IN PARTNERSHIP WITH:
CNRS

Institut polytechnique de Grenoble

Université Joseph Fourier (Grenoble 1)

Activity Report 2011

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, Virtual Reality, Adaptive Algorithm, Fault Tolerance, Grid'5000, Parallel Algorithms

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]

Technical Staff

Christian Séguy [CNRS/LIG Engineer, 40%]
Pierre Neyron [CNRS/LIG, Research engineer, 40%]
Eric Amat [2010-2012. INRIA Grant (ADT VGATE)]
Fabien Le Mentec [2009, Engineer ADT Kaapi]
Fabrice Schuler [2010, Engineer Minalogic contract SHIVA]

PhD Students

Mohamed-Slim Bouguerra [2008, INRIA Cordi]
Daniel Cordeiro [2007, Alban scholarship]
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]
Jean-Noel Quintin [2008, Minalogic CILOE contract]

Post-Doctoral Fellow

Joachim Lepping [2011, Inria contract]

Administrative Assistants

Ahlem Zammit-Boubaker [INRIA Administrative Assistant, 50% (till sept.)]
Annie Simon [INRIA Administrative Assistant, 40% (from oct.)]
Annie-Claude Vial-Dallais [CNRS Administrative Assistant, 40%]

Other

Xavier Martin [2011–2014, Apprenti]

2. Overall Objectives

2.1. Introduction

The objective 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.
The research directions of the MOAIS team are focused on the scheduling problem with a multi-criteria performance objective: precision, reactivity, resources comsuption, reliability, ... The originality of the MOAIS approach is to use the application's adaptability to enable its control by the scheduling. The critical points concern designing adaptive malleable algorithms and coupling the various components of the application to reach interactivity with performance guarantees.

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.

For those themes, the scientific methodology of MOAIS consists in:

- designing algorithms with provable performance on generic theoretical models;
- 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);
- 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/).

2.2. Highlights

- Denis Trystram received the *Google Research Award* for his contributions within Moais on efficent management of distributed resources and multicriteria scheduling on emerging parallel platforms.
- The book *Foundations of Coding: Compression, Encryption, Error-Correction* (426 p.), cowritten by Jean-Guillaume Dumas, Jean-Louis Roch, Eric Tannier and Sébastien Varrette is published by Springer (should be available in early 2012).

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.

This adaptation technique is now integrated in softwares that we are developing with external partners within contracts. In particular, in partnership with STM within the Minalogic SCEPTRE contract we have developed a specific optimized C interface, dedicated to stream computation for multi-processor system on chips (MPSoC); this interface is named AWS (Adaptive Work-Stealing).

Besides, we developed a parallel implementation of the C++ Standard Template Library STL on top of Kaapi; this library, named KaSTL, provides adaptive parallel algorithms for distributed containers (such as transform, foreach and findif on vectors). By optimizing the work-stealing to our adaptive algorithm scheme, a new non-blocking (wait-free) implementation of Kaapi has been designed. A first prototype of this C library, named X-KaapiThe benchmarks experimented on SMPs and NUMAs architectures provides good performances with respect to concurrent libraries MCSTL, PaSTL, Intel TBB, and Cilk+, while improving the grain where parallelism can be exploited.

Extensions concern the development of algorithms that are both cache and processor oblivious. The processor algorithms proposed for prefix sums and segmentation of an array are cache oblivious too. We are currently working on sorting and mesh partitioning within a collaboration with the CEA.

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 2 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: 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. 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 PERCEPTION project focuses on developing solutions to enable real time 3D modeling from multiple cameras using a PC cluster. An operational code base was transferred to the 4DViews Start-up in September 2007. 4DViews is now selling turn key solutions for real-time 3D modeling. Recent developments take two main directions:
  - Using a HMD (Head Mounted Display) and a Head Mounted Camera to provide the user a high level of interaction and immersion in the mixed reality environment. Having a mobile camera raises several concerns. 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 DALIA we are developing a collaborative application where a user at Bordeaux (iParla project-team) using a real time 3D modeling platform can meet in a virtual world with a user in Grenoble also using a similar platform. We rely on the Grid'5000 dedicated 10 Gbits/s network to enable a low latency. 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.

  On these issues, Benjamin Petit started a Ph.D. in October 2007, co-advised by Edmond Boyer (PERCEPTION) and Bruno Raffin.
- Real time physical simulation. We are collaborating with the EVASION project on the SOFA simulation framework. Everton Hermann, a Ph.D. co-advised by François Faure (EVASION) 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). The goal is to have the cores collaborating to efficiently share the L3 cache for a better performance than with a more traditional approach that leads to split the L3 cache between the cores. We are obtaining good performance gains with a parallel iso-surface extraction algorithm. This work is the main focus of Marc Tchiboukdjian Ph.D.

4.2. Code Coupling and Grid Programming

Participants: 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.

4.3. 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. To open the grid usage to commercial applications from small-size companies (namely in the field of micro and nano-technology within the global competitiveness cluster Minalogic in Grenoble), we are currently studying the scalability issues related to systematic ciphering of all components of a distributed application in relation with CS Group (thesis of Thomas Roche, CIFRE scholarship). Dedicated to multicore architectures, an adaptive parallelization of a block cipher (based on counter mode) has been evaluated. Within the SHIVA contract and the Ph.D. of Ludovic Jacquin (coadvised with the PLANETE EPI), we develop a high-rate systematic ciphering architecture based on the coupling of a multicore architecture with security components (FPGA and smart card).

4.4. 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 reusing the scheduling technologies developed in MOAIS for embedded systems.

In the framework of our cooperation with STM (Serge de Paoli, Miguel Santana) and within the SCEPTRE project (global competitiveness cluster MINALOGIC/EMSOC), Julien Bernard in his thesis (grant cofunded by STM and CNRS) provides a specialized version of Kaapi for adaptive stream computations, named AWS, on MPSoCs platforms. AWS has been implemented and is being evaluated on two platforms: STM-8010 (3 processors on chip) and a cycle-approximate simulation (TIMA, Frédéric Pétrot). We are also studying self-specialized implementation of work-stealing from an abstract description (from SPIRIT standard) of the MPSoC architecture. Since those applications are developed based on component models, we are developing adaptive schedules for such component applications within the Nano2012 HiPeCoMP contract.

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. For instance today machines can be built with several dual-core processors and graphical processing units. New architectures, like the Cell processors, also integrate several computing units. First works concern balancing work load on multi GPU and CPU architectures workload balancing for scientific visualization problems.

5. Software

5.1. KAAPI

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

KAAPI means Kernel for Adaptative, Asynchronous Parallel and Interactive programming. It is a C++ library that allows to execute multithreaded computation with data flow synchronization between threads. The library is able to schedule fine/medium size grain program on distributed machine. The data flow graph is dynamic (unfold at runtime). Target architectures are clusters of SMP machines.Main features are * It is based on work-stealing algorithms ; * It can run on various processors ; * It can run on various architectures (clusters or grids) ; * It contains non-blocking and scalable algorithms.
See also the web page http://kaapi.gforge.inria.fr.

- ACM: D.1.3
- License: CeCILL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++, Fortran

5.2. OAR

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

OAR is a batch scheduler developed by Mescal team (correspondant: Olivier Richard). 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.
See also the web page http://oar.imag.fr.

5.3. SOFA

Participant: Bruno Raffin [correspondant].

SOFA 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. based on an advanced software architecture, it allows to:- create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA- modify most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an xml file- build complex models from simpler ones using a scene-graph description- efficiently simulate the dynamics of interacting objects using abstract equation solvers- reuse and easily compare a variety of available methods.
See also the web page http://www.sofa-framework.org/.

- ACM: J.3
- Programming language: C/C++

5.4. TakTuk - Adaptive large scale remote execution deployment

Participants: Guillaume Huard [correspondant], Pierre Neyron.

TakTuk is a tool for deploying remote execution commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and set up an interconnection network to transport commands and perform I/Os multiplexing/demultiplexing. The TakTuk algorithms dynamically adapt to environment (machine performance and current load, network contention) by using a reactive algorithm that mix local parallelization and work distribution. Characteristics:

- adaptivity: efficient work distribution is achieved even on heterogeneous platforms thanks to an adaptive work-stealing algorithm
- scalability TakTuk has been tested to perform large size deployments (hundreds of nodes), either on SMPs, regular clusters or clusters of SMPs
- portability: TakTuk is architecture independent (tested on x86, PPC, IA-64) and distinct instances can communicate whatever the machine they're running on
- configurability: mechanics are configurable (deployment window size, timeouts, ...) and TakTuk outputs can be suppressed/formatted using I/O templates Outstanding features:
- auto-propagation: the engine can spread its own code to remote nodes in order to deploy itself
- communication layer: nodes successfully deployed are numbered and perl scripts executed by TakTuk can send multicast communications to other nodes using this logical number
- information redirection: I/O and commands status are multiplexed from/to the root node. http://taktuk.gforge.inria.fr under GNU GPL licence.

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

Participants: Guillaume Huard [correspondant], Swann Perarnau.

KRASH is a tool for reproducible generation of system-level CPU load. This tool is intended for use in shared memory machines equipped with multiple CPU cores that are usually exploited concurrently by several users. The objective of KRASH is to enable parallel application developers to validate their resources use strategies on a partially loaded machine by replaying an observed load in concurrence with their application. To reach this objective, KRASH relies on a method for CPU load generation which behaves as realistically as possible: the resulting load is similar to the load that would be produced by concurrent processes run by other users. Nevertheless, contrary to a simple run of a CPU-intensive application, KRASH is not sensitive to system scheduling decisions. The main benefit brought by KRASH is this reproducibility: no matter how many processes are present in the system the load generated by our tool strictly respects a given load profile. This last characteristic proves to be hard to achieve using simple methods because the system scheduler is supposed to share the resources fairly among running processes. http://krash.ligforge.imag.fr under GNU GPL licence.

5.6. Cache Control

Participants: Guillaume Huard [correspondant], Swann Perarnau.

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.

6. New Results

6.1. Kaapi

New version of Kaapi, called X-Kaapi, has been released. The kernel is written in C for hypothetical required from embedded system. On top of the kernel, several APIs co-exist: a template based C++ library called Kaapi++; a C API; a Fortran API; and a compiler that transform a source code annotated with pragma directive to a source code with calls to the runtime library function. The compiler works with C and a subset of C++. http://kaapi.gforge.inria.fr

6.2. Multi-criteria optimization

The main idea is the development of a methodology to generate a reasonable set of approximated Pareto' solutions (closed to the best achievable solutions). Especially, we have applied this methodology to better take into account users' criteria than the other existing methods offer. We have also studied the problem of selection of best algorithms in a portfolio. This research axis is currently enforced by the INRIA postdoc position of Joachim Lepping where we have started to include a learning process to select the best algorithm on a given instance.

6.3. Stochastic models for optimizing checkpoint protocol

After our past studied on design of origin checkpoint protocols, we have proposed a new stochastic performance model of the parallel execution in presence of failures. Thanks to this formulation, we are able to optimize several criteria (the time lost due to failure; the expected completion time) by making right decision of the date of each checkpoint. The model is general and it does not take into account the failure distribution law and accept variable checkpoint time estimation, which is important for dynamic parallelism applications.

6.4. Work stealing scheduling algorithm taking care of communication

On some applications, the amount of data transfers can be high. To minimize the amount of data transfers during the execution, Jean-Noel Quintin has developed an algorithm called WSCOM which uses the DAG structure of the application. For each steal request, the work-stealing algorithm tries to balance the load between the thief and the stolen processor. Thus, WSCOM tries to divide the work on the stolen processor into two parts with a small number of edges between the two parts. This cutting is done with a negligible overhead at each steal request. This algorithm has been implemented in a tool called DSMake. This tool executes the set of tasks described by a Makefile on a distributed platform. In addition, I have developed a simulator to validate algorithm performance and its behavior. We compared WSCOM and severals static list-scheduling algorithms. The comparison shows that WSCOM outperforms list-scheduling algorithms, on clusters with some network congestion.
Besides, based on SIPS analysis of work stealing, Stefano Mor in his thesis compared the influence of the choice of the stolen tasks on the number of steal operations, distinguishing unsuccessful and successful steals. While standard bounds are related to unsuccessful steals, they are pessimistic with respect to the number of successful steals that define intensive data communications.

6.5. Homomorphic coding for soft error resilience

We extended our results for fault-tolerant modular computations in two directions. To improve the correction rate of Reed-Solomon codes, power-decoding techniques consist in augmenting the number of syndrom equations by raising the received word to successive powers. The correction is done by a generalization of Berlekamp-Massey algorithm acting on multiple sequences. This method is, if not equivalent, at least very close to the list-decoding proposed by Sudan in its first version, in particular, error correction rates are identical. We improve the power-decoding method by reformulation into a vector rational function reconstruction, with benefit from fast polynomial matrix arithmetic. Besides, for basic exact linear algebra computations (eg dense linear system), we designed interactive protocols between a trusted platform and a non trusted one for resilience to soft-errors.

6.6. Chimeric algorithms design

To reach provable muticriteria performance, we used the coupling of various algorithms that adapt in several contexts: recursive cascading of both sequential and parallel algorithms with work-stealing; coupling specific algorithms on heterogeneous platforms (eg CPU/GPU); interactive distributed computations; fault-tolerant computations by coupling both a trustfully platform with low computation bandwidth and and an unreliable computing platform with high bandwitdh. A unification work is currently developed for the design of a chimeric algorthms that is composed of the parts of multiple algorithms, interactively cascaded to achieve provable multicriteria performance.

7. Contracts and Grants with Industry

7.1. Contracts 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

8. Partnerships and Cooperations

8.1. Regional Initiatives

- CILOE, 2008-2011, 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.
- HiPeComp, NANO 2008-2012 contract. 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. For Moais, the contract funds the phD thesis of Christophe Laferrière who started on 1/9/2009.
- SHIVA, Minalogic 2009-2012 contract. This project aims at the development of a high throughput 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. The contract funds 1 phD thesis and 1 year engineer.

8.2. National Initiatives

- **ANR EXAVIZ (2011-2015).** Large-scale interactive visual analysis for life sciences and materials. Partners: project-team INRIA MOAIS, LIFO-lab Université d'Orléans, Laboratoire de Biochimie Théorique de l'IBPC, the LIMSI lab and the CEMHTI.
- **ANR REPDYN (2010-2012).** Scaling high performance computations in fluid and structure transient dynamics. Partners: project-teams INRIA MOAIS and EVASION, CEA, ONERA, EDF, LaMSID lab CNRS and LaMCoS lab at INSA Lyon.
- **ANR PETAFLOW (2010-2012).** Objet : peta-scale data intensive computing with transnational high-speed networking: application to upper airway flow. Programme ANR blanc France/Japon. Partners: l'équipe-projet INRIA MOAIS, le LIP de l'ENS Lyon, le Gipsa-lab de l'UJF, le NITC (japon), le Cyber Center d'Osaka, le DITS (Osaka), le Visualization Lab de Kyoto.
- PEPS LINBOX. 2010-2011. High Performance Library for Computer Algebra . Coordinator: C. Pernet. Partners: LIP (Lyon), LJK (Grenoble), LIRMM (Montpellier).
- New accepted ANR HPAC (2012-2015). High Performance Algebraic Computing. Coordinator: Jean-Guillaume Dumas (CASY team, LJK, Grenoble). Partners: project-team MOAIS (Grenoble), team CASYS (LJK, Grenoble), project-team ARENAIRE (LIP, Lyon), project-team SALSA (LIP6, Paris), the ARITH group (LIRMM lab, Montpellier).

8.3. European Initiatives

8.3.1. FP7 Projet

8.3.1.1. 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)

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.

8.4. International Initiatives

8.4.1. INRIA Associate Teams

8.4.1.1. DIODEA

Title: Paralllel and distributed computing, scalability and visualization

INRIA principal investigator: Bruno Raffin

International Partner:

Institution: Federal University of Rio Grande del Sul (Brazil)

Laboratory: Instituto de Informática

Researcher: Philippe Navaux

Duration: 2006 - 2011

See also: http://diodea.imag.fr/

The French research teams MOAIS and MESCAL, Grenoble, INRIA, and the Brazilian University UFRGS, Porto Alegre closely collaborate since 1992. This collaboration is centered on: Grid computing tools related to system and application deployment, job scheduling, execution monitoring and visualisation ; Modeling, evaluating and experimenting on large scale computer systems (performance evaluation, experimentations, simulation, emulation) ; New parallel programming paradigms: work stealing, fault tolerance, processor and cache oblivious algorithms, multi-core and multi-GPU programming. Frequent visits between partners and numerous co-adviced Master and Ph.D. students make it a really fruitful collaboration. It as a strong influence on the development of many of our software tools, including KAAPI, OAR, Kadeploy, Taktuk. We also share some of our computing resources. The cluster from UFRGS was integrated in 2009 as the first non european non of the Grid?5000 french experimental grid.
The success of the associated team leads to the creation of the first *Laboratoire International Associé* (LIA) in computer science between the French CNRS and the Brazil.

8.4.2. Brazil

CAPES/COFECUB n° Ma660/10 (2010-2013) on the management of resources for parallel computing on a grid. Partners: University of Sao Paulo, project MOAIS.

8.5. Hardware Platforms

8.5.1. The GRIMAGE platform

The GrImage platform (http://grimage.inrialpes.fr) gathers a network of cameras and a PC cluster. It is dedicated to interactive applications. GrImage is co-leaded by the Moais and Perception projects . It is the milestone of a strong and fruitful collaboration between Moais and Perception (common publications, software and application development).

GrImage (Grid and Image) aggregates commodity components for high performance video acquisition, computation and graphics rendering. Computing power is provided by a PC cluster, with some PCs dedicated to video acquisition and others to graphics rendering. A set of digital cameras enables real time video acquisition. The main goal is to rebuild in real time a 3D model of a scene shot from different points of view. Visualization can be performed using a head mounted display for first-person interactions or on a multi-projector display-wall for high resolution rendering.

Since July 2009, the computing cluster was upgraded through grants from INRIA and CNRS-LIG. Grimage uses some specific nodes from the Digitalis machine capable of hosting several daughter boards (mainly video acquisition and graphics cards). It relies on Intel Nehalem processors and a high speed Infiniband network. This integrated approach will enable to test interactive applications using a very high number of processing resources as other nodes from the Digitalis machine can be reserved if needed.

8.5.2. The Digitalis machine

Digitalis is a 780 cores cluster based on Intel Nehalem processors and Infiniband network located at INRIA Rhône-Alpes. Digitalis has been designed to suit both the needs for batch computations and interactive applications. As mentioned before, one rack is dedicated to nodes hosting video acquisition boards and graphics cards. These nodes are mainly used for the Grimage platform, but can also be used for batch computing. Additional nodes with Nvidia Tesla GPUs have been installed.

By having a single unified machine for batch and interactive computing we expect to better use the available resources, favor the emergence of high performance applications integrating interactive steering and vice versa enable the development of a new generation of interactive 3D applications using a significantly larger number of CPUs and GPUs that what has been done so far on the Grimage platform.

8.5.3. Multicore Machines

MOAIS invested in 2006 on two multicore architectures

- A 8-way 16-cores machine equipped with Itanium processors.
- A 8-way 16-cores machine equipped with dual core processors (total of 8 sockets) and 2 GPUs.

These set of machines have been extended in 2010 with a new machines:

- A 8-way, 48-cores machine equipped with 12-core AMD processors (total of 4 sockets)
- A 6-cores machine equipped with 8 GPUs

These machines enables us to keep-up with the evolution of parallel architectures and in particular today's availability of large multi-core machines. They are used to develop and test parallel adaptive algorithms taking advantage of the processing power provided by the multiple CPUs and GPUs available.

9. Dissemination

9.1. Animation of the scientific community

- 2011 2010 Chair / Symposium co-chair
  - –
- 2011 Program committee
  - EGPGV (Eurographics Symposium on Parallel Rendering and Visualization)
    - Program Committee member since 2007
  - Eurographics 2012 (short papers topic)
  - IEEE VR 2008-2012 (IEEE Conference on Virtual Reality). Co-chair of exhibition in 2012

  - VRC 2011
  - WEHA 2011 (Workshop on Exploitation of Hardware Accelerators)
  - ICAT 2011 (21st International Conference on Artificial Reality and Telexistence)
  - SEARIS 2011 (Fourth Workshop on Software Engineering and Architectures for Realtime Interactive Systems)
  - ISVC 201 et 2012 (International Symposium on Visual Computing)
  - SVR 2011 (Symposium on Virtual and Augmented Reality), Brazil
  - PAPP 2011 (International Workshop on Applications of Declarative and Object-oriented Parallel Programming)
  - CLCAR 2011 (Conferencia Latinamericana de Computatición de Alto Rendimiento)
  - RenPar 2011 (20ièmes Rencontres Francophones du Parallelisme), may 10-13, 2011, Saint Malo, France
  - HCW'2011 (20th IEEE Heterogeneous Computing Workshop) may 2011, Anchorage, Alaska, USA
  - LSAP 2011 (3rd Workshop on Large-Scale System and Application Performance) june 2010, San Jose, USA
  - OPTIM'11 (Workshop on Opt. Issues in Energy Efficient Distributed Systems) july 4-8, 2011, Istanbul, Turkey
  - ISPDC (10th Internat Symposium on Parallel and Distributed Computing) july 6-8, 2011, Cluj-Napoca, Romania
  - IC3 2011 (4th International Conference of Contemporary Computing) august 8-10, 2011, New Delhi, India
  - ParCo'2011, august 30 - sept. 2, 2011, Ghent, Belgium
  - ScalSol (scalable solutions for greenIT), august 31 - sept. 2, 2011, Pafos, Cyprus
  - WAOA 2011, september 8-9, Saarbruecken, Germany
  - PPAM 2011, september 10-14, 2011, Torun, Poland
  - LaSCoG (7th Workshop on Large Scale Computations on Grids) september 2011, Torun, Poland
  - New perspectives in scheduling theory, october 9-14, 2011, Hangzhou, China
  - 23th SBAC-PAD, october 26-29, 2011, Esperito Santo, Brazil

- 2011 Other.
  - Steering Board member of EGPGV 2011 (Eurographics Symposium on Parallel Rendering and Visualization)
  - Local chair of EuroPar 2011 (Parallel and Distributed Programming), Bordeaux, France

9.2. Teaching

Master M1. Introduction à la visualisation scientifique et à la programmation parallèle des architectures hybrides. 12 h de cours. Université de Saint Jacques de Compostelle, Espagne.

Master M2R. Architectures parallèles et protocoles hautes performances. 8h de cours. Université d'Orléans, France.

Master M1. Mathematics for Computer Science, Master International (MoSIG).

Master M2. Modèles de calcul, Complexité, Approximation et Heuristiques.

Master M1. Ordonnancement dans les systèmes informatiques et manufacturiers.

Algorithmique avancée" ENSIMAG 2A-apprentissage.

Ensimag - Master M1. Algorithmique et Programmation Orientée Objet.

Ensimag - Master M1. Information et Codage Numérique.

Ensimag - Master M1. Algorithmique avancée: Algorithmes d'approximation, parallèles et probabilistes Complexité.

Ensimag - Master M1. Codes: cryptographie, compression, correction d'erreurs.

Ensimag - Master M2. Security models: proofs and protocols.

Master M2R Mosig. Parallel Systems.

10. Bibliography

Major publications by the team in recent years

[1] P.-F. DUTOT, L. EYRAUD, G. MOUNIÉ, D. TRYSTRAM. *Scheduling on large scale distributed platforms: from models to implementations*, in "Internat. Journal of Foundations of Computer Science", april 2005, vol. 16, n^{o} 2, p. 217-237.

[2] S. JAFAR, A. KRINGS, T. GAUTIER. *Flexible Rollback Recovery in Dynamic Heterogeneous Grid Computing*, in "Transactions on Dependable and Secure Computing, (TDSC)", jan-mar 2009, vol. 6, n^{o} 1.

[3] J.-D. LESAGE, B. RAFFIN. *A Hierarchical Component Model for Large Parallel Interactive Applications*, in "Journal of Supercomputing", July 2008, Extended version of NPC 2007 article., http://dx.doi.org/10.1007/s11227-008-0228-7.

[4] G. MOUNIÉ, C. RAPINE, D. TRYSTRAM. *A 3/2-Dual Approximation Algorithm for Scheduling Independent Monotonic Malleable Tasks*, in "SIAM Journal on Computing", 2007, vol. 37, n^{o} 2, p. 401–412, http://hal.archives-ouvertes.fr/hal-00002166/en/.

[5] D. TRAORE, J.-L. ROCH, N. MAILLARD, T. GAUTIER, J. BERNARD. *Deque-free work-optimal parallel STL algorithms*, in "EUROPAR 2008", Las Palmas, Spain, Springer-Verlag, Aug 2008, http://www-id.imag.fr/Laboratoire/Membres/Roch_Jean-Louis/perso_html/papers/2008-europar-adaptSTL.pdf.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[6] S. PERARNAU. *Environnements pour l'analyse expérimentale d'applications de calcul haute performance*, Université de Grenoble, December 2011, http://hal.inria.fr/tel-00650047.

[7] B. PETIT. *Téléprésence, immersion et interactions pour le reconstruction 3D temps-réel*, Université de Grenoble, February 2011, http://hal.inria.fr/tel-00584001/en.

[8] J.-N. QUINTIN. *Equilibrage de charge dynamique sur plates-formes hiérarchiques*, Université de Grenoble, December 2011.

Articles in International Peer-Reviewed Journal

[9] M. BOUGERET, P.-F. DUTOT, A. GOLDMAN, Y. NGOKO, D. TRYSTRAM. *Approximating the discrete resource sharing scheduling problem*, in "International Journal of Foundations of Computer Science", 2011, vol. 22, n° 3, DOI: 10.1142/s0129054111008271.

[10] M. CHAVENT, A. VANEL, A. TEK, B. LÉVY, S. ROBERT, B. RAFFIN, M. BAADEN. *GPU-accelerated atom and dynamic bond visualization using HyperBalls, a unified algorithm for balls, sticks and hyperboloids*, in "Journal of Computational Chemistry", October 2011, vol. 32, n° 13, p. 2924-2935.

[11] J. COHEN, D. CORDEIRO, D. TRYSTRAM, F. WAGNER. *Multi-organization scheduling approximation algorithms*, in "Concurrency and Computations: Practice and experience", dec 2011, vol. 23, n° 17, p. 2220–2234, http://dx.doi.org/10.1002/cpe.1752.

[12] P.-F. DUTOT, F. PASCUAL, K. RZADCA, D. TRYSTRAM. *Approximation Algorithms for the Multi-Organization Scheduling Problem*, in "IEEE Transactions on Parallel and Distributed Systems", 2011, vol. 99, PrePrints, http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.47.

[13] 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", 2011, doi: 10.1016/j.jpdc.2011.11.003.

[14] A. MAHJOUB, J. E. PECERO-SÁNCHEZ, D. TRYSTRAM. *Scheduling with uncertainties on new computing platforms*, in "Computational Optimization and Applications", 2011, vol. 48, n° 2, p. 369-398.

International Conferences with Proceedings

[15] X. BESSERON, T. GAUTIER. *Impact of over-decomposition on coordinated checkpoint/rollback protocol*, in "Workshop on Resiliency in High-Performance Computing, 17-th International European Conference On Parallel and Distributed Computing", Bordeaux, France, August 2011.

[16] M. BOUGERET, P.-F. DUTOT, K. JANSEN, C. ROBENEK, D. TRYSTRAM. *Scheduling jobs on heterogeneous platforms*, in "Proceedings of COCOON, the 17th Annual International Computing and Combinatorics", Dallas, USA, LNCS, Springer, August 2011, vol. 6842, p. 271-283, http://dl.acm.org/citation.cfm?id=2033119.

[17] M. S. BOUGUERRA, D. KONDO, D. TRYSTRAM. *On the Scheduling of Checkpoints on Desktop Grids*, in "11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2011)", May 2011, p. 305-313, http://mescal.imag.fr/membres/derrick.kondo/pubs/slim_ccgrid11.pdf.

[18] L.-C. CANON, A. ESSAFI, G. MOUNIÉ, D. TRYSTRAM. *A bi-objective scheduling algorithm for desktop grids with uncertain resource avalaibilities*, in "Proceedings of the 17th EuroPar international Conference", Bordeaux, France, LNCS, Springer, 2011, vol. 6853, p. 238-249, acceptance rate 29%.

[19] J. COHEN, D. CORDEIRO, D. TRYSTRAM, F. WAGNER. *Coordination Mechanisms for Selfish Multi-Organization Scheduling*, in "Proceedings of the 18th annual IEEE International Conference on High Performance Computing (HiPC)", Los Alamitos, CA, USA, IEEE Computer Society, December 2011, To appear.

[20] D. CORDEIRO, P.-F. DUTOT, G. MOUNIÉ, D. TRYSTRAM. *Tight Analysis of Relaxed Multi-Organization Scheduling Algorithms*, in "Proceedings of the 25th IEEE International Parallel & Distributed Processing

Symposium (IPDPS)", Anchorage, AL, USA, IEEE Computer Society, May 2011, p. 1177–1186, http://dx.doi.org/10.1109/IPDPS.2011.112.

[21] S. PERARNAU, M. TCHIBOUKDJIAN, G. HUARD. *Controlling Cache Utilization of Parallel Applications*, in "International Conference on Supercomputing (ICS)", April 2011.

[22] V. REHN-SONIGO, D. TRYSTRAM, F. WAGNER, H. XU, G. ZHANG. *Off-line scheduling of multi-Threaded request streams on a caching server*, in "Proceedings of the 25th IPDPS", Anchorage, Alaska, IEEE, 2011, acceptance rate 19%.

National Conferences with Proceeding

[23] S. PERARNAU, D. TRYSTRAM. *Ggen : génération aléatoire de graphes pour l'ordonnancement*, in "ROADEF", Saint Etienne, France, March 2011.

[24] B. PETIT, A. LETOUZEY, E. BOYER. *Flot de surface à partir d'indices visuels*, in "ORASIS - Congrès des jeunes chercheurs en vision par ordinateur", Praz-sur-Arly, France, INRIA Grenoble Rhône-Alpes, 2011, http://hal.inria.fr/inria-00595247/en.

Scientific Books (or Scientific Book chapters)

[25] J.-G. DUMAS, J.-L. ROCH, E. TANNIER, S. VARRETTE. *Foundations of Coding: Compression, Encryption, Error-Correction*, Springer, 2012.

Research Reports

[26] L.-C. CANON, A. ESSAFI, G. MOUNIÉ, D. TRYSTRAM. *A Bi-Objective Scheduling Algorithm for Desktop Grids with Uncertain Resource Availabilities*, LIG, Grenoble, France, 2011, n^{o} RR-LIG-014, http://rr.liglab.fr/research_report/RR-LIG-014.pdf.

[27] A. GOLDMAN, Y. NGOKO, D. TRYSTRAM. *Optimizing resource sharing on cooperative execution of algorithms*, LIG, Grenoble, France, 2011, n^{o} RR-LIG-021, http://rr.liglab.fr/research_report/RR-LIG-021.pdf.

[28] F. LE MENTEC, V. DANJEAN, T. GAUTIER. *X-Kaapi C programming interface*, INRIA, December 2011, n^{o} RT-0417, http://hal.inria.fr/hal-00647474/en/.

[29] F. LE MENTEC, T. GAUTIER, V. DANJEAN. *The X-Kaapi's Application Programming Interface. Part I: Data Flow Programming*, INRIA, December 2011, n^{o} RT-0418, http://hal.inria.fr/hal-00648245/en/.

[30] B. PETIT, A. LETOUZEY, E. BOYER. *Surface Flow from Visual Cues*, INRIA, May 2011, n^{o} RR-7619, http://hal.inria.fr/inria-00593206/en.

[31] L. L. PILLA, C. POUSA RIBEIRO, D. CORDEIRO, A. BHATELE, P. O. A. NAVAUX, J.-F. MEHAUT, L. V. KALÉ. *Improving Parallel System Performance with a NUMA-aware Load Balancer*, INRIA-Illinois Joint Laboratory on Petascale Computing, July 2011, n^{o} TR-JLPC-11-02, http://hdl.handle.net/2142/25911.

[32] J.-N. QUINTIN, F. WAGNER. *WSCOM: Online task scheduling with data transfers*, INRIA, November 2011, n^{o} RR-7792, http://hal.inria.fr/hal-00639922/en.

Other Publications

[33] B. RAFFIN. *GPU Computing: Does it Worth the Effort?*, 2011, International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE), Invited Speaker.

[34] B. RAFFIN. *High Perfomance Interactive Computing*, 2011, Latin American Conference on High Performance Computing (CLCAR), Invited Speaker.