IN PARTNERSHIP WITH:
CNRS

Institut polytechnique de Grenoble

Université Pierre Mendes-France (Grenoble)

Université Joseph Fourier (Grenoble)

Activity Report 2012

Project-Team MOAIS

PrograMming and scheduling design fOr Applications in Interactive Simulation

IN COLLABORATION WITH: Laboratoire d'Informatique de Grenoble (LIG)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Distributed and High Performance Computing

Table of contents

Project-Team MOAIS

Keywords: Scheduling, Interactive Computing, Parallel And Distributed Algorithms, High Performance Computing, Fault Tolerance, Parallel Programming Model

Creation of the Project-Team: January 01, 2006 .

1. Members

Research Scientists
Thierry Gautier [Junior Researcher CR1]
Bruno Raffin [Junior Researcher CR1, HdR]

Faculty Members
Jean-Louis Roch [Team leader, Associate Professor]
François Broquedis [Associate Professor]
Vincent Danjean [Associate Professor]
Pierre-François Dutot [Associate Professor]
Guillaume Huard [Associate Professor]
Grégory Mounié [Associate Professor]
Denis Trystram [Professor, HdR]
Frédéric Wagner [Associate Professor]
Clément Pernet [Associate Professor]

Engineers
Christian Séguy [CNRS/LIG Engineer, 40%]
Pierre Neyron [CNRS/LIG, Research engineer, 40%]
Eric Amat [2010-2012. Inria Grant (ADT VGATE)]
Philippe Virouleau [2012, Engineer ADT KAHWA]
Martin Xavier [Apprenti, VISIONAIR project]

PhD Students
Mohamed-Slim Bouguerra [2008, Inria Cordi]
Matthieu Dreher [2011-2015.]
Stefano Drimon Kurz Mor [2011-2015, co-tutelle with UFRGS.]
Marie Durand [2010-2013. Funded by ANR project REPDYN]
Adel Essafi [2006, co-tutelle ESST Tunis, Tunisia (Amine Mahjoub)]
Mathias Ettinger [2011-2015. Funded by Inria contract EDF]
Joao Ferreira Lima [2010, co-tutelle Grenoble Univ – UFRGS Brazil, CAPES COFECUB]
Ludovic Jacquin [2009, common to PLANETE and MOAIS]
Christophe Laferrière [2009, Nano2012-HiPeComp contract]
Florence Monna [2011, co-advised Paris-6]
Swann Perarnau [2008, MRNT scholarship]
Vinicius Pinheiro [2011, co-advised with USP]
Ziad Sultan [2012-2015, ANR HPAC scholarship]

Post-Doctoral Fellows
Alexandre Ancel [Petaflow project]
Joachim Lepping [1 year]

Visiting Scientists
Joseph Peters [Simon Fraser University, Canada, 2 months]
Wieslaw Kubiak [Memorial University, Canada, 1 month]
Jacek Blazewicz [Poznan University of Technology, 2 months]

Afredo Goldman [USP Sao Paulo, Brasil, 1 month]

Administrative Assistants

Annie Simon [Inria Administrative Assistant, 40% (from oct.)]
Annie-Claude Vial-Dallais [CNRS Administrative Assistant, 10%]
Christine Guimet [CNRS Administrative Assistant, 20%]

2. Overall Objectives

2.1. Introduction

The goal of the MOAIS team-project is to develop the scientific and technological foundations for parallel programming that enable to achieve provable performances on distributed parallel architectures, from multi-processor systems on chips to computational grids and global computing platforms. Beyond the optimization of the application itself, the effective use of a larger number of resources is expected to enhance the performance. This encompasses large scale scientific interactive simulations (such as immersive virtual reality) that involve various resources: input (sensors, cameras, ...), computing units (processors, memory), output (videoprojectors, images wall) that play a prominent role in the development of high performance parallel computing.

To reach this goal, MOAIS gathers experts in : algorithm design, scheduling, parallel programming (both low level and high level API), interactive applications. The research directions of the MOAIS team are focused on scheduling problems with a multi-criteria performance objective: precision, reactivity, resources consumption, reliability, ... The originality of the MOAIS approach is to use the application's adaptability to control its scheduling:

- the application describes synchronization conditions;
- the scheduler computes a schedule that verifies those conditions on the available resources;
- each resource behaves independently and performs the decision of the scheduler.

To enable the scheduler to drive the execution, the application is modeled by a macro data flow graph, a popular bridging model for parallel programming (BSP, Nesl, Earth, Jade, Cilk, Athapascan, Smarts, Satin, ...) and scheduling. A node represents the state transition of a given component; edges represent synchronizations between components. However, the application is malleable and this macro data flow is dynamic and recursive: depending on the available resources and/or the required precision, it may be unrolled to increase precision (e.g. zooming on parts of simulation) or enrolled to increase reactivity (e.g. respecting latency constraints). The decision of unrolling/enrolling is taken by the scheduler; the execution of this decision is performed by the application.

The MOAIS project-team is structured around four axis:

- **Scheduling**: To formalize and study the related scheduling problems, the critical points are: the modeling of an adaptive application; the formalization and the optimization of the multi-objective problems; the design of scalable scheduling algorithms. We are interested in classical combinatorial optimization methods (approximation algorithms, theoretical bounds and complexity analysis), and also in non-standard methods such as Game Theory.
- **Adaptive parallel and distributed algorithms**: To design and analyze algorithms that may adapt their execution under the control of the scheduling, the critical point is that algorithms are either parallel or distributed; then, adaptation should be performed locally while ensuring the coherency of results.
- **Programming interfaces and tools for coordination and execution**: To specify and implement interfaces that express coupling of components with various synchronization constraints, the critical point is to enable an efficient control of the coupling while ensuring coherency. We develop the **Kaapi** runtime software that manages the scheduling of multithreaded computations with billions

of threads on a virtual architecture with an arbitrary number of resources; Kaapi supports node additions and resilience. Kaapi manages the *fine grain* scheduling of the computation part of the application. To enable parallel application execution and analysis. We develop runtime tools that support large scale and fault tolerant processes deployment (**TakTuk**), visualization of parallel executions on heterogeneous platforms (**Triva**), reproducible CPU load generation on many-cores machines (**KRASH**).

- **Interactivity**: To improve interactivity, the critical point is scalability. The number of resources (including input and output devices) should be adapted without modification of the application. We develop the **FlowVR** middleware that enables to configure an application on a cluster with a fixed set of input and output resources. FlowVR manages the *coarse grain* scheduling of the whole application and the latency to produce outputs from the inputs.

Often, computing platforms have a dynamic behavior. The dataflow model of computation directly enables to take into account addition of resources. To deal with resilience, we develop softwares that provide **fault-tolerance** to dataflow computations. We distinguish non-malicious faults from malicious intrusions. Our approach is based on a checkpoint of the dataflow with bounded and amortized overhead.

2.2. Highlights of the Year

- Moais participates to the Kinovis project (leaded by E. Boyer, Morpheo team): Kinovis is the successor of the Grimage platform and has been selected in the equipex call for proposal.

3. Scientific Foundations

3.1. Scheduling

Participants: Pierre-François Dutot, Guillaume Huard, Grégory Mounié, Jean-Louis Roch, Denis Trystram, Frédéric Wagner.

The goal of this theme is to determine adequate multi-criteria objectives which are efficient (precision, reactivity, speed) and to study scheduling algorithms to reach these objectives.

In the context of parallel and distributed processing, the term *scheduling* is used with many acceptations. In general, scheduling means assigning tasks of a program (or processes) to the various components of a system (processors, communication links).

Researchers within MOAIS have been working on this subject for many years. They are known for their multiple contributions for determining the target dates and processors the tasks of a parallel program should be executed; especially regarding execution models (taking into account inter-task communications or any other system features) and the design of efficient algorithms (for which there exists a performance guarantee relative to the optimal scheduling).

Parallel tasks model and extensions. We have contributed to the definition and promotion of modern task models: parallel moldable tasks and divisible load. For both models, we have developed new techniques to derive efficient scheduling algorithms (with a good performance guaranty). We proposed recently some extensions taking into account machine unavailabilities (reservations).

Multi-objective Optimization. A natural question while designing practical scheduling algorithms is "which criterion should be optimized ?". Most existing works have been developed for minimizing the *makespan* (time of the latest tasks to be executed). This objective corresponds to a system administrator view who wants to be able to complete all the waiting jobs as soon as possible. The user, from his-her point of view, would be more interested in minimizing the average of the completion times (called *minsum*) of the whole set of submitted jobs. There exist several other objectives which may be pertinent for specific use. We worked on the problem of designing scheduling algorithms that optimize simultaneously several objectives with a theoretical guarantee on each objective. The main issue is that most of the policies are good for one criterion but bad for another one.

We have proposed an algorithm that is guaranteed for both *makespan* and *minsum*. This algorithm has been implemented for managing the resources of a cluster of the regional grid CIMENT. More recently, we extended such analysis to other objectives (makespan and reliability). We concentrate now on finding good algorithms able to schedule a set of jobs with a large variety of objectives simultaneously. For hard problems, we propose approximation of Pareto curves (best compromizes).

Incertainties. Most of the new execution supports are characterized by a higher complexity in predicting the parameters (high versatility in desktop grids, machine crash, communication congestion, cache effects, etc.). We studied some time ago the impact of incertainties on the scheduling algorithms. There are several ways for dealing with this problem: First, it is possible to design robust algorithms that can optimized a problem over a set of scenarii, another solution is to design flexible algorithms. Finally, we promote semi on-line approaches that start from an optimized off-line solution computed on an initial data set and updated during the execution on the "perturbed" data (stability analysis).

Game Theory. Game Theory is a framework that can be used for obtaining good solution of both previous problems (multi-objective optimization and incertain data). On the first hand, it can be used as a complement of multi-objective analysis. On the other hand, it can take into account the incertainties. We are curently working at formalizing the concept of cooperation.

Scheduling for optimizing parallel time and memory space. It is well known that parallel time and memory space are two antagonists criteria. However, for many scientific computations, the use of parallel architectures is motivated by increasing both the computation power and the memory space. Also, scheduling for optimizing both parallel time and memory space targets an important multicriteria objective. Based on the analysis of the dataflow related to the execution, we have proposed a scheduling algorithm with provable performance.

Coarse-grain scheduling of fine grain multithreaded computations on heterogeneous platforms. Designing multi-objective scheduling algorithms is a transversal problem. Work-stealing scheduling is well studied for fine grain multithreaded computations with a small critical time: the speed-up is asymptotically optimal. However, since the number of tasks to manage is huge, the control of the scheduling is expensive. We proposed a generalized lock-free cactus stack execution mechanism, to extend previous results, mainly from Cilk, based on the *work-first principle* for strict multi-threaded computations on SMPs to general multithreaded computations with dataflow dependencies. The main result is that optimizing the sequential local executions of tasks enables to amortize the overhead of scheduling. This distributed work-stealing scheduling algorithm has been implemented in **Kaapi**.

3.2. Adaptive Parallel and Distributed Algorithms Design

Participants: François Broquedis, Pierre-François Dutot, Thierry Gautier, Guillaume Huard, Bruno Raffin, Jean-Louis Roch, Denis Trystram, Frédéric Wagner.

This theme deals with the analysis and the design of algorithmic schemes that control (statically or dynamically) the grain of interactive applications.

The classical approach consists in setting in advance the number of processors for an application, the execution being limited to the use of these processors. This approach is restricted to a constant number of identical resources and for regular computations. To deal with irregularity (data and/or computations on the one hand; heterogeneous and/or dynamical resources on the other hand), an alternate approach consists in adapting the potential parallelism degree to the one suited to the resources. Two cases are distinguished:

- in the classical bottom-up approach, the application provides fine grain tasks; then those tasks are clustered to obtain a minimal parallel degree.
- the top-down approach (Cilk, Cilk+, TBB, Hood, Athapascan) is based on a work-stealing scheduling driven by idle resources. A local sequential depth-first execution of tasks is favored when recursive parallelism is available.

Ideally, a good parallel execution can be viewed as a flow of computations flowing through resources with no control overhead. To minimize control overhead, the application has to be adapted: a parallel algorithm on p resources is not efficient on $q < p$ resources. On one processor, the scheduler should execute a sequential algorithm instead of emulating a parallel one. Then, the scheduler should adapt to resource availability by changing its underlying algorithm. This first way of adapting granularity is implemented by Kaapi (default work-stealing schedule based on work-first principle).

However, this adaptation is restrictive. More generally, the algorithm should adapt itself at runtime to improve its performance by decreasing the overheads induced by parallelism, namely the arithmetic operations and communications. This motivates the development of new parallel algorithmic schemes that enable the scheduler to control the distribution between computation and communication (grain) in the application to find the good balance between parallelism and synchronizations. MOAIS has exhibited several techniques to manage adaptivity from an algorithmic point of view:

- amortization of the number of global synchronizations required in an iteration (for the evaluation of a stopping criterion);
- adaptive deployment of an application based on on-line discovery and performance measurements of communication links;
- generic recursive cascading of two kind of algorithms: a sequential one, to provide efficient executions on the local resource, and a parallel one that enables an idle resource to extract parallelism to dynamically suit the degree of parallelism to the available resources.

The generic underlying approach consists in finding a good mix of various algorithms, what is often called a "poly-algorithm". Particular instances of this approach are Atlas library (performance benchmark are used to decide at compile time the best block size and instruction interleaving for sequential matrix product) and FFTW library (at run time, the best recursive splitting of the FFT butterfly scheme is precomputed by dynamic programming). Both cases rely on pre-benchmarking of the algorithms. Our approach is more general in the sense that it also enables to tune the granularity at any time during execution. The objective is to develop processor oblivious algorithms: similarly to cache oblivious algorithms, we define a parallel algorithm as *processor-oblivious* if no program variable that depends on architecture parameters, such as the number or processors or their respective speeds, needs to be tuned to minimize the algorithm runtime.

We have applied this technique to develop processor oblivious algorithms for several applications with provable performance: iterated and prefix sum (partial sums) computations, stream computations (cipher and hd-video transformation), 3D image reconstruction (based on the concurrent usage of multi-core and GPU), loop computations with early termination. Finally, to validate these novel parallel computation schemes, we developed a tool named **KRASH**. This tool is able to generate dynamic CPU load in a reproducible way on many-cores machines. Thus, by providing the same experimental conditions to several parallel applications, it enables users to evaluate the efficiency of resource uses for each approach.

By optimizing the work-stealing to our adaptive algorithm scheme, the non-blocking (wait-free) implementation of Kaapi has been designed and leads to the C library X-kaapi.

Extensions concern the development of algorithms that are both cache and processor oblivious on heterogeneous processors. The processor algorithms proposed for prefix sums and segmentation of an array are cache oblivious too.

3.3. Interactivity

Participants: Vincent Danjean, Pierre-François Dutot, Thierry Gautier, Bruno Raffin, Jean-Louis Roch.

The goal of this theme is to develop approaches to tackle interactivity in the context of large scale distributed applications.

We distinguish two types of interactions. A user can interact with an application having only little insight about the internal details of the program running. This is typically the case for a virtual reality application where the user just manipulates 3D objects. We have a "user-in-the-loop". In opposite, we have an "expert -in-the-loop" if the user is an expert that knows the limits of the progam that is being executed and that he can interacts with it to steer the execution. This is the case for instance when the user can change some parameters during the execution to improve the convergence of a computation.

3.3.1. User-in-the-loop

Some applications, like virtual reality applications, must comply with interactivity constraints. The user should be able to observe and interact with the application with an acceptable reaction delay. To reach this goal the user is often ready to accept a lower level of details. To execute such application on a distributed architecture requires to balance the workload and activation frequency of the different tasks. The goal is to optimize CPU and network resource use to get as close as possible to the reactivity/level of detail the user expect.

Virtual reality environments significantly improve the quality of the interaction by providing advanced interfaces. The display surface provided by multiple projectors in CAVE -like systems for instance, allows a high resolution rendering on a large surface. Stereoscopic visualization gives an information of depth. Sound and haptic systems (force feedback) can provide extra information in addition to visualized data. However driving such an environment requires an important computation power and raises difficult issues of synchronization to maintain the overall application coherent while guaranteeing a good latency, bandwidth (or refresh rate) and level of details. We define the coherency as the fact that the information provided to the different user senses at a given moment are related to the same simulated time.

Today's availability of high performance commodity components including networks, CPUs as well as graphics or sound cards make it possible to build large clusters or grid environments providing the necessary resources to enlarge the class of applications that can aspire to an interactive execution. However the approaches usually used for mid size parallel machines are not adapted. Typically, there exist two different approaches to handle data exchange between the processes (or threads). The synchronous (or FIFO) approach ensures all messages sent are received in the order they were sent. In this case, a process cannot compute a new state if all incoming buffers do not store at least one message each. As a consequence, the application refresh rate is driven by the slowest process. This can be improved if the user knows the relative speed of each module and specify a read frequency on each of the incoming buffers. This approach ensures a strong coherency but impact on latency. This is the approach commonly used to ensure the global coherency of the images displayed in multi-projector environments.The other approach, the asynchronous one, comes from sampling systems. The producer updates data in a shared buffer asynchronously read by the consumer. Some updates may be lost if the consumer is slower than the producer. The process refresh rates are therefore totally independent. Latency is improved as produced data are consumed as soon as possible, but no coherency is ensured. This approach is commonly used when coupling haptic and visualization systems. A fine tuning of the application usually leads to satisfactory results where the user does not experience major incoherences. However, in both cases, increasing the number of computing nodes quickly makes infeasible hand tuning to keep coherency and good performance.

We propose to develop techniques to manage a distributed interactive application regarding the following criteria :

- latency (the application reactivity);
- refresh rate (the application continuity);
- coherency (between the different components);
- level of detail (the precision of computations).

We developed a programming environment, called FlowVR, that enables the expression and realization of loosen but controlled coherency policies between data flows. The goal is to give users the possibility to express a large variety of coherency policies from a strong coherency based on a synchronous approach to an uncontrolled coherency based on an asynchronous approach. It enables the user to loosen coherency where it is acceptable, to improve asynchronism and thus performance. This approach maximizes the refresh rate and minimizes the latency given the coherency policy and a fixed level of details. It still requires the user to tune many parameters. In a second step, we are planning to explore auto-adaptive techniques that enable to decrease the number of parameters that must be user tuned. The goal is to take into account (possibly dynamically) user specified high level parameters like target latencies, bandwidths and levels of details, and to have the system automatically adapt to reach a trade-off given the user wishes and the resources available. Issues include multi-criterion optimizations, adaptive algorithmic schemes, distributed decision making, global stability and balance of the regulation effort.

3.3.2. Expert-in-the-loop

Some applications can be interactively guided by an expert who may give advices or answer specific questions to hasten a problem resolution. A theoretical framework has been developed in the last decade to define precisely the complexity of a problem when interactions with an expert is allowed. We are studying these interactive proof systems and interactive complexity classes in order to define efficient interactive algorithms dedicated to scheduling problems. This, in particular, applies to load-balancing of interactive simulations when a user interaction can generate a sudden surge of imbalance which could be easily predicted by an operator.

3.4. Adaptive middleware for code coupling and data movements

Participants: François Broquedis, Vincent Danjean, Thierry Gautier, Clément Pernet, Bruno Raffin, Jean-Louis Roch, Frédéric Wagner.

This theme deals with the design and implementation of programming interfaces in order to achieve an efficient coupling of distributed components.

The implementation of interactive simulation application requires to assemble together various software components and to ensure a semantic on the displayed result. To take into account functional aspects of the computation (inputs, outputs) as well as non functional aspects (bandwidth, latency, persistence), elementary actions (method invocation, communication) have to be coordinated in order to meet some performance objective (precision, quality, fluidity, *etc*). In such a context the scheduling algorithm plays an important role to adapt the computational power of a cluster architecture to the dynamic behavior due to the interactivity. Whatever the scheduling algorithm is, it is fundamental to enable the control of the simulation. The purpose of this research theme is to specify the semantics of the operators that perform components assembling and to develop a prototype to experiment our proposals on real architectures and applications.

3.4.1. Application Programming Interface

The specification of an API to compose interactive simulation application requires to characterize the components and the interaction between components.The respect of causality between elementary events ensures, at the application level, that a reader will see the *last* write with respect to an order. Such a consistency should be defined at the level of the application to control the events ordered by a chain of causality. For instance, one of the result of Athapascan was to prove that a data flow consistency is more efficient than other ones because it generates fewer messages. Beyond causality based interactions, new models of interaction should be studied to capture non predictable events (delay of communication, capture of image) while ensuring a semantic.

Our methodology is based on the characterization of interactions required between components in the context of an interactive simulation application. For instance, criteria could be coherency of visualization, degree of interactivity. Beyond such characterization we hope to provide an operational semantic of interactions (at least well suited and understood by usage) and a cost model. Moreover they should be preserved by composition to predict the cost of an execution for part of the application.

The main result relies on a computable representation of the future of an execution; representations such as macro data flow are well suited because they explicit which data are required by a task. Such a representation can be built at runtime by an interpretation technique: the execution of a function call is differed by computing beforehand at runtime a graph of tasks that represents the (future) calls to execute.

3.4.2. Kernel for Asynchronous, Adaptive, Parallel and Interactive Application

Managing the complexity related to fine grain components and reaching high efficiency on a cluster architecture require to consider a dynamic behavior. Also, the runtime kernel is based on a representation of the execution: data flow graph with attributes for each node and efficient operators will be the basis for our software. This kernel has to be specialized for the considered applications. The low layer of the kernel has features to transfer data and to perform remote signalization efficiently. Well known techniques and legacy code have to be reused. For instance, multithreading, asynchronous invocation, overlapping of latency by computing, parallel communication and parallel algorithms for collective operations are fundamental techniques to reach performance. Because the choice of the scheduling algorithm depends on the application and the architecture, the kernel will provide an *causally connected representation* of the system that is running. This allows to specialize the computation of a good schedule of the data flow graph by providing algorithms (scheduling algorithms for instance) that compute on this (causally connected) representation: any modification of the representation is turned into a modification on the system (the parallel program under execution). Moreover, the kernel provides a set of basic operators to manipulate the graph (*e.g.* computes a partition from a schedule, remapping tasks, ...) to allow to control a distributed execution.

4. Application Domains

4.1. Outline

The scientific methodology of MOAIS consists in:

- designing algorithms with provable performance on generic theoretical models. In particular we develop randomized algorithms for distributed scheduling and approximate multi-objective optimization theory..
- implementing and evaluating those algorithms with our main softwares:
  - Kaapi for fine grain scheduling of compute-intensive applications;
  - FlowVR for coarse-grain scheduling of interactive applications;
  - TakTuk, a tool for large scale remote executions deployment.
  - Triva, for the visualization of heterogeneous parallel executions.
  - KRASH, to generate reproducible CPU load on many-cores machines.
- customizing our softwares for their use in real applications studied and developed by other partners. Applications are essential to the validation and further development of MOAIS results. Application fields are: virtual reality and scientific computing (simulation, visualization, combinatorial optimization, biology, computer algebra). Depending on the application the target architecture ranges from MPSoCs (multi-processor system on chips), multicore and GPU units to clusters and heterogeneous grids. In all cases, the performance is related to the efficient use of the available, often heterogeneous, parallel resources.

MOAIS research is not only oriented towards theory but also focuses on applicative software and hardware platforms developed with external partners. Significant efforts are made to build, manage and maintain these platforms. We are involved with other teams in four main platforms:

- SOFA, a real-time physics simulation engine (http://www.sofa-framework.org/;
- Grimage, a 3D modeling and high performance 3D rendering platform (http://www.inrialpes.fr/grimage) and its evolution with the new Kinovis platform.
- Digitalis, a 780 core cluster based on Intel Nehalem processors and Infiniband network. Digitalis is used both for batch computations and interactive applications;
- Grid'5000, the exprimental national grid (http://www.grid5000.fr/).

4.2. Virtual Reality

Participants: Thierry Gautier, Bruno Raffin, Jean-Louis Roch.

We are pursuing and extending existing collaborations to develop virtual reality applications on PC clusters and grid environments:

- Real time 3D modeling. An on-going collaboration with the MORPHEO project focuses on developing solutions to enable real time 3D modeling from multiple cameras using a PC cluster. This work is tightly coupled to the FlowVR software. Our recent developments take two main directions:
  - to provide the user a high level of interaction and immersion in the mixed reality environment. This work is focused on the Grimage platform and its successor, the new equipex Kinovis managed by Morpheo team. The camera position and orientation need to be precisely known at anytime, requiring to develop on-line calibration approaches. The background subtraction cannot anymore be based on a static background learning for the mobile camera, required here too new algorithms.
  - Distributed collaboration across distant sites. In the context of the ANR DALIAn we have developped a collaborative application where multiple users, distributed in several sites each using a real time 3D modeling platform, can meet in a virtual world with a user in Grenoble also using a similar platform. The main issues are related to data transfers that need to be carefully managed to ensure a good latency while keeping a good quality, and the development of new interaction paradigms. Focusing on distributed scientific simulation, we extend those technologies in the context of the FVNANO and PetaFlow contracts.
- Real time physical simulation. We are collaborating with the Imagine project on the SOFA simulation framework. Marie Durand a Ph.D. co-advised by François Faure (IMAGINE) and Bruno Raffin, works on parallelizing SOFA using the KAAPI programming environment. The challenge is to provide SOFA with a parallelization that is efficient (real-time) while not being invasive for SOFA programmers (usually not parallel programmer). We developed a first version using the Kaapi environment for SMP machines that relies on a mix of work-stealing and dependency graph analysis and partitioning. A second version targets machines with multiples CPUs and multiple GPUs. We extended the initial framework to support a work stealing based load balancing between CPUs and GPUs. It required to extend Kaapi to support heterogeneous tasks (GPU and CPU ones) and to adapt the work stealing strategy to limit data transfers between CPUs and GPUs (the main bottleneck for GPU computing).
- Distant collaborative work. We conduct experiments using FlowVR for running applications on Grid environments. Two kinds of experiments will be considered: collaborative work by coupling two or more distant VR environments ; large scale interactive simulation using computing resources from the grid. For these experiments, we are collaborating with the LIFO and the LABRI.
- Parallel cache-oblivious algorithms for scientific visualization. In collaboration with the CEA DAM, we have developed a cache-oblivious algorithm with provable performance for irregulars meshes. Based on this work, we are studying parallel algorithms that take advantage of the shared cache usually encountered on multi-core architectures (L3 shared cache) and of harware accelerators. In collaboration with EDF, we develop new parallel algorithms for scientific vizualization (eg VTK) on multicore (phD thesis of Mathias Ettinger). We are also considering adaptive algorithms to take advantage of the new trend of computers to integrate several computing units that may have different computing abilities (multicore arithmetic and graphical processing units, eventually integrated on one chip). We study balancing workload on multi GPU and CPU architectures for scientific visualization problems.

4.3. Code Coupling and Grid Programming

Participants: François Broquedis, Thierry Gautier, Jean-Louis Roch, Vincent Danjean, Frédéric Wagner.

Code coupling aim is to assemble component to build distributed applications by reusing legacy code. The objective here is to build high performance applications for cluster and grid infrastructures.

- **Grid programming model and runtime support.** Programming the grid is a challenging problem. The MOAIS Team has a strong knowledge in parallel algorithms and develop a runtime support for scheduling grid program written in a very high level interface. The parallelism from recursive divide and conquer applications and those from iterative simulation are studied. Scheduling heuristics are based on online work stealing for the former class of applications, and on hierarchical partitioning for the latter. The runtime support provides capabilities to hide latency by computation thanks to a non-blocking one-side communication protocol and by re-ordering computational tasks.
- **Grid application deployment.** To test grid applications, we need to deploy and start programs on all used computers. This can become difficult if the real topology involves several clusters with firewall, different runtime environments, etc. The MOAIS Team designed and implemented a new tool called `karun` that allows a user to easily deploy a parallel application wrote with the KAAPI software. This KAAPI tool relies on the `TakTuk` software to quickly launch programs on all nodes. The user only needs to describe the hierarchical networks/clusters involved in the experiment with their firewall if any.
- **Visualization of grid applications execution.** The analysis of applications execution on the grid is challenging both because of the large scale of the platform and because of the heterogeneous topology of the interconnections. To help users to understand their application behavior and to detect potential bottleneck or load unbalance, the MOAIS team designed and implemented a tool named **Triva**. This tool proposes a new three dimensional visualization model that combines topological information to space time data collected during the execution. It also proposes an aggregation mechanism that eases the detection of application load unbalance.

4.4. Safe Distributed Computations

Participants: Vincent Danjean, Thierry Gautier, Clément Pernet, Jean-Louis Roch.

Large scale distributed platforms, such as the GRID and Peer-to-Peer computing systems, gather thousands of nodes for computing parallel applications. At this scale, component failures, disconnections (fail-stop faults) or results modifications (malicious faults) are part of operation, and applications have to deal directly with repeated failures during program runs. Indeed, since failure rate in such platform is proportional to the number of involved resources, the mean time between failure is dramatically decreased on very large size architectures. Moreover, even if a middleware is used to secure the communications and to manage the resources, the computational nodes operate in an unbounded environment and are subject to a wide range of attacks able to break confidentiality or to alter the resources or the computed results. Beyond fault-tolerancy, yet the possibility of massive attacks resulting in an error rate larger than tolerable by the application has to be considered. Such massive attacks are especially of concern due to Distributed Denial of Service, virus or Trojan attacks, and more generally orchestrated attacks against widespread vulnerabilities of a specific operating system that may result in the corruption of a large number of resources. The challenge is then to provide confidence to the parties about the use of such an unbound infrastructure. The MOAIS team addresses two issues:

- fault tolerance (node failures and disconnections): based on a global distributed consistent state , for the sake of scalability;
- security aspects: confidentiality, authentication and integrity of the computations.

Our approach to solve those problems is based on the efficient checkpointing of the dataflow that described the computation at coarse-grain. This distributed checkpoint, based on the local stack of each work-stealer process, provides a causally linked representation of the state. It is used for a scalable checkpoint/restart protocol and for probabilistic detection of massive attacks.

Moreover, we study the scalability of security protocols on large scale infrastructures. Within the SHIVA contract (global competitiveness cluster Minalogic in Grenoble) and in collaboration with C-S company, the Ph.D. of Ludovic Jacquin (coadvised with the PLANETE EPI) we developed a high-rate systematic ciphering platform based on the coupling of a multicore architecture with security components (FPGA and smart card) developed by industrial partners.

4.5. Embedded Systems

Participants: Jean-Louis Roch, Guillaume Huard, Denis Trystram, Vincent Danjean.

To improve the performance of current embedded systems, Multiprocessor System-on-Chip (MPSoC) offers many advantages, especially in terms of flexibility and low cost. Multimedia applications, such as video encoding, require more and more intensive computations. The system should be able to exploit the resources as much as possible to save power and time. This challenge may be addressed by parallel computing coupled with performant scheduling. On-going work focuses on developing the scheduling and monitoring technologies developed in MOAIS for embedded systems.

In the framework of our cooperation with STM (Miguel Santana) and within the SocTrace project, we are developping tools to manage distributed large scale traces. We especially focus on visualization, developping visual aggregation techniques (Phd Damien Dosimont, started in 2/2012 advised by Guillaume Huard in collaboration with Jean-Marc Vincent).

5. Software

5.1. KAAPI

Participants: Thierry Gautier [correspondant], Vincent Danjean, François Broquedis, Pierre Neyron.

Kaapi (http://kaapi.gforge.inria.fr, coordinator T. Gautier) Kaapi is a middleware for high performance applications running on multi-cores/multi- processors as well as cluster or computational grid. Kaapi provides 1/ a very high level API based on macro data flow language; 2/ several scheduling algorithms for multi-threaded computations as well as for iterative applications for numerical sim- ulation on multi-CPUs / multi-GPUs; 3/ fault-tolerant protocols. Publicly available at http://kaapi.gforge.inria.fr under CeCILL licence. Kaapi has won the 2008 Plugtest organized by Grid@Works. Kaapi provides ABI compliant implementations of Quark (PLASMA, Linear Algebra, Univ. of Tennesse) and libGOMP (GCC runtime for OpenMP). Direct competitors with 1/: Quark, StarSs (UPC, BSC), OpenMP. Direct competitors with 2/: StarSs, StarPU (Inria RUNTIME), Quark, OpenACC runtimes. Direct competitors providing 3/: Charm++, MPI.

- ACM: D.1.3
- License: CeCILL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++, Fortran
- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4

5.2. FlowVR

Participant: Bruno Raffin [correspondant MOAIS].

- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4

- Additional information: FlowVR (http://flowvr.sf.net, coordinator B. Raffin) provides users with the necessary tools to develop and run high performance interactive applications on PC clusters and Grids. The main target applications include virtual reality, scientific visualization and Web3D. FlowVR enforces a modular programming that leverages software engineering issues while enabling high performance executions on distribued and parallel architectures. FlowVR is the reference API for Grimage. See also the web page http://flowvr.sf.net. The FlowVR software suite has 3 main components:
  - FlowVR : The core middleware library. FlowVR relies on the data-flow oriented programming approach that has been successfully used by other scientific visualization tools.
  - FlowVR Render : A parallel rendering library.
  - VTK FlowVR : a VTK / FlowVR / FlowVR Render coupling library.

5.3. TakTuk - Adaptive large scale remote execution deployment

Participants: Guillaume Huard [correspondant], Pierre Neyron.

- Characterization of Software : A-2 / SO-3 / SM-5 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
  - web site: http://taktuk.gforge.inria.fr, Coordinator G. Huard
  - Objective of the software: TakTuk is a tool for deploying parallel remote executions of commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and sets up an interconnection network to transport commands and perform I/Os multiplexing/demultiplexing. The TakTuk mechanics dynamically adapt to environment (machine performance and current load, network contention) by using a reactive work-stealing algorithm that mixes local parallelization and work distribution.
  - Users community: TakTuk is a research open source project available in the Debian GNU/Linux distribution (package taktuk) used in lower levels of Grid5000 software architectures (nodes monitoring in OAR, environment diffusion in Kadeploy). The community is small : developers and administrators for large scale distributed platforms, but active.
  - Positioning: main competing tools are pdsh (but uses linear deployment) and gexec (not fault tolerant, requires installation), for more details : B. Claudel, G. Huard and O. Richard. TakTuk, Adaptive Deployment of Remote Executions. In Proceedings of the International Symposium on High Performance Distributed Computing (HPDC), 2009. TakTuk is the only tool to provide to deployed processes a communication layer (just like an MPIrun, but not tied to a specific environment) and synchronization capabilities.

5.4. KRASH - Kernel for Reproduction and Analysis of System Heterogeneity

Participants: Guillaume Huard [correspondant], Swann Perarnau.

- Characterization of Software : A-1 / SO-3 / SM-4 / EM-2 / SDL-3
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
  - web site: http://krash.ligforge.imag.fr
  - Objective of the software: Krash is a tool to create a synthetic heterogeneity on top of a dedicated system while preserving the OS state and algorithms (no modification). It makes use of the control groups (cgroups) in Linux kernel newer than version 2.6.24 to create a dynamic CPU load enforced no matter how many applications are running in parallel.
  - Users community: Research open source project, small community: developers of parallel applications in heterogeneous contexts.
  - Positioning: Competing tool is Wreakavoc (less scalable, less precise), more details in : Swann Perarnau and Guillaume Huard. Krash: Reproducible cpu load generation on many-core machines. In IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2010.

5.5. Cache Control

Participants: Guillaume Huard [correspondant], Swann Perarnau.

- Characterization of Software : A-1 / SO-3 / SM-3 / EM-2 / SDL-3
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
  - web site: http://ccontrol.ligforge.imag.fr/
  - Objective of the software: Cache Control is a Linux kernel module enabling user applications to restrict their memory allocations to a subset of the hardware memory cache. This module reserves and exports available physical memory as virtual devices that can be mmap'd to. It gives to calling processes physical memory using only a subset of the cache (similarly to page coloring). It actually creates cache partitions that can be used simultaneously by a process to control how much cache a data structure can use.
  - Users community: Research open source project, small community: developers wanting to measure or tune the cache usage of their applications. Does not apply to recent NUCA caches.
  - Positioning: Competing tool is ULCC which does the same thing at the runtime level, more details in : Swann Perarnau, Marc Tchiboukdjian, and Guillaume Huard. Controlling cache utilization of hpc applications. In International Conference on Supercomputing (ICS), 2011.

5.6. GGen

Participants: Guillaume Huard [correspondant], Swann Perarnau.

- Characterization of Software : A-2 / SO-4 / SM-4 / EM-2 / SDL-3
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
  - web site: http://ggen.ligforge.imag.fr/, Coordinator Swann Perarnau
  - Objective of the software: GGen is a free (GPL-compatible) command line application and library for generating and analyzing directed acyclic graphs. Designed primarily to be used in simulations of scheduling algorithms, it helps researchers understand fully the nature of the graphs generated. It implements the most known graph generation algorithms enabling comparisons betweens them.
  - Users community: Research open source project, task scheduling community: ggen provides a meaningful way to generate test cases.
  - Positioning: To our knowledge, there's no competing tool, more details in : Daniel Cordeiro, Grégory Mounié, Swann Perarnau, Denis Trystram, Jean-Marc Vincent, and Frédéric Wagner. Random graph generation for scheduling simulations. In International ICST Conference on Simulation Tools and Techniques (SIMUTools), 2010.

5.7. Triva

Participants: Guillaume Huard [correspondant], Lucas Schnorr.

- Characterization of Software : A-2 / SO-4 / SM-5 / EM-3 / SDL-3
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-3
- Additional information:
  - web site: http://triva.gforge.inria.fr/, Coordinator, Lucas Schnorr

  - Objective of the software: Triva is an open-source tool used to analyze traces (in the pajé format) registered during the execution of parallel applications. The tool serves also as a sandbox to the development of new visualization techniques.
  - Users community: Research open source project, applications developers, especially parallel applications.
  - Positioning: Main competing tools are Vampir (classical 2D Gantt charts) and Tau (less advanced agregation techniques), more details in : A Hierarchical Aggregation Model to achieve Visualization Scalability in the analysis of Parallel Applications. Lucas Mello Schnorr, Guillaume Huard, Philippe Olivier Alexandre Navaux. Parallel Computing. Volume 38, Issue 3, March 2012.

5.8. OAR

Participants: Pierre Neyron [correspondant MOAIS], Grégory Mounié.

- Characterization of Software : A-5 / SO-3 / SM-4 / EM-4 / SDL-5
- Own Contribution: DA-3 / CD-2 / MS-1 / TPM-1
- Additional information: OAR (http://oar.imag.fr, Coordinator O. Richard, Inria MESCAL) is a batch scheduler. The MOAIS team develops the central automata and the scheduling module that includes successive evolutions and improvements of the policy.OAR is used to schedule jobs both on the CiGri (Grenoble region) and Grid50000 (France) grids. CiGri is a production grid that federates about 500 heterogeneous resources of various Grenoble laboratories to perform computations in physics. MOAIS has also developed the distributed authentication for access to Grid5000.

5.9. SOFA

Participant: Bruno Raffin [correspondant].

Inria category: ????

- Characterization of Software : A-5 / SO-4 / SM-4 / EM-4 / SDL-5
- Own Contribution: DA-2 / CD-2 / MS-1 / TPM-1
- Additional information: SOFA (http://www.sofa-framework.org/, Coordinator F. Faure, Inria IMAGINE) is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop newer algorithms, but can also be used as an efficient prototyping tool. Moais contributes to parallelization of kernel algorithms used in the simulation.
- ACM: J.3
- Programming language: C/C++

5.10. LinBox

Participants: Clément Pernet [correspondant], Thierry Gautier.

- Characterization of Software : A-3 / SO-4 / SM-2 / EM-3 / SDL-5
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4
- Additional information:
  - web site: http://linalg.org
  - Objective of the software: LinBox is an open-source C++ template library for exact, high-performance linear algebra computations. It is considered as the reference library for numerous computations (such as linear system solving, rank, characteristic polynomial, Smith normal forms,...) over finite fields and integers with dense, sparse, and structured matrices.

- The LinBox group is an international collaboration (USA: NCSU, UDel; Canada: U Waterloo, U Calgary; France: LIP, LIRMM, LJK and LIG). Articles related to the library have been published in the main Conferences of the area: ISSAC, ICMS. MOAIS contributes to its development and more specifically to its parallelization in the context of ANR HPAC project. It is currently experiencing a major change of design, to better integrate parallelism.
- Users community: mostly researchers doing computational mathematics (number theory, cryptology, group theory, persistent homology. They use the library by either linking against it directly (the library is packaged in Debian, Fedora, etc) or withing the general purpose math software Sage (sagemath.org very broad diffusion) which includes LinBox as a kernel for exact linear algebra.

6. New Results

6.1. Work Stealing inside GPU

Graphics Processing units (GPU) have become a valuable support for High Performance Computing (HPC) applications. However, despite the many improvements of General Purpose GPUs, the current programming paradigms available, such as NVIDIA?s CUDA, are still low-level and require strong programming effort, especially for irregular applications where dynamic load balancing is a key point to reach high performances. We have introduced a new hybrid programming scheme for general purpose graphics processors using two levels of parallelism. In the upper level, a program creates, in a lazy fashion, tasks to be scheduled on the different Streaming Multiprocessors (MP), as defined in the NVIDIA?s architecture. We have embedded inside GPU a well-known work stealing algorithm to dynamically balance the workload. At lower level, tasks exploit each Streaming Processor (SP) following a data-parallel approach. Preliminary comparisons on data-parallel iteration over vectors show that this approach is competitive on regular workload over the standard CUDA library Thrust, based on a static scheduling. Nevertheless, our approach outperforms Thrust-based scheduling on irregular workloads.

6.2. XKaapi on top of Multi-CPU Multi-GPU

Most recent HPC platforms have heterogeneous nodes composed of a combination of multi-core CPUs and accelerators, like GPUs. Programming such nodes is typically based on a combination of OpenMP and CUDA/OpenCL codes; scheduling relies on a static partitioning and cost model. We have experiment XKaapi runtime system for multi-CPU and multi-GPU architectures, which supports a data-flow task model and a locality-aware work stealing scheduler. The XKaapi enables task multi-implementation on CPU or GPU and multi-level parallelism with different grain sizes. We demonstrate performance results on two dense linear algebra kernels, matrix product (GEMM) and Cholesky factorization (POTRF), to evaluate XKaapi on a heterogeneous architecture composed of two hexa-core CPUs and eight NVIDIA Fermi GPUs. Our conclusion is two-fold: First, fine grained parallelism and online scheduling achieve performance results as good as static strategies, and in most cases outperform them. This is due to an improved work stealing strategy that includes locality information; to a very light implementation of the tasks in XKaapi; and to an optimized search for ready tasks. Next, our XKaapi Cholesky is highly efficient on multi-CPU/multi- GPU due to its multi-level parallelism. Using eight NVIDIA Fermi GPUs and four CPUs, we measure up to 2.43 TFlop/s on double precision matrix product and 1.79 TFlop/s on Cholesky factorization; and respectively 5.09 TFlop/s and 3.92 TFlop/s in single precision. This is the first time that such a performance is obtained with more than four GPUs.

6.3. Formalizing the concept of cooperation

We study how to optimize scheduling problems for a large number of objectives, when multiple users are competing for common resources, with some appropriate notion of fairness between users. Formalizing the concept of cooperation in relation with multi-objective optimization, we can refine the classical methods in combinatorial optimization (that usually optimize one centralized objective) by introducing extra features (adding more objectives or constraints). The PhD thesis of Daniel Cordeiro [2] proposed various ways for handling this problem: multi-organization scheduling and its relaxed variant, impact of selfishness. In the same context, we investigated the field of Game Theory through the existence of Nash equilibria in some situations.

6.4. Fault-tolerance for large parallel systems

This PhD thesis of Slim Bouguerra [1] studied fault-tolerance issues for large parallel systems. We revisited, via a formal proof, the old well-known result which states that the optimal policy for exponential failure law is to put the check-points at periodic moments. We proposed new algorithms to handle check-points for any law in the input and variable check-point costs (JPDC paper).

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Grants with Industry

- Contract with EDF (2010-2013). High performance scientific visualization. Fund 1 postdoc and 1 PhD. Partners: Inria (MOAIS and EVASION), EDF R&D
- HiPeComp, NANO 2008-2012 contract with ST-MicroElectronics. The project HiPeCoMP (High Performance Components for MPSoC) consists in the development an coupling of: on the one hand, wait-free scheduling techniques (pre-partitioning and mapping, on-line work stealing) of component based multimedia applications on MPSoC architectures; and on the other hand, monitoring, debug and performance software tools for the programming of MPSoC with provable performances.
- CEA: Contract with CEA (2012): Europlexus Parallelization with KAAPI. Partners: Inria Rhônes-Alpes and CEA Saclay.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

- **ANR grant REPDYN (2010-2012).** High performance computing for structure and fluid computing. Partners: Inria Rhône-Alpes, CEA, ONERA, EDF, LaMSID lab from CNRS and LaMCoS lab from INSA Lyon.
- **ANR/JST grant PETAFLOW (2010-2012).** France/Japan international program. Peta-scale data intensive computing with transnational high-speed networking: application to upper airway flow. Inria Rhône-Alpes, Gipsa-lab from UJF, NITC (Japan), Cyber Center of Osaka, DITS (Osaka) and the Visualization Lab of Kyoto.
- **ANR grant EXAVIZ (2011-2015).** Large Scale Interactive Visual Analysis for Life Science. Partners: Inria Rhône-Alpes, Université d'Orléans, the LBT lab from IBPC, the LIMSI from Université d'Orsay, and the CEMHTI labs from CNRS.
- **ANR HPAC (2012-2015)**. High Performance Algebraic Computing. Coordinator: UJF (LJK/CASYS team). Partners: project-team MOAIS (Grenoble), project-team ARENAIRE (LIP, Lyon), project-team SALSA (LIP6, Paris), the ARITH group (LIRMM lab, Montpellier).

- **Equipex Kinovis (2012-2017)**. 2.6 Meuros. Large scale multi-camera platform (extension of the Grimage platform to 60 cameras, depth and X-ray cameras). Coordinator E Boyer, LJK Inria MORPHEO team. Partners: Inria Rhône-Alpes and the LJK, LIG, LADAF and GIPSA labs.

8.1.2. Competitivity Clusters

- CILOE, 2008-2012, Minalogic: This project is to develop tools and high level interfaces for compute-intensive applications for nano and micro-electronic design and optimizations. The partners are: two large companies CS-SI (leader), Bull; three small size companies EDXACT, INFINISCALE, PROBAYES; and four research units Inria, CEA-LETI, GIPSA-LAB, TIMA. For Moais, the contract funds the phD thesis of Jean-Noel Quintin.
- SHIVA, Minalogic 2009-2012 contract. This project aims at the development of a high througput backbone ciphering that ensures a high level of security for intranet and extranet communications over internet. The partners are: CS-SI (leader); 1 small size companies: Easii-IC (support for Xilinx FPGA) IWall-Mataru (key management), Netheos (customizable FPGA for ciphering); INRIA; CEA-LETI (security certification); Grenoble-INP (TIMA lab, integration of cryptography on FPGA); UJF (LJK and Institut Fourier: open cryptographic protocols and handshake; VERIMAG: provable security). Within Inria, the MOAIS and the PLANET teams provide the parallel implementation on a multicore pltaform of IP-Sec and coordination with hardware accelerators (Frog?s and GPUs). The contract funds the phD thesis of Ludovic Jacquin, coadvised by PLANET and MOAIS and a 1 year engineer (Fabrice Schuler, from 11/2010).
- SoC-Trace, Minalogic 2011-2014 contract. This project aims the development of tools for the monitoring and debug of mumticore systems on chip. Leader: ST-Microelectonic. Partners: Inria (Mescal, Moais); UJF (TIMA, LIG/Hadas); Magilem, ProBayes. Moais contributes with technics and tools for visual aggregation of application traces. The contract funds 1 phD thesis and 1 year engineer.

8.2. European Initiatives

8.2.1. FP7 Projects

8.2.1.1. VISIONAIR

Title: VISIONAIR

Type: CAPACITIES (Infrastructures)

Instrument: Combination of COLLABORATIVE PROJECTS and COORDINATION and SUPPORT ACTIONS (CPCSA)

Duration: February 2011 - January 2015

Coordinator: Grenoble-INP, France

Others partners: http://www.infra-visionair.eu/members.html

See also: http://www.infra-visionair.eu/

VISIONAIR European platform. With the Grimage platform, we participate to the European project Visionair which objective is to provide an infrastructure that gathers advanced visualization and interaction infrastructures. Visionair is leaded by Grenoble-INP (Frédéric Noel, G-Scop lab) and gathers 25 international partners from 12 countries; it has been funded in 2010 and start in Q1 2011.

8.2.2. Collaborations with Major European Organizations

- **ADT Vcore (2011-2013)**. Partners: Fraunhofer IGD (Darmstad), Inria IMAGINE and MOAIS (Grenoble), SHAMAN and MINT (Lille), VR4i (Rennes), IN SITU (Saclay), SED Sophia Antipolis. This project is currently an ADT Inria (funds IJD). Software infrastructure for advanced applications in augnmented and virtual reality.

8.3. International Initiatives

8.3.1. Inria International Partners

MOAIS has a long term collaboration with several universities in Brazil, and in particular with UFRGS, Porto Alegre and USP, Sao Paulo. Several mobility grants support these collaborations:

- Inria Diode-A associated team (2006-2011),
- CNRS/Cnpq (2011-2013).
- Inria/Cnpq (2008-2010),
- Capes/Cofecub (2006-2007, 2008-2009, 2010-2012),
- Associated International Laboratory LICIA (http://www.inf.ufrgs.br/licia) funded by CNRS (since 2011).

This collaboration is important to get access to high quality students. Classically students pursue their PhD in our team full or half time in "co-tutelle” (double graduation). These PhDs are almost all funded by Brazil. Over the 2008-2012 period, 5 PhD students (3 from UFRGS, 2 from USP) were advised at Moais. Initially based on experimented researcher exchanges, the increase of fundings enabled to involve Master students that usually stay 2-4 months in our team and often come back later for a PhD.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

- Wieslaw Kubiak (memorial Univiersity, New Foundland, Canada), invited prof UJF (2 months)
- Joseph Peters (SFU Vancouver, Canada, contract INP VOLVIC (3 months)

8.4.1.1. Internships

Julio TOSS (from Apr 2012 until Sep 2012)

Subject: A new programming paradigm for GPU

Institution: Universidade Federal do Rio Grande do Sul (Brazil)

Nikhil BANSAL (from Jun 2012 until Sep 2012)

Subject: Multi-objective optimization strategies for parallel multi-users applications

Institution: IIT Delhi (India)

9. Dissemination

9.1. Scientific Animation

ISSAC 2012 : Treasurer and Local Arrangements committee (the coference held in Grenoble in July 2012)

ComPAS/Renpar 2013 : Chair of the Program Committee for the next edition of ComPAS/Renpar that will be held at Grenoble in January 2013.

ComPAS/Renpar 2013 : local organization of ComPAS/Renpar.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master: V Danjean, T Gautier: course "Parallel Programming" (M2), Grenoble University,.

Master: J-L. Roch co-director (Grenoble-INP) with P Elbaz-Vincent (Université Joseph Fourier, Math. Dept) of the Master "SCCI Security, Cryptology and Coding of Information Systems" (M2) joined between UJF and INP Grenoble Universities. This Master, started in 2001, is taught in English from sept 2007 (international Master).

Master: C. Pernet and Denis Trystram are responsible of the first year (M1) of the international Master of Science in Informatics at Grenoble (MOSIG-M1).

Master: J-L. Roch, "Security models" 24h (M2), Grenoble University

Master: D. Trystram, P.-F. Dutot, J.-L. Roch, "Complexity, approximation theory and randomization" master course (M2) at Grenoble University

Master: François Broquedis. 192 hours per year. 192 hours per year. Engineering school Grenoble-INP/Ensimag, 1st year/L3 and Master (M1/2nd year and M2/3rd year).

Master: Vincent Danjean. 242 hours per year. Licence (third year) and Master (first and second year) at Joseph Fourier University. First to third year of engineering school at Polytech' Grenoble.

Master: Pierre-François Dutot. 226 hours per year. Licence (first and second year) at IUT2/UPMF (Institut Universitaire Technologique de l'Université Pierre Mendès-France) and 9 hours Master M2R-ISC Informatique-Systèmes-Communication at Joseph Fourier University.

Master: Guillaume Huard. 242 hours per year. Licence (first and third year) and Master (first year) at Joseph Fourier University.

Master: Grégory Mounié. 242 hours per year. Master (first year) and Computer Science for Non Computer Scientist Post-Master at Engineering school ENSIMAG and Dept TELECOM, Grenoble-INP.

Master: Clement Pernet. 210 hours per year. University J. Fourier. Master (first year and second year) and Licence (3rd year).

Master: Bruno Raffin. 22 hours per year. Master at Université d'Orléans and Polytech'Grenoble.

Master: Jean-Louis Roch. 242 hours per year. Engineering school Grenoble-INP/Ensimag and Master (M1/2nd year and M2/3rd year)

Master: Denis Trystram. 200 hours per year in average, mainly at first level of Engineering School.

Master: Frédéric Wagner. 220 hours per year. Engineering school ENSIMAG, Grenoble-INP (M1/2nd year and M2/3rd year) (190h) ; Master DESS/M2-P SCCI Security (30h).

10. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] M. S. Bouguerra. *Tolérance aux pannes dans des environnements de calcul parallèle et distribué : optimisation des stratégies de sauvegarde/reprise et ordonnancement*, University of Grenoble, April 2012.

[2] D. Cordeiro. *The impact of cooperation on new high performance computing platforms*, Université de Grenoble, February 2012, http://tel.archives-ouvertes.fr/tel-00690908.

Articles in International Peer-Reviewed Journals

[3] M. S. Bouguerra, D. Trystram, F. Wagner. *Complexity Analysis of Checkpoint Scheduling with Variable Costs*, in "IEEE Transactions on Computers", 2012, vol. 99, n^{o} PrePrints, http://doi.ieeecomputersociety.org/10.1109/TC.2012.57.

[4] M. DURAND, P. MARIN, F. FAURE, B. RAFFIN. *DEM-based simulation of concrete structures on GPU*, in "European Journal of Environmental and Civil Engineering", August 2012, p. 1-13 [*DOI :* 10.1080/19648189.2012.716590], http://hal.inria.fr/hal-00733674.

[5] A. FÖLLING, J. LEPPING. *Knowledge Discovery for Scheduling in Computational Grids*, in "WIREs Data Mining and Knowledge Discovery", 2012, vol. 2, n^o 4, p. 287-297, http://doi.wiley.com/10.1002/widm.1060.

[6] E. JEANNOT, É. SAULE, D. TRYSTRAM. *Optimizing Performance and reliability on heterogeneous parallel systems: Approximation algorithms and heuristics*, in "Journal of Parallel and Distributed Computing", 2012, vol. 72, n^o 2, p. 268-280, doi: 10.1016/j.jpdc.2011.11.003.

[7] J.-D. LESAGE, B. RAFFIN. *A Hierarchical Component Model for Large Parallel Interactive Applications*, in "Journal of Supercomputing", July 2012, vol. 60, n^o 3, p. 389-409, Extended version of NPC 2007 article., http://dx.doi.org/10.1007/s11227-008-0228-7.

[8] L. MELLO SCHNORR, G. HUARD, P. O. A. NAVAUX. *A hierarchical aggregation model to achieve visualization scalability in the analysis of parallel applications*, in "Parallel Computing", 2012, vol. 38, n^o 3, p. 91-110.

International Conferences with Proceedings

[9] M. BOUGERET, P.-F. DUTOT, K. JANSEN, C. ROBENEK, D. TRYSTRAM. *Tight approximation for scheduling parallel jobs on identical cluster*, in "APDCM (IPDPS workshop)", France, 2012, http://hal.inria.fr/hal-00738499.

[10] F. BROQUEDIS, T. GAUTIER, V. DANJEAN. *libKOMP, an Efficient OpenMP Runtime System for Both Fork-Join and Data Flow Paradigms*, in "IWOMP", Rome, Italy, jun 2012, p. 102-115.

[11] M. T. COMER, E. L. KALTOFEN, C. PERNET. *Sparse Polynomial Interpolation and Berlekamp/Massey Algorithm That Correct Outlier Errors in Input Values*, in "ISSAC '12: Proceedings of the 2012 international symposium on symbolic and algebraic computation", July 2012.

[12] J. V. FERREIRA LIMA, T. GAUTIER, N. MAILLARD, V. DANJEAN. *Exploiting Concurrent GPU Operations for Efficient Work Stealing on Multi-GPUs*, in "24rd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)", Columbia University, New York, United States, October 2012, http://hal.inria.fr/hal-00735470.

[13] T. GAUTIER, F. LEMENTEC. *XKaapi*, in "11-th African Conference on Research in Computer Science and Applied Mathematics", Algiers, Algeria, October 2012.

[14] A. GOLDMAN, Y. NGOKO, D. TRYSTRAM. *Malleable resource sharing algorithms for cooperative resolution of problems*, in "Proceedings of IEEE World Congress on Computational Intelligence", Brisbane, Australia, June 2012, p. 1438-1445.

[15] C. GRIMME, J. LEPPING. *An Approach to Instantly Use Single-objective Results for Multi-objective Evolutionary Combinatorial Optimization*, in "Learning and Intelligent Optimization Conference (LION6)", Lecture Notes in Computer Science, Springer, 2012, n^o 7219, p. 396-401.

[16] L. Jacquin, V. Roca, M. A. Kaafar, F. Schuler, J.-L. Roch. *IBTrack: an ICMP black holes tracker*, in "IEEE GLOBECOM", IEEE, 2012, http://hal.inria.fr/hal-00748153.

[17] L. Pilla, C. Pousa Ribeiro, D. Cordeiro, C. Mei, A. Bhatele, P. Navaux, F. Broquedis, J.-F. Mehaut, L. Kale. *A Hierarchical Approach for Load Balancing on Parallel Multi-core Systems*, in "Proceedings of the 41st International Conference on Parallel Processing, ICPP 2012", Pittsburgh, Pennsylvania, September 2012.

[18] L. Pilla, C. Pousa Ribeiro, P. Navaux, P. Coucheney, F. Broquedis, B. Gaujal, J.-F. Mehaut. *Asymptotically Optimal Load Balancing f0r Hierarchical Multi-Core Systems*, in "Proceedings of the 18th IEEE International Conference on Parallel and Distributed Systems, ICPADS", Singapore, December 2012.

[19] J.-N. Quintin, F. Wagner. *WSCOM: Online Task Scheduling with Data Transfers*, in "CCGRID", 2012, p. 344-351.

[20] J. Toss, T. Gautier. *A New Programming Paradigm for GPGPU*, in "EUROPAR 2012", Rhodes Island, Greece, August 2012.

Scientific Books (or Scientific Book chapters)

[21] J.-G. Dumas, J.-L. Roch, E. Tannier, S. Varrette. *Foundations of Coding: Compression, Encryption, Error-Correction*, Springer, 2012.

[22] J. Lepping. *Genetic Fuzzy Scheduling - Development of Rule-based Scheduling Strategies for Parallel Machines*, AV Akademikerverlag, July 2012, https://www.morebooks.de/store/de/book/genetic-fuzzy-scheduling/isbn/978-3-639-43762-1.

Research Reports

[23] M. Bougeret, P.-F. Dutot, D. Trystram, K. Jansen, C. Robenek. *Tight Approximation for Scheduling Parallel Job on Identical Clusters*, LIRMM, January 2012, http://hal.inria.fr/lirmm-00656780.

[24] 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, March 2012, http://hal.inria.fr/hal-00720815.

[25] J.-G. Dumas, C. Pernet. *Computational linear algebra over finite fields*, Grenoble University, April 2012, http://hal.inria.fr/hal-00688254.

[26] J. Emeras, V. Pinheiro, K. Rzadca, D. Trystram. *Fair Scheduling for Multiple Submissions*, LIG, Grenoble, France, 2012, n^{o} RR-LIG-033, http://rr.liglab.fr/research_report/RR-LIG-033_orig.pdf.

[27] T. Gautier, F. Lementec, V. Faucher, B. Raffin. *X-Kaapi: a Multi Paradigm Runtime for Multicore Architectures*, Inria, February 2012, n^{o} RR-8058, 16, http://hal.inria.fr/hal-00727827.

[28] L. Jacquin, V. Roca, M. A. Kaafar, F. Schuler, J.-L. Roch. *IBTrack: An ICMP Black holes Tracker*, Inria, March 2012, http://hal.inria.fr/hal-00695746.

Other Publications

[29] G. HUARD. *Taktuk: efficient large scale deployment of remote executions*, 2012, Inria Forge, https://gforge.inria.fr/projects/taktuk/.