INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team moais

ProgramMming and scheduling design fOr Applications in Interactive Simulation

Grenoble - Rhône-Alpes

Theme : Distributed and High Performance Computing

Activity Report

2010

Table of contents

1. Team

Research Scientists

Thierry Gautier [Research Scientist CR1]
Bruno Raffin [Research Scientist CR1, HdR]

Faculty Members

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

Technical Staff

Eric Amat [2010-2012. INRIA Grant (ADT VGATE)]
Fabien Le Mentec [2009, Engineer ADT Kaapi]
Fabrice Schuler [2010, Engineer Minalogic contract SHIVA]
Antoine Vanel [2008, Engineer ANR FVNano]

PhD Students

Sami Achour [2006, co-tutelle ESST Tunis, Tunisia (Mohamed Jemni)]
Xavier Besseron [2006, MRNT scholarship]
Marin Bougeret [2007, BDI CNRS / DGA scholarship]
Mohamed-Slim Bouguerra [2008, INRIA Cordi]
Daniel Cordeiro [2007, Alban scholarship]
Marie Durand [2010-2013. Funded by ANR project REPDYN]
Adel Essafi [2006, co-tutelle ESST Tunis, Tunisia (Amine Mahjoub)]
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]
Yanik N'Goko [2006, co-tutelle Univ. Yaoundé, Cameroon, SARIMA scholarship]
Swann Perarnau [2008, MRNT scholarship]
Benjamin Petit [2007, common to PERCEPTION and MOAIS]
Jean-Noel Quintin [2008, Minalogic CILOE contract]
Thomas Roche [2007, common to UJF-Institut Fourier and MOAIS, CIFRE C-S scholarship]
Marc Tchiboukdjian [2007, BDI CNRS / CEA DAM scholarship]

Post-Doctoral Fellows

Ingo Assenmacher [18 months]
Louis-Claude Canon [2010, ATER Ensimag, Grenoble-INP]

Visiting Scientists

Denise Stringhini [Universidade Presbiteriano Mackenzie, Sao Paulo, Brasil, 4 months]
Guochuan Zhang [Zhejiang University, Hangzhou, China, 1 month]

Administrative Assistant

Ahlem Zammit-Boubaker [INRIA Administrative Assistant, 50%]

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 has received a *Google research award* 2010 for his work on multi-criteria optimization in distributed systems.
- The INRIA project teams Moais, Perception, Iparla and the University of Orléans, collaborated to set-up a tele-presence experiment across Bordeaux, Grenoble and Orléans. Participants from Bordeaux and Grenoble were present in the shared virtual space through their 3D model computed in real time from multiple surrounding cameras. This experiment shows that our approach provides a strong feel of presence and immersion favoring distant collaboration. A video was produced (http://www.dailymotion.com/swf/video/xe03fq_daliafinalacmm2010_tech) and a short paper published at ACMM 2010.

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: 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, 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.

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
- AMS: 68N19
- License: CeCILL
- Type of human computer interaction: console
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++ gnu++98 (GNU dialect of ANSI C++98)),Java (bindings to C++), script (perl 5.8 and 5.10) * Fortran (bindings to C++)
- Documentation: with doxygen

5.2. OAR

Participant: Grégory Mounié [correspondant MOAIS].

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.

- License: LGPL

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.2 Physics, J.3 LIFE AND MEDICAL SCIENCES
- License: GPL
- License: LGPL
- Type of human computer interaction: console, opengl, qt
- OS/Middelware: linux, windows, mac
- Required library or software: Qt - GPL - www.qtsoftware.comGLEW - BSD/MIT - glew.sourceforge.netTinyxml - zlib - tinyxml http://www.grinninglizard.com/tinyxml
- Programming language: C/C++
- Documentation: doxygen

5.4. FlowVR

Participant: Bruno Raffin [correspondant].

The goal of the FlowVR library is to provide 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.
See also the web page http://flowvr.sf.net.

- Version: 1.7.0
- ACM: D.1.3 Concurrent Programming
- APP: IDDN.FR.001.400021.000.S.P.2008.000.10000
- License: GPL
- License: LGPL
- Type of human computer interaction: Console
- OS/Middelware: Linux et Mac OS X
- Required library or software: Tinyxml (GPL - intégré dans FlowVR)
- Programming language: C++
- Documentation: Doxygen

5.5. TakTuk - Adaptive large scale remote execution deployment

Participant: Guillaume Huard [corespondant].

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.

5.6. Triva - Three dimension-Interactive Visualization Analysis

Participant: Guillaume Huard [corespondant].

Parallel applications use grid infrastructures to obtain more performance during their execution. The successful result of these executions depends directly on a performance analysis that takes into account the grid characteristics, such as the network topology and resources location. Triva is a software analysis tool that implements a novel technique to visualize the behavior of parallel applications. The proposed technique explores 3D graphics in order to show the application behavior together with a description of the resources, highlighting communication patterns, the network topology and a visual representation of a logical organization of the resources. We have used a real grid infrastructure in order to execute and trace applications composed of thousands of processes. http://triva.gforge.inria.fr under GNU GPL licence.

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

Participants: Swann Perarnau [corespondant], Guillaume Huard.

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.

6. New Results

6.1. FlowVR

FlowVR was extended to run applications on grid infrastructures. Important development efforts were focused on interaction and rendering. The rendering engine has been rewritten extending its capabilities to support multiple FBOs. Interaction management is now clearly separated from rendering and we added the capability to easily create 2D overlays for menus based on CeGUI.

6.2. Cache-Oblivious Mesh Layout

One important bottleneck when visualizing large data sets is the data transfer between processor and memory. Cache-aware (CA) and cache-oblivious (CO) algorithms take into consideration the memory hierarchy to design cache efficient algorithms. CO approaches have the advantage to adapt to unknown and varying memory hierarchies. Recent CA and CO algorithms developed for 3D mesh layouts significantly improve performance of previous approaches, but they lack of theoretical performance guarantees. We developped a algorithm, called FastCOL, to compute a CO layout for unstructured but well shaped meshes. We proved that a coherent traversal of a N -size mesh in dimension d induces less than cache-misses where B and M are the block size and the cache size, respectively. Experiments show that our layout computation is faster and significantly less memory consuming than the best known CO algorithm (OpenCCL). The FastCOL performance is comparable to the OpenCCL algorithm for classica! l visualization algorithm access patterns, or better when the BSP tree produced while computing the layout is used as an acceleration data structure adjusted to the layout. We also show that cache oblivious approaches lead to significant performance increases on recent GPU architectures. We also combined cache oblivious layouts with work stealing, relying on a cooperative cache access policy to decrease the number of cache misses on shared caches. The CO algorithm was published in IEEE TVCG, 2010. The extension for multi-core processors was published at EGPGV 2010 and HPCC 2010.

6.3. Kaapi

New version of Kaapi, called X-Kaapi, has been released. The kernel is written in C for hypothetical required from embedded system. The remainder part is pure C++. Several APIs have been developed : C++ tasks API to create tasks with dependencies on same reference to memory region; STL like parallel algorithms (KASTL) and novel Adaptive Application Interface that allows to write application that directly interact with the work stealing scheduler in order to avoid task prior to their execution.

6.4. Adaptive and hierarchical work-stealing

Regarding work-stealing and greedy scheduling, the new analysis developed last year has been further extended to drastically improve upper bound guarantees. This result has been published to ISAAC 2010. Considering large distributed infrastructures (lightweight grids or cluster of clusters), several workstealing variants have been designed to reduce communications by taking topology into account (Europar 2010). In order to deal with unknown or dynamic topologies, we are currently working on extensions in the online context by exploiting knowledge of task dependencies.

6.5. Output-Sensitive Decoding for Redundant Residue Systems

In order to provide fault tolerant computations on clouds or global computing platforms, we proposed a unified point of view on redundant residue systems for both polynomial and integer computations (published in ISSAC 2010). In contrast to list decoding for error-correcting codes, our algorithm is oblivious to the parameter of the codes. Providing a list of possible consistent results, it extends algorithm-based fault tolerance to online parallel computations following the evaluation/interpolation scheme while considering early termination.

7. Contracts and Grants with Industry

7.1. Contracts with Industry

+ **BDI funded by C-S (2007-2010).** The first objective is to design efficient extensions and integration of the cipher CS (initially designed by C-S group) in order to exploit parallelism (based on parallel mode of operations). The second one concerns the design of scalable protocols to provide confidence and security in a large scale infrastructure. Fund 1 PhD.
+ **BDI co-funded by CNRS and CEA/DIF (2007-2010).** This PhD is focused on cache and processor oblivious approaches applied to high performance visualization. The goal is to study rendering algorithms (mainly volume rendering and isosurface extraction) for large meshes (irregular and adaptive) that are proven efficient without requiring the mesh layout or the algorithm to actually know the memory hierarchy of the target architecture or the number of processor available. Fund 1 PhD.
+ **Contract with EDF (2010-2013).** High performance scientific visualization. Fund 1 postdoc and 1 PhD.

8. Other Grants and Activities

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 Quiintin.
- *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 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).
- *GRIMDEV*, 2009-2010, ADT INRIA. This project brings engineering support for maintaining, operating and developing new experimentson the Grimage platform: Hervé Mathieu, INRIA SED, part time, Nicolas Turro, INRIA SED, part time and IJD Thomas Dupeux, INRIA, full time. The partners are the EPI MOAIS and PERCEPTION.

8.2. National Initiatives

- *Ggen*, 09-10, funded by the GdR RO to study the impact of random graph generation methods on the quality of scheduling simulations. http://ggen.ligforge.imag.fr/. Partners: projects MOAIS and MESCAL (INRIA Rhône-Alpes).
- *FVNANO*, 07-10, ANR-CIS: the project focuses on developing a framework for the interactive manipulation of nano objects. FlowVR is the core middleware used to build interactive applications coupling nano simulations, visualization and haptic force feedback. Partners : projects MOAIS (INRIA Rhône-Alpes), the CEA/DIF, the Laboratoire de Biochimie Théorique (LBT) and the LIFO (Université d'Orléans).
- *Vulcain*, 07-10, ANR Programme Génie Civil et Urbain: the project focuses on studying industrial structure reliability under dynamic constrinats (explosions, impacts). The role of the INRIA projects MOAIS and EVASION in this project is to provide a parallel framework based on SOFA for fast dynamic simulations. Partners: projects EVASION and MOAIS (INRIA Rhône-Alpes), 3S-R, IPSC-ELSA, CEG-DGA, LEES, LaM, INERIS, IRSN, CEA, SME Environnement, Phimeca, Bull.
- *DALIA*, 06-10, ARA Masse de Données: the project deals with multi-site interactive applications involving from handheld devices up to large multi-camera and multi-projector platforms. Partners : projects PERCEPTION, MOAIS (INRIA Rhône-Alpes), project I-parla (Bordeaux, INRIA Futurs) and the LIFO (Université d'Orléans).

- *ANR REPDYN (2010-2012).* Coordinator for the INRIA Rhône-Alpes. High performance structure and fluid computing. Partners: INRIA Rhône-Alpes, CEA, ONERA, EDF, LaMSID lab from CNRS and LaMCoS lab from INSA Lyon.
- *ANR/JS PETAFLOW (2010-2012).* France/Japon international program. Coordinator for the INRIA Rhône-Alpes. 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.
- *PEPS MIMPAC.* 2009-2010. Parallel Computational Chemistry. Coordinator: V. Louvet. Partners: LEM2C (Région Parisienne, CEA (Paris), Institut Camille Jordan (Lyon), Lab. J. Dieudonné (Nice).
- *PEPS LINBOX.* 2010-2011. High Performance Library for Computer Algebra . Coordinator: C. Pernet. Partners: LIP (Lyon), LJK (Grenoble), LIRMM (Montpellier).

8.3. European Initiatives

- *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. Brazil

- We have a long term and strong collaboration with the Universities of Rio Grande do Sul, Brazil, and in particular with UFRGS, Porto Alegre. This collaboration is funded in 2010 by 2 different grants:
  - INRIA/Cnpq (2008-2010).
  - INRIA Diode-A (2006-2011).
- 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.
- USP-COFECUB project with the universities of Sao Paulo and Fortaleza, Brazil, focused on the impact of communications on parallel task scheduling. One year funding.

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.

8.5.4. MPSoC

ST Microelectronics provided us a STM8010 machine for experimenting parallel adaptive algorithms on MPSoC.

9. Dissemination

9.1. Animation of the scientific community

- 2010 Chair / Symposium co-chair
  - 12th Virtual Reality International Conference, April 7-9, 2010, Laval, France
  - 5th ACM International workshop PASCO 2010, Parallel Algebraic and Symbolic Computing, July 22-24, 2010, Grenoble, France
  - chair of "New trends in scheduling theory", 12-17 september 2010, Frejus, France
- 2010 Program committee member
  - Program Committee member of JVRC 2010 (Joint Virtual Reality Conference of EuroVR - EGVE - VEC).
  - Program Committee of PDPS 2010 (21st IEEE International Parallel & Distributed Processing Symposium) april, Atlanta, USA
  - Program Committee of Hicomb (9th IEEE International Workshop on High Performance Computational Biology) april 2010, Atlanta
  - Program Committee of HCW'2010 (19th IEEE Heterogeneous Computing Workshop) april 2010, Atlanta

  - Program Committee of LSAP 2010 (Workshop on Large-Scale System and Application Performance) june 2010, Chicago, USA
  - Program Committee of OPTIM'10 (Workshop on Optimization Issues in Energy Efficient Distributed Systems) june 2010, Caen, France
  - Program Committee of ISPDC 2010 (9th International Symposium on Parallel and Distributed Computing) july 2010, Istanbul, Turkey
  - Program Committee of WAOA 2010 (8th Workshop on Approximation and Online Algorithms) Liverpool, UK
  - Program Committee of LaSCoG 2010(6th Workshop on Large Scale Computations on Grids) october 2010, Visla, Poland
  - Program Committee of CARI 2010 (Colloque Africain sur la Recherche en Informatique) oct. 2010, Yamassoukro, Cote d'Invoire
  - Program Committee of PASCO 2010 (Parallel and Distributed Programming), Grenoble, France

- 2011 Program committee member
  - RenPar 2011 (the 20ièmes Rencontres Francophones du Parallèlisme), 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 Optimization Issues in Energy Efficient Distributed Systems) july 4-8, 2011, Istanbul, Turkey
  - ISPDC 2011 (10th International 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
  - PPAM 2011, september 10-14, 2011, Torun, Poland
  - New perspectives in scheduling theory, october 9-14, 2011, Hangzhou, China
  - EGPGV 2011 (Eurographics Symposium on Parallel Rendering and Visualization)
  - IEEE VR 2011 (IEEE Conference on Virtual Reality)
  - SVR 2011 (Symposium on Virtual and Augmented Reality), Brazil
  - ISVC 2011 (International Symposium on Visual Computing)
  - WEHA 2011 (Workshop on Exploitation of Hardware Accelerators)
  - PAPP 2011 (Seventh International Workshop on applications of declArative and object-oriented Parallel Programming)
  - HSNCE (First Workshop on High Speed Network and Computing Environments for Scientific Applications in Conjunction with SAINT2010), Korea

- 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

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] X. BESSERON. *Tolérance aux fautes et reconfiguration dynamique pour les applications distribuées à grande échelle*, Université de Grenoble, Apr 2010, http://hal.inria.fr/tel-00486939/en.

[7] M. BOUGERET. *Systèmes interactifs pour la résolution de problèmes complexes*, Université de Grenoble, December 2010.

[8] E. HERMANN. *Interactive Physical Simulation on Multi-COre and Multi-GPU Architectures*, Université de Grenoble, June 2010.

[9] Y. NGOKO. *L'approche du portfolio d'algorithmes pour la constructions des algorithmes robustes et adaptatifs*, Université de Grenoble, July 2010.

[10] T. ROCHE. *Dimensionnement et intégration d'un chiffre symétrique dans le contexte d'un système d'information distribué de grande taille.*, Université Joseph-Fourier - Grenoble I, Jan 2010, http://hal.inria.fr/tel-00452399/en.

[11] M. TCHIBOUKDJIAN. *Algorithmes parallèles efficaces en cache Applications à la visualisation scientifique*, Université de Grenoble, December 2010.

Articles in International Peer-Reviewed Journal

[12] J. Allard, J.-D. Lesage, B. Raffin. *Modularity for Large Virtual Reality Applications*, in "Presence: Teleoperators and Virtual Environments", April 2010, vol. 19, n^{o} 2, p. 142-162.

[13] 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", 2010.

[14] E. Boyer, B. Petit, B. Raffin. *The Virtualization Gate Project*, in "ERCIM NEWS", 2010, http://perception.inrialpes.fr/Publications/2010/BPR10.

[15] F. Diedrich, K. Jansen, F. Pascual, D. Trystram. *Approximation Algorithms for Scheduling with Reservations*, in "Algorithmica", 2010, vol. 58, n^{o} 2, p. 391-404.

[16] A. Mahjoub, J. P. Sanchez, D. Trystram. *Scheduling with uncertainties on new computing platforms*, in "Computational Optimization and Applications", 2010, DOI: 10.1007/s10589-009-9311-0.

[17] Q. Meunier, F. Pétrot, J.-L. Roch. *Hardware/software support for adaptive work-stealing in on-chip multiprocessor*, in "Journal of Systems Architecture", 2010, vol. 56, n^{o} 8, p. 392–406, http://dx.doi.org/10.1016/j.sysarc.2010.06.007.

[18] C. Pernet, W. Stein. *Fast computation of Hermite normal forms of random integer matrices*, in "Journal of Number Theory", jul 2010, vol. 130, n^{o} 7, p. 1675–1683, http://dx.doi.org/doi:10.1016/j.jnt.2010.01.017.

[19] B. Petit, J.-D. Lesage, M. Clément, J. Allard, J.-S. Franco, B. Raffin, E. Boyer, F. Faure. *Multicamera Real-Time 3D Modeling for Telepresence and Remote Collaboration*, in "International journal of digital multimedia broadcasting", 2010, Article ID 247108, 12 pages, http://hal.inria.fr/inria-00436467/en.

[20] L. M. Schnorr, G. Huard, P. O. A. Navaux. *Triva: Interactive 3D visualization for performance analysis of parallel applications*, in "Future Generation Computer Systems", 2010, vol. 26, n^{o} 3, p. 348 - 358 [*DOI :* DOI: 10.1016/j.future.2009.10.006], http://www.sciencedirect.com/science/article/B6V06-4XFY0B3-6/2/73a0ee2437a5affe13ab572dba4dd262.

[21] M. Tchiboukdjian, V. Danjean, B. Raffin. *Binary Mesh Partitioning for Cache-Efficient Visualization*, in "IEEE Transaction on Visualization and Computer Graphics (TVCG)", 2010, vol. 16, n^{o} 5, p. 815–828, http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.19.

International Peer-Reviewed Conference/Proceedings

[22] M. Albrecht, C. Pernet. *Efficient Decomposition of Dense Matrices over GF(2)*, in "Proceedings of the Workshop on Tools for Cryptanalysis", jun 2010, arXiv:1006.1744 [cs.MS].

[23] K.-I. Baba, J. Cisonni, Y. Ebara, P. Gonzales, X. Grandchamp, T. Kawamura, K. Koyamada, K. Nozaki, H. Ohsaki, X. Pelorson, P. Primet, B. Raffin, E. Sakane, N. Sakamoto, S. Shimojo, A. V. Hirtum. *Petaflow: a project towards information and communication technologies in society*, in "First Workshop on High Speed Network and Computing Environments for Scientific Applications (in conjonction with SAINT 2010)", Seoul, South Corea, July 2010.

[24] F. Blachot, G. Huard, J. Pecero Sanchez, E. Saule, D. Trystram. *Scheduling Instructions on Hierarchical Machines*, in "11th International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2010)", Atlanta, United-States, 2010.

[25] M. Bougeret, P.-F. Dutot, K. Jansen, C. Otte, D. Trystram. *A fast $\frac{5}{2}$-approximation algorithm for hierarchical scheduling*, in "Proceedings of the 16th International EuroPar Conference", Ischia, Italy, Lecture Notes in Computer Science, Springer, 2010, vol. 6272, p. 157-167.

[26] M. Bougeret, P.-F. Dutot, K. Jansen, C. Otte, D. Trystram. *Approximating the non-contiguous Multiple Organization Packing Problem*, in "Proceedings of the 6th International Conference on Theoretical Computer Science (TCS)", IFIP Advances in Information and Communication Technology, 2010, vol. 323, p. 316-327.

[27] M. Bougeret, P.-F. Dutot, K. Jansen, C. Otte, D. Trystram. *Approximation Algorithms for Multiple Strip Packing*, in "Approximation and Online Algorithms", Lecture Notes in Computer Science, 2010, vol. 5893, p. 37-48.

[28] M.-S. Bouguerra, T. Gautier, D. Trystram, J.-M. Vincent. *A Flexible Checkpoint/Restart Model in Distributed Systems*, in "Parallel Processing and Applied Mathematics", Lecture Notes in Computer Science, 2010, vol. 6067, p. 206-215.

[29] A. Bourki, G. Chaslot, M. Coulm, V. Danjean, H. Doghmen, T. Hérault, J.-B. Hoock, A. Rimmel, F. Teytaud, O. Teytaud, P. Vayssière, Z. Yu. *Scalability and Parallelization of Monte-Carlo Tree Search*, in "The International Conference on Computers and Games 2010", Japon Kanazawa, 2010, http://hal.inria.fr/inria-00512854/en.

[30] J. Cohen, D. Cordeiro, D. Trystram, F. Wagner. *Analysis of Multi-Organization Scheduling Algorithms*, in "Parallel Processing, 16th International Euro-Par Conference", Italie Ischia, P. D'Ambra, M. R. Guarracino, D. Talia (editors), Lecture Notes in Computer Science, Springer, 2010, vol. 6272, p. 367–379, http://hal.inria.fr/inria-00536510/en.

[31] D. Cordeiro, G. Mounié, S. Pérarnau, D. Trystram, J.-M. Vincent, F. Wagner. *Random graph generation for scheduling simulations*, in "3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010)", Espagne Malaga, ICST, Mar 2010, 10, http://hal.inria.fr/hal-00471255/en.

[32] J.-G. Dumas, T. Gautier, C. Pernet, D. B. Saunders. *LinBox founding scope allocation, parallel building blocks, and separate compilation*, in "The Third International Congress on Mathematical Software", Japon Kobe, K. Fukuda, J. van der Hoeven, M. Joswig (editors), Springer, Sep 2010, vol. 6327, 6, http://hal.inria.fr/hal-00506599/en.

[33] J.-G. Dumas, T. Gautier, J.-L. Roch. *Generic design of Chinese remaindering schemes*, in "International Symposium on Parallel Symbolic Computation", France Grenoble, M. Moreno Maza, J.-L. Roch (editors), Association for Computing Machinery, Jul 2010, p. 26-34 [*DOI :* 10.1145/1837210.1837218], http://hal.inria.fr/hal-00449864/en.

[34] E. Hermann, B. Raffin, F. Faure, T. Gautier, J. Allard. *Multi-GPU and Multi-CPU Parallelization for Interactive Physics Simulations*, in "Europar 2010", Italie Ischia-Naples, Sep 2010, http://hal.inria.fr/inria-00502448/en.

[35] L. Jacquin, V. Roca, J.-L. Roch, M. Al Ali. *Parallel arithmetic encryption for high-bandwidth communications on multicore/GPGPU platforms*, in "Parallel Symbolic Computation'10 (PASCO'10)", Grenoble, France, ACM, July 2010, http://portal.acm.org/ft_gateway.cfm?id=1837223&type=pdf&coll=portal&dl=ACM&CFID=97976165&CFTOKEN=93533625.

[36] M. KHONJI, C. PERNET, J.-L. ROCH, T. ROCHE, T. STALINSKI. *Output-sensitive decoding for redundant residue systems*, in "ISSAC'10: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation", New York, NY, USA, ACM, 2010, p. 265–272, http://doi.acm.org/10.1145/1837934.1837985.

[37] L. MASKO, M. TUDRUJ, G. MOUNIÉ, D. TRYSTRAM. *Comparison of Program Task Scheduling Algorithms for Dynamic SMP Clusters with Communication on the Fly*, in "Parallel Processing and Applied Mathematics", Lecture Notes in Computer Science, 2010, vol. 6068, p. 31-34.

[38] B. PETIT, T. DUPEUX, B. BOSSAVIT, J. LEGAUX, B. RAFFIN, E. MELIN, J.-S. FRANCO, I. ASSENMACHER, E. BOYER. *A 3D Data Intensive Tele-immersive Grid*, in "ACM Multimedia (ACMM'10)", Firenze, Italia, ACM, October 2010.

[39] S. PÉRARNAU, G. HUARD. *KRASH: Reproducible CPU Load Generation on Many-Core Machines*, in "IEEE International Parallel and Distributed Processing Symposium (IPDPS)", IEEE, 2010.

[40] S. PÉRARNAU, G. HUARD. *KRASH: Reproducible CPU Load Generation on Many-Core Machines*, in "Poster session in proceedings of the 15th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming", Bangalore India, PPoPP'10, ACM, 2010, p. 327–328.

[41] J.-N. QUINTIN, F. WAGNER. *Hierarchical work-stealing*, in "Proceedings of the 16th international Euro-Par conference on Parallel processing: Part I", Berlin, Heidelberg, EuroPar'10, Springer-Verlag, 2010, p. 217–229, http://portal.acm.org/citation.cfm?id=1887695.1887719.

[42] M. TCHIBOUKDJIAN, V. DANJEAN, T. GAUTIER, F. LE MENTEC, B. RAFFIN. *A Work Stealing Algorithm for Parallel Loops on Shared Cache Multicores*, in "4th Workshop on Highly Parallel Processing on a Chip (HPPC)", Napoli, Italia, August 2010.

[43] M. TCHIBOUKDJIAN, V. DANJEAN, B. RAFFIN. *Cache-Efficient Parallel Isosurface Extraction*, in "Eurographics 2010 Symposium on Parallel Graphics and Visualization (EGPGV'10)", Norrkping, Sweden, Eurographics, May 2010.

[44] M. TCHIBOUKDJIAN, V. DANJEAN, B. RAFFIN. *Cache-Efficient Parallel Isosurface Extraction for Shared Cache Multicores*, in "Eurographics Symposium on Parallel Graphics and Visualization 2010", May 2010, accepted on March, 11th 2010.

[45] M. TCHIBOUKDJIAN, N. GAST, D. TRYSTRAM, J.-L. ROCH, J. BERNARD. *A Tighter Analysis of Work Stealing*, in "The 21st International Symposium on Algorithms and Computation (ISAAC)", 2010, http://moais.imag.fr/membres/marc.tchiboukdjian/pub/isaac10.pdf.

National Peer-Reviewed Conference/Proceedings

[46] M. CHAVENT, A. VANEL, A. TECK, B. LEVY, B. RAFFIN, M. BAADEN. *A Rendering Method for Small Molecules up to Macromolecular Systems: HyperBalls Accelerated by Graphics Processors*, in "Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM'10)", Montpellier, France, September 2010.

Books or Proceedings Editing

[47] M. BENDER, J. BLAZEWICZ, E. PESCH, D. TRYSTRAM, G. ZHANG (editors). *Journal of Scheduling: Guest editorial for the special issue 'New challenges in scheduling theory 2008'*, Springer Verlag, 2010, vol. 13, p. 451-452.

[48] M. Moreno Maza, J.-L. Roch (editors). *PASCO '10: Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*, ACM, New York, NY, USA, July 2010, http://portal.acm.org/toc.cfm?id=1837210.

Research Reports

[49] M. Tchiboukdjian, V. Danjean, T. Gautier, F. Le Mentec, B. Raffin. *Adaptive Algorithms for Shared Cache on Multicore*, INRIA, April 2010, n^o RR-7256, http://hal.inria.fr/inria-00473617/PDF/RR-7256.pdf.

[50] M. Tchiboukdjian, D. Trystram, J.-L. Roch, J. Bernard. *List Scheduling: The Price of Distribution*, INRIA, Jan 2010, n^o RR-7208, http://hal.inria.fr/inria-00458133/en.

Other Publications

[51] L. Jacquin, V. Roca, J.-L. Roch, M. Al Ali. *Parallel arithmetic encryption for high-bandwidth communications on multicore/GPGPU platforms.*, 2010, Report Hal, http://hal.inria.fr/hal-00493044/en.

[52] B. Petit, T. Dupeux, B. Bossavit, J. Legaux, B. Raffin, E. Melin, J.-S. Franco, I. Assenmacher, E. Boyer. *A 3D Data Intensive Tele-immersive Grid*, 2010, Report Hal, http://hal.inria.fr/hal-00514549/en.