interstices.info

10. 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, p. 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, p. 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, p. 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, p. 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, p. 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, p. 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

Articles in International Peer-Reviewed Journals

[12] L. 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", 2012, http://dx.doi.org/10.1002/cpe.2862.

[13] E. JEANVOINE, L. SARZYNIEC, L. NUSSBAUM. *Kadeploy3: Efficient and Scalable Operating System Provisioning for Clusters*, in "USENIX ;login:", March 2013, http://hal.inria.fr/hal-00764813.

International Conferences with Proceedings

[14] L. BOBELIN, A. LEGRAND, M. DAVID, P. NAVARRO, M. QUINSON, F. SUTER, C. THIERY. *Scalable Multi-Purpose Network Representation for Large Scale Distributed System Simulation*, in "CCGrid 2012 – The 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing", Ottawa, Canada, May 2012, 19, http://hal.inria.fr/hal-00650233.

[15] T. BUCHERT, L. NUSSBAUM. *Using business workflows to improve quality of experiments in distributed systems research*, in "SC12 - SuperComputing 2012 (poster session)", Salt Lake City, United States, August 2012, http://hal.inria.fr/hal-00724312.

[16] S. CONTASSOT-VIVIER, D. ELIZONDO. *A near linear algorithm for testing linear separability in two dimensions*, in "EANN 2012 - 13th International Conference on Engineering Applications of Neural Networks", London, United Kingdom, September 2012, http://hal.inria.fr/hal-00726624.

[17] S. CONTASSOT-VIVIER, T. JOST, S. VIALLE. *Impact of Asynchronism on GPU Accelerated Parallel Iterative Computations*, in "PARA 2010 - 10th International Conference on Applied Parallel and Scientific Computing", Reykjavík, Iceland, K. JÓNASSON (editor), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, vol. 7133, p. 43-53 [*DOI :* 10.1007/978-3-642-28151-8_5], http://hal.inria.fr/hal-00685153.

[18] J. GUSTEDT, E. JEANVOINE. *Relaxed Synchronization with Ordered Read-Write Locks*, in "Euro-Par 2011: Parallel Processing Workshops", Bordeaux, France, M. ALEXANDER, P. D'AMBRA, A. BELLOUM, G. BOSILCA, M. CANNATARO, M. DANELUTTO, B. D. MARTINO, M. GERNDT, E. JEANNOT, R. NAMYST, J. ROMAN, S. L. SCOTT (editors), LNCS, Springer, May 2012, vol. 7155, p. 387-397, This article is accepted for publication in the post-proceedings of the Workshop on Algorithms and Programming Tools for Next-Generation High-Performance Scientific Software (HPSS) 2011, held in the context of Euro-Par 2011, August 29, 2011, Bordeaux, France., http://hal.inria.fr/hal-00639289.

[19] J. GUSTEDT, S. VIALLE, H. FREZZA-BUET, D. BOUMBA SITOU, N. FRESSENGEAS, J. FIX. *InterCell: a Software Suite for Rapid Prototyping and Parallel Execution of Fine Grained Applications*, in "PARA 2010 - 10th International Conference on Applied Parallel and Scientific Computing", Reykjavík, Iceland, K. JÓNASSON (editor), LNCS, Springer, January 2012, vol. 7133, p. 282-292 [*DOI :* 10.1007/978-3-642-28151-8], http://hal.inria.fr/hal-00645121.

[20] 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.

[21] S. HERNANE, J. GUSTEDT, M. BENYETTOU. *A Dynamic Distributed Algorithm for Read Write Locks (extended abstract)*, in "PDP 2012 - 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing", München, Germany, R. STOTZKA, M. SCHIFFERS, Y. COTRONIS (editors), IEEE, February 2012, p. 180-184 [*DOI :* 10.1109/PDP.2012.32], http://hal.inria.fr/hal-00641068.

[22] W. KIRSCHENMANN, L. PLAGNE, S. VIALLE. *Multi-Target Vectorization with MTPS C++ Generic Library*, in "PARA 2010 - 10th International Conference on Applied Parallel and Scientific Computing", Reykjavík, Iceland, K. JÓNASSON (editor), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, vol. 7134, p. 336-346 [*DOI :* 10.1007/978-3-642-28145-7_33], http://hal.inria.fr/hal-00685159.

[23] C. MAKASSIKIS, S. VIALLE, X. WARIN. *FT-GReLoSSS: a Skeletal-Based Approach towards Application Parallelization and Low-Overhead Fault Tolerance*, in "20th Euromicro International Conference on Parallel, Distributed and Network-Based Computing - PDP 2012", Garching, Germany, February 2012, 8 pages, http://hal.inria.fr/hal-00681664.

[24] E. MICHON, J. GOSSA, S. GENAUD. *Free elasticity and free CPU power for scientific workloads on IaaS Clouds*, in "18th IEEE International Conference on Parallel and Distributed Systems", IEEE, 2012.

[25] M. QUINSON, C. ROSA, C. THIERY. *Parallel Simulation of Peer-to-Peer Systems*, in "CCGrid 2012 – The 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing", Ottawa, Canada, May 2012, http://hal.inria.fr/inria-00602216.

[26] 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, 2013, http://hal.inria.fr/hal-00724308.

National Conferences with Proceeding

[27] T. BUCHERT. *Orchestration d'expériences à l'aide de processus métier*, in "ComPAS : Conférence d'informatique en Parallélisme, Architecture et Système.", Grenoble, France, October 2012, http://hal.inria.fr/hal-00749601.

[28] T. BUCHERT, L. NUSSBAUM. *Leveraging business workflows in distributed systems research for the orchestration of reproducible and scalable experiments*, in "MajecSTIC 2012", Lille, France, August 2012, http://hal.inria.fr/hal-00724313.

Conferences without Proceedings

[29] L. SARZYNIEC, S. BADIA, E. JEANVOINE, L. NUSSBAUM. *Scalability Testing of the Kadeploy Cluster Deployment System using Virtual Machines on Grid'5000*, in "SCALE Challenge 2012, held in conjunction with CCGrid'2012", Ottawa, Canada, May 2012, http://hal.inria.fr/hal-00700962.

Scientific Books (or Scientific Book chapters)

[30] M. Sauget, S. Contassot-Vivier, M. Salomon. *Parallelization of neural network building and training: an original decomposition method*, in "Horizons in Computer Science Research", A. R. Baswell (editor), Advances in Mathematics Research, Nova Publishers, 2012, vol. 17, http://hal.inria.fr/hal-00643920.

[31] 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, February 2013, to appear.

Research Reports

[32] D. Balouek, A. Carpen Amarie, G. Charrier, F. Desprez, E. Jeannot, E. Jeanvoine, A. Lèbre, D. Margery, N. Niclausse, L. Nussbaum, O. Richard, C. Pérez, F. Quesnel, C. Rohr, L. Sarzyniec. *Adding Virtualization Capabilities to Grid'5000*, Inria, July 2012, n^{o} RR-8026, 18, http://hal.inria.fr/hal-00720910.

[33] F. Desprez, G. Fox, E. Jeannot, K. Keahey, M. Kozuch, D. Margery, P. Neyron, L. Nussbaum, C. Pérez, O. Richard, W. Smith, G. Von Laszewski, J. Vöckler. *Supporting Experimental Computer Science*, Inria, July 2012, n^{o} RR-8035, 29, http://hal.inria.fr/hal-00722605.

[34] F. Desprez, G. Fox, E. Jeannot, K. Keahey, M. Kozuch, D. Margery, P. Neyron, L. Nussbaum, C. Pérez, O. Richard, W. Smith, G. Von Laszewski, J. Vöckler. *Supporting Experimental Computer Science*, March 2012, n^{o} Argonne National Laboratory Technical Memo 326, http://hal.inria.fr/hal-00720815.

[35] E. Jeanvoine, L. Sarzyniec, L. Nussbaum. *Kadeploy3: Efficient and Scalable Operating System Provisioning for HPC Clusters*, Inria, June 2012, n^{o} RR-8002, http://hal.inria.fr/hal-00710638.

Other Publications

[36] S. Vialle. *Energy issues of GPU computing clusters*, in "Colloquium EJC-ICT-2012 on Towards ecological and energy efficient Information and Communication Technology", Lyon, France, November 19-20, 2012, Invited Speaker.

References in notes

[37] 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, Paper 31, An extended version is to appear in LNCS, http://hal.archives-ouvertes.fr/hal-00547300/en/.