INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team moais

Multi-programmation et Ordonnancement pour les Applications Interactives de Simulation

Grenoble - Rhône-Alpes

THEME NUM

Activity Report

2007

Table of contents

1. Team

The MOAIS project-team is a supported by the INRIA and LIG lab (UMR 5217 - CNRS, INPG, UJF).

Head of project-team
Jean-Louis Roch [Assistant Professor, INPG]

Administrative staff
Marion Ponsot [INRIA Administrative Assistant, 30%]

INRIA Staff
Thierry Gautier [Research Associate CR1]
Bruno Raffin [Research Associate CR1]

INPG Staff
Grégory Mounié [Assistant Professor]
Denis Trystram [Professor, HdR]
Frédéric Wagner [Assistant Professor]

UJF Staff
Guillaume Huard [Assistant Professor]
Vincent Danjean [Assistant Professor]

UPMF Staff
Pierre-François Dutot [Assistant Professor]

Invited Scientist
Alfredo Goldman [USP Sao Paulo Brasi, 4 months]
Klaus Jansen [U. Kiel Germany, 2 weeks]
Nicolas Maillard [UFRGS Porto Alegre, 1 month]
Andreï Tchernyk [CICESE, Ensenada, Mexico, 2 weeks]
Marek Tudruj [Polish Academy of Sciences, Warsaw, 1 week]
Lukasz Masko [Polish Academy of Sciences, Warsaw, 1 week]

Postdoc
Fanny Pascual [1 year]

Engineers
Serge Guelton [1 year]
Liyun He [1 year]
Fabrice Salpetrier [11 months]

PhD Students
Sami Achour [2006, co-tutelle ESST Tunis, Tunisia (Mohamed Jemni), EGIDE scholarship]
Julien Bernard [2005, BDI CNRS / ST Microelectronics scholarship]
Xavier Besseron [2006, MRNT scholarship]
Marin Bougeret [2007, BDI CNRS / DGA scholarship]
Daniel Cordeiro [2007, Alban scholarship]
Florian Diedrich [2006, co-tutelle U. Kiel Germany (Klaus Jansen), DAAD scholarship]
Adel Essafi [2006, co-tutelle ESST Tunis, Tunisia (Mohamed Jemni), EGIDE scholarship]
Everton Hermann [2006, INRIA Cordi]
Jean-Denis Lesage [2006, MRNT scholarship]
Clément Ménier [2003, Normalien, common to PERCEPTION and MOAIS]
Feryal-Kamila Moulai [2003, LIPS INRIA-BULL contract, INRIA scholarship]
Yanik N'Goko [2006, co-tutelle Univ. Yaoundé, Cameroon, SARIMA scholarship]
Jonathan Pecero-Sanchez [2003, SFERE CONACYT scholarship]
Benjamin Petit [2007, common to PERCEPTION and MOAIS]
Laurent Pigeon [2003, CIFRE IFP scholarship]
Thomas Roche [2007, common to UJF-Institut Fourier and MOAIS, CIFRE C-S scholarship]

Krysztof Rzadca [2004, co-tutelle INPG – PJIT Warsaw, Poland (Franciszek Seredynski), French embassy scholarship]
Erik Saule [2005, MRNT scholarship]
Lucas Schnorr [2007, co-tutelle INPG – UFRGS Porto Alegre, Brasil (Philippe Navaux), CAPES COFECUB scholarship]
Marc Tchiboukdjian [2007, BDI CNRS / CEA DAM scholarship]
Daouda Traore [2005, Egide France-Mali scholarship]
Gérald Vaisman [2006, DCN contract]
Sébastien Varette [2004, co-tutelle INPG – U. Luxembourg (Franck Leprévost), Luxembourg scholarship]
Haifeng Xu [2007, co-tutelle INPG – Zhejiang University, Hangzhou, China (Guochuan Zhang)]

2. Overall Objectives

2.1. Introduction

MOAIS project-team is dedicated to the parallel programming. In particluar we focus on high-performance interactive computing where performance is a matter of resources. Beyond the optimization of the application itself, the effective exploitation of both a large number of resources (computation, input and output units) and interaction with external components (user, expert or another application) is expected to enhance the performance. Generally, performance corresponds to a multi-criteria objective, for instance associating precision, fluidity and reactivity in interactive simulations.

Ideally, to achieve portability, the application should be independent to the platform and should support any configuration: adaptation to the platform is then managed by scheduling. Thus, fundamental researches undertaken in the MOAIS project are focused on this scheduling problem to manage the distribution of the application on the architecture. 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. Here, 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.

Research axes of MOAIS are directed towards:

- **Scheduling**: To formalize and study the related scheduling problem, the critical points are: the modeling of an adaptive application; the formalization of the multi-criterion objective; the design of scalable scheduling algorithms.
- **Adaptive parallel and distributed algorithms design**: To design and analyze algorithms that may adapt their execution under the control of the scheduling, the critical point is that algorithms are parallel and distributed; then, adaptation should be performed locally while ensuring the coherency of results.

- **Design and implementation of programming interfaces for coordination**. 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.
- **Interactivity.** To improve interactivity, the critical point is scalability. The number of resources (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 theoretical models;
- implementing and evaluating those algorithms with our main softwares: Kaapi for fine grain scheduling and FlowVR for coarse-grain scheduling;
- 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.

2.2. Highlights of the year

- The Moais, Perception and Evasion project-teams brought to the Siggraph 2007 Conférence, San Diego, a small scale of the GrImage platform. During 5 days we demonstrated a camera based approach for markerless 3D interactions. This demo was selected with 22 others amongst 75 submissions. The demo relies on the FlowVR middelware, developped by Moais, to reach a real-time performance (http://www.inrialpes.fr/grimage/#siggraph).
- After the special Jury prize for the III PLUGTEST contest during the Grid@Work event, the Moais project-team has won the first prize of the IV PLUGTEST contest among 7 teams during the 2007 edition of the Grid@Work event in Beijing. KAAPI software has been successfully deployed using TakTuk, developed by Moais, on almost all machines of the french national grid Grid5000. Applications have run on 3654 cores with sustained efficiencies.
- The book "Théorie des codes: compression, cryptage, correction" (Dunod Sciences-Sup publisher, 352 pages) co-written by Jean-Guillaume Dumas (LJK laboratory, Grenoble), Jean-Louis Roch (MOAIS team, INRIA Grenoble-Rhône-Alpes), Sébastien Varrete (MOAIS team, INRIA Grenoble-Rhône-Alpes and Université du Luxembourg) and Eric Tannier (HELIX team, INRIA Grenoble-Rhône-Alpes) has been published by Dunod editor in January 2007. This book, for Master and Engineer students in Computer Science and Applied Mathematics, introduces compression, cryptography and error-correcting algorithms and protocols

3. Scientific Foundations

3.1. Scheduling

Keywords: *load-sharing*, *mapping*, *scheduling*.

Participants: P.F. Dutot, T. Gautier, G. Huard, G. Mounié, J.-L. Roch, D. Trystram, F. 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 a date and a processor on which the tasks of a parallel program will 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 with 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 which 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 which start from an optimized off-line solution computed on an initial data set which is updated during the execution on the "perturbed" data (stability analysis).

Game Theory. Game Theory is a framework which 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 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. Using a generalized lock-free cactus stack execution mechanism, we have extended 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 sequential local execution of tasks enables to amortize the overhead of scheduling. The related distributed work-stealing scheduling algorithm has been implemented in **Kaapi**, the runtime library that supports the execution of Athapascan programs (Athapascan was studied and designed in the APACHE project).

3.2. Adaptive Parallel and Distributed Algorithms Design

Keywords: *adaptive*, *anytime*, *autonomic*, *complexity*, *hybrid.*

Participants: P.F. Dutot, T. Gautier, G. Huard, B. Raffin, J.-L. Roch, D. Trystram, F. 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, 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); an implementation of Athapascan, the parallel programming interface developed by the APACHE project, is available on top of Kaapi.

However, this adaptation is restrictive. More generally, the algorithm should adapt itself at runtime in order to improve performance by decreasing overheads induced by parallelism, namely 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 in order to find the good balance between parallelism and synchronizations. MOAIS project 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: sequential ones, to provide efficient execution on the local resource; parallel ones, that enables to extract parallelism from an idle resource in order to dynamically suit the degree of parallelism with respect to idle 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 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* when no program variable dependent on architecture parameters, such as the number or processors or their respective speeds, need to be tuned to minimize its runtime.

This year, this technique has been applied 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.

From 2007, this adaptation technique is now integrated in softwares that we are developping with external partners within contracts. Within the ANR SafeScale contract, we have developped on top of Athapascan a high level library STL-like library that provides adaptive parallel algorithms for distributed containers (such as transform, foreach and findif on vectors). A specific optimized C interface, dedicated to stream computation, has been developped within the Minalogic SCEPTRE contract for multi-processor system on chipsi (MPSoC) developped by STM; this interface is named AWS (Adaptive Work-Stealing).

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

3.3. Interactivity

Keywords: *high performance interactive computing*, *multimedia*, *virtual reality*.

Participants: V. Danjean, P.F. Dutot, T. Gautier, B. Raffin, J.-L. 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 tradeoff given the user wishes and the resources available. Issues include multicriterion 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

Keywords: *coordination languages*, *coupling*, *middleware*, *programming interface*.

Participants: V. Danjean, T. Gautier, B. Raffin, J.-L. Roch, F. 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 in order 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 in order to predict the cost of an execution for part of the application.

This work is based on the experience of the APACHE project and the collaborative research actions ARC SIMBIO and ARC COUPLAGE. 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 prealably computing at runtime a graph of tasks that represents the (future) calls to execute. Based on this technique, Athapascan, the language developed by the APACHE project, enables to write a single program for both the code to execute and the description of the future of the execution.

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 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 algorithm (scheduling algorithm 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: T. Gautier, B. Raffin, J.-L. 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. Clément Ménier, Ph.D. student co-advised by Edmond Boyer (PERCEPTION) and Bruno Raffin, defended its Ph.D. on this subject in 2007. Benjamin Petit started a Ph.D. in October 2007 also co-advised by Edmond Boyer (PERCEPTION) and Bruno Raffin. While Clément Menier mainly focused on making textured 3D models in real time using a PC cluster, Benjamin Petit will focus on using a multi-camera environments for increasing the interaction possibilities in virtual environments.
- 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 first target SMP machines with some of the computations delegated to GPUs.
- Distant collaborative work. We will 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 sites ; large scale interactive simulation using computing resources from the grid. For these experiments, we are collaborating with the LIFO and the LABRI.

4.2. Code Coupling and Grid Programming

Participants: T. Gautier, J.-L. Roch, V. Danjean, F. Wagner.

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

- **CAPE-OPEN applications.** A collaboration with IFP (Institut Français du Pétrole) has studied the design of high performance CAPE (Computer Aided Process Engineering) runtime for cluster architecture to exploit intrinsic parallelism of the CAPE industrial standard. CAPE-OPEN is a standard of interface of components in process engineering. The thesis of Laurent PIGEON proves the ability to exploit intrinsic parallelism of CAPE applications based on the standard CAPE-OPEN. We proposed a prototype that allows CAPE-OPEN compliant application to be executed on cluster. This work has been published and it shows good parallel efficiency if the application has enough parallelism. The resulting design allows any CAPE-OPEN compliant application to be automatically deployed on cluster without any development.

- **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. In order to test grid applications, we need to deployed and start programs on all used computers. This can become a difficult if the real topology involve 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 use 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.

4.3. Safe Distributed Computations

Participants: V. Danjean, T. Gautier, J.-L. 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. 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 confidency 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 ised both for a scalable checkpoint/restart protocol and for probabilistic detection of massive attacks.

Moreover, we study the scalability of security protocols on large scale infrastructure. During his thesis at Moais, Sébastien Varrette has developed and validated a scalable distributed authentication protocol based on LDAP which is operationally used since two years on the national grid Grid'5000. In order to open the grid usage to commercial applications from small-size companies (namely in the field of micro and nanotechnology 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 adpative parallelization of a block cipher (based on counter mode) has been evaluated. An FPGA implementation is in progress.

Conversely, large scale computing infrastructure are very useful to evaluate the robustness of cryptographic protocols (eg SHA-1 Collisison and PrimeGrid projects on BOINC). In collaboration with Institut Fourier (Roland Gillard), we use Kaapi and grid platforms to generate boxes with no quadratic relations.

4.4. Embedded Systems

Participants: J.-L. Roch, G. Huard, D. Trystram, V. 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 in order to save power and time. This challenge may be addressed by parallel computing coupled with performant scheduling. Also 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). This work has been achieved thanks to the support of two engineers employed on the SCEPTRE contract (Fabrice Salpetrier and, later Serge Guelton). A HD-video streaming application developped by STM is the target benchmark. We are also studying self-specialized implementation of work-stealing from an abstract description (from SPIRIT standard) of the MPSoC architecture.

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. FlowVR

Participants: C. Ménier, J-D. Lesage, B. 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 and scientific visualization. FlowVR enforces a modular programming that leverages software engineering issues while enabling high performance executions on distribued and parallel architectures.

The FlowVR software suite has today 3 main components:

- **FlowVR**: The core middleware library. FlowVR relies on the data-flow oriented programming approach that has been successfully used by other scientific visualization tools. Developing a FlowVR application is a two step process. First, modules are developed. Modules encapsulate a piece of code, imported from an existing application or developed from scratch. The code can be a multi-threaded or parallel, as FlowVR enables parallel code coupling. In a second step, modules are mapped on the target architecture and assembled into a network to define how data are exchanged. This netwok can make use of advanced features, from simple routing operations to complex message filtering or synchronization operations.
- **FlowVR Render**: A parallel rendering library. FlowVR Render proposes a framework to take advantage of the power offered by graphics clusters to drive display walls or immersive multi-projector environments like Caves. It relies on an original approach making an intensive use of hardware shaders. FlowVR Render comes with a port of the MPlayer Movie Player. This enables to play movies on a multi display environment. This application also a good example of the potential of FlowVR and FlowVR Render.

- **VTK FlowVR**: a VTK / FlowVR / FlowVR Render coupling library. VTK FlowVR enables to render VTK applications using FlowVR Render with minimal modifications of the original code. VTK FlowVR enalbes to encapsulate VTK code into FlowVR modules to get access to the FlowVR capabilities for modularizing and distributing VTK processings.

The FlowVR suite is freely available under a GP/LGPL licence at http://flowvr.sf.net with a full documentation and related publications.

5.2. Kaapi - Kernel for Asynchronous, Adaptive, Parallel and Interactive Application

Participants: V. Danjean, T. Gautier [correspondant], F. Wagner.

Kaapi is an efficient fine grain multithreaded runtime that runs on more than 500 processors and supports addition/resilience of resources. Kaapi means *Kernel for Asynchronous, Adaptive, Parallel and Interactive Application*. Kaapi runtime support uses a macro data flow representation to build, schedule and execute programs on distributed architectures. Kaapi allows the programmer to tune the scheduling algorithm used to execute its application. Currently, Kaapi only considers data dependencies between multiple producers and multiple consumers. A high level C++ API, called Athapascan and developed by the APACHE project, is implemented on top of Kaapi. Kaapi provides methods to schedule a data flow on multiple processors and then to evaluate it on a parallel architecture. The important key point is the way communications are handled. At a low level of implementation, Kaapi uses an active message protocol to perform very high performance remote write and remote signalization operations. This protocol has been ported on top of various networks (Ethernet/Socket, Myrinet/GM). Moreover, Kaapi is able to generate broadcasts and reductions that are critical for efficiency.

The performance of applications on top of Kaapi scales on clusters and large SMP machines (Symmetric Multi Processors): the kernel is developed using distributed algorithms to reduce synchronizations between threads and UNIX processes. Kaapi, through the use of the Athapascan interface, has been used to compute combinatorial optimization problems on the French Grid Etoile and Grid5000.

The work stealing algorithm implemented in Kaapi has a predictive cost model. Kaapi is able to report important measures to capture the parallel complexity or parallel bottleneck of an application.

Kaapi is developed for UNIX platform and has been ported on most of the UNIX systems (LINUX, IRIX, Mac OS X); it is compliant with both 32 bits and 64 bits architectures (IA32, G4, IA64, G5, MIPS). All Kaapi related material are available at https://gforge.inria.fr/projects/kaapi/ under CeCILL licence.

5.3. TakTuk - Adaptive large scale remote execution deployment

Participant: G. 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:

- autopropagation: 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 communication to other nodes using this logical number
- informations redirection: I/O and commands status are multiplexed from/to the root node.

http://taktuk.gforge.inria.fr under GNU GPL licence.

6. New Results

6.1. Parallel algorithms, complexity and scheduling

6.1.1. Scheduling

The work on scheduling mainly concerns multi-objective optimization and jobs scheduling on resources grid (ARC OTAPHE). We have exhibited techniques to find good trade-off between criteria that are commonly antagonistic; one major result is a scheduling competitive simultaneously for both average completion time and makespan.

Two emerging subjects have been initiated last year, namely the use of game theory to solve complex resource management problems and how to deal with uncertainty and disturbance in classical Combinatorial Optimization problems.

6.1.2. Adaptive algorithm

The main results concern the performance prediction of parallel adaptive algorithms; it enables to develop adaptive parallel programs for various applications. This work was done by most members of MOAIS team and has lead to the development of parallel and adaptive schemes of computation for different applications, studied by other research teams in Grenoble and Lyon within the IMAG-INRIA project AHA:

- 3D vision (E Boyer, C Menier, B Raffin, JL Roch, L Soares, E Hermann);
- computer algebra with Givaro/Linbox in collaboration with LJK (JG Dumas) (T Gautier, JL Roch); (internship of Marc Tchiboukdjian);
- cryptography (differential and algebraic analysis of symmetric boxes) in collaboration with Institut Fourier (R Gillard) (V Danjean, JL Roch);
- quadratic assignment problem (X Besseron, VD Cung, T Gautier, S Jafar, JL Roch);
- dynamic deployment on network (G Huard, T Gautier).

The runtime behavior is mainly based on the workstealing scheduling algorithm of Kaapi. Within a collaboration with ST, Kaapi is currently ported on MPSoCs (MultiProcessorS on Chips). We have demonstrated the efficiency of the adaptive scheme for stream computation (eg video encoding) within our collaboration with ST through the MINALOGIC/EmSoc Sceptre project.

6.1.3. Adaptive Octree for interactive 3D modelling

A work has been performed to develop an anytime and adaptive parallel algorithm for real time octree construction. The target application is 3D modeling: an octree is computed from projecting each voxel into a set of images taken from several cameras. By using a modified work-stealing approach that ensures the octree exploration is always balanced (width-first octree exploration), the algorithm can be stopped at any time to respect the real time constraints. Experimental results show a speed-up reaching 14.4 on a 16 processor SMP machine.

Here, due to interactivity, the time limit T is fixed (typically 20 ms). We have proved that our adaptive algorithm on p identical processors compute almost the same result than the reference sequential one on a single processor but with a time limit pT. This property, that has also been experimentally validated, is very important: the parallel adaptive algorithm enables to efficiently increase precision while managing interactivity constraints.

On-going developments focus on balancing the work load on CPUs as well as on GPUs to further improve the performance. It requires a dynamic coupling of a CPU specific algorithm and a GPU specific one.

6.1.4. Adaptive parallel prefix and extensions

Parallel prefix is a famous folk scheme in parallel programming of great practical importance. For this problem, even if fine grain fast parallel algorithms are known, decreasing the (parallel) time requires to increase the number of operations to perform and thus the load of the system. Based on an on-line coupling of a sequential algorithm and a recursive extraction of parallelism by work-stealing, a near-optimal parallel prefix algorithm has been exhibited on p processors with changing speeds (PhD thesis of Daouda Traore). We have extended this scheme to provide a near optimal parallel algorithm which does not use the number p of processors. Moreover, the scheme has been used to implement various algorithms on containers (some of STL algorithms) both in C++ on top of Kaapi and C on top of AWS. Comparizons with Intel TBB corresponding STL-like implementation on multicore systems and experimentations on small size distributed architectures have exhibit its good practical performance.

Such algorithms are called *processor oblivious*. This scheme is generic and could be applied to a wide spectrum of problems, not only in the context of the ANR Safescale and MINALOGIC Spectre contracts. We will apply it to sorting and mesh partitioning.

6.1.5. Safe distributed computation

We have developped an efficient checkpointing of the dataflow that described the computation at coarse-grain.

- a scalable checkpoint/restart protocol based on the stack has been developed. In order to improve efficiency, a coordinated checkpoint mechanism is currently developed (Xavier Besseron thesis) and is being integrated in Kaapi (Liyun Ye-Guelton engineer work within CHOC contract for huge combinatorial optimization computations).
- A probabilistic algorithm for malicious attack detection has been developed (thesis of Sébastien Varrette) in the framework of collaborations with Université du Luxembourg (F Leprevost) and University of Idaho (A Krings). Within the BGPR-SafeScale contract, it has been experimented with Kaapi on Grid'5000 for detection of forgery. We have proved this mechanism to be efficient in average for in-tree, out-tree and strictly nested parallel computations; this enables the development of fault-tolerant exact linear algebra algorithms based on error-correcting codes (such as large matrix-vector iterations).

Concerning the use of grid computing to evaluate robustness of cryprotgraphic protocols, a code, developped by Roland Gillard (Institu Fourier) that generates S-boxes with no quadratic relations has been parallelized thanks to Kaapi and executed on Grid5000. As a result, 2^{40} S-boxes have been tested on 2120 processors in a time between 24 hours and 65 hours, among them 6 have differential invariant $\delta = 8$ and linear invariant $\lambda = 28$.

6.2. Software

6.2.1. FlowVR Suite

The latest FlowVR Suite release (version 1.3.1) was downloaded 207 times between February and November 2007. The main changes for 2007 are:

- FlowVR now uses the cmake utility to simplify its installation.
- The LIFO, Université d'Orléans added a VRPN support to FlowVR.

- On previous versions of FlowVR, the application description language was based on a flat description using a mix of XML and Perl. We developed a new approach based on C++ and a hierarchical application description following the Fractal model. This new approach enables to clearly separate the description of the target architecture from the application architecture. Applications description is very compact, modular (an application can become a component for an other application without recompilation), and portable. An iterative process resolve the missing elements of the described application taking into account a file describing the target architecture. For instance collective communication are computed during this step when the process is able to identify the different source and destination components. We expect this ADL to become the default one for the next FLowVR release. This work presented at NPC 2007, received the *Excellent Student Paper Award* [26].

6.2.2. Fault-tolerance in KAAPI

We have developed a new algorithm to have a high performance fault tolerant mechanism in KAAPI. The protocol is based on coordinated checkpointing. The algorithm is well suited for iterative parallel application. The originality of our protocol is to allows partial restart of processes after detection of a fault.

6.2.3. Scalability of KAAPI

KAAPI software has been tested on whole Grid50000 during the 4th PLUGTEST event organised by ETSI and project OASIS at Beijing, China, Octobre, 29th - November, 1st, 2007. The implemented workstealing algorithm has demonstrated its capacity to schedule fine grain programs on 3654 cores using two level scheduling strategy: a thief tries to steal work first from a thread running on the same process. In case of failure, the thief emits steal request to an other process. The KAAPI team took part of the NQueens contest during the PLUGTEST event and was the winner in front of 7 teams (from China, Poland, France).

6.2.4. GRID5000: scheduling algorithm for OAR and authentication

OAR is a batch scheduler developed by Mescal team. 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.

7. Contracts and Grants with Industry

7.1. Technology transfer to 4D Views Solutions

The real time 3D modeling software developed in collaboration with the PERCEPTION project was transfered to the 4D Views Solutions start-up. 4D views has the exclusivity on this software.

7.2. BDI co-funded CNRS-STM with ST Microelectronics, 05-08

STM is cofunding a PhD thesis in collaboration with MOAIS. This PhD focuses on the design of adaptive multimedia applications on MPSoC (Multi-Processor System on Chip). The target application is MPEG encoding. The goal is to provide SystemC components that enable the development of SystemC applicative component that can be ported on different MPSoCs configurations with provable performances. The key point is the scheduling which is based on the technology that MOAIS has developed in Kaapi (distributed workstealing with coupling of an efficient sequential code and a paralel fine grain parallelism extraction). It consists in the specification and implementation of AWS, a dedicated version of Kaapi software for MPSoCs abstract architectures. The validation is performed on experimental MPSoC platforms provided by STM and on a simulation platform provided by TIMA.

7.3. BDI funded by C-S, 07-10

C-S is funding a PhD thesis in joined collaboration with MOAIS and Institute Fourier (Roland Gillard). This PhD is focused on the dimensioning and the integration of a symmetric cipher in the context of a large scale distributed infrastructure. 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 confidency and security in a large scale infrastructure.

7.4. BDI co-funded CNRS and CEA/DIF, 07-10

CEA/DIF is cofunding a PhD these in collaboration with MOAIS. 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 isosuMrface extraction) for large meshes (irregular and adaptive) that are proven efficient without requiring the mesh layout or the algorithm to actually know the memory hierachy of the target architecture or the number of processor available. We will conduct experiments rendering large data sets provided by the CEA/DIF on NUMA machines. We will also study the benefits of such approaches for programming GPUs.

7.5. Contract with DCN, 05-08

The objective of the contract is to provide an efficient evaluation and planification of actions with real-time reactivity constraints and multicriteria performance guarantees. This contract is joined with POPART INRIA team (realtime aspects) and ProBayes company (probabilistic inference engine ProBT). MOAIS is in charge of the planification, which is computed on a parallel scalable architecture and adaptive to suit reactivity and performance constraints.

8. Other Grants and Activities

8.1. Regional initiatives

SCEPTRE, 06-09, Minalogic: Started in 10/2006, SCEPTRE is a joint project with ST (coordinator), INRIA Rhône-Alpes (MOAIS, MESCAL, ARENAIRE, COMPSYS), IRISA (CAPS), TIMA-IMAG and VERIMAG. Within the SCEPTRE project, MOAIS is transfering its technology of fine grain workstealing to support adaptive multimedia applications on MPSoCs that include from 10 to 100 processors on a single chip (general purpose units, DSP, ...).

8.2. National initiatives

- *FVNANO*, 07-10, ANR-CIS: the project focuses on developping 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 absed on SOFA for fast dybnamic simulations. Partners : projects AVASION and MOAIS (INRIA Rhône-Alpes), 3S-R, IPSC-ELSA, CEG-DGA, LEES, LaM, INERIS, IRSN, CEA, SME Environnement, Phimeca, Bull.
- *DALIA*, 06-09, 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).

- *BGPR/SAFESCALE*, 05-08, ARA Sécurité: the projects deals with adaptive and safe computations on global computing platforms. Since october 2006, Serge Guelton has been recruited as an engineer on this contract. T A version of Kaapi has been provided with documentation to partners of the contract, together with an interface for distributed containers. The thesis of Sébastien Varrette (presented in 09/2007) proposed a probabilistic detection against massive attack and evaluated it within SafeScale on Grid'5000. Partners: LIPN (Paris XIII), IRISA (Rennes), ENST (Brest), VASCO team (LSR Grenoble), LMC-IMAG and Institut Fourier (Grenoble).
- *CHOC*, 06-09, ANR Grid. The project deals with combinatorial problems and software to compute exact and approximate solutions over a grid. Partners: PRiSM (Versailles), LIFL (Lille), GILCO (Grenoble), MOAIS (Grenoble)
- *DISCO*, 06-09, ANR Grid. The project deals with evaluating middleware to do scientific computation over computational grid. Partners: CAIMAN (Sophia-Antipolis), OASIS (Sophia-Antipolis), SMASH (Rennes), PARIS(Rennes), LABRI (Bordeaux), EAD (Toulouse), MOAIS (Grenoble)
- *GRID'5000*, the french grid platform. MOAIS has participated to the development of the distibuted authentication protocol for Grid5000 (namely deployment with TakTuk, scheduling policies in OAR and distributed authentication based on LDAP).

8.3. International initiatives

8.3.1. Europe

The project MOAIS participates to the Network Of Excellence CoreGrid (workpackages 6 - scheduling).

8.3.2. Poland

Bilateral agreement between the CNRS and the Polish Academy of Sciences, Warsaw, focused on the scheduling in embedded systems and SoC (2004-2007)

8.3.3. 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 2007 by 3 different grants:
  - PICS CNRS (2005-2007).
  - Capes/cofecub (2006-2008).
  - Equipe associée INRIA Diode-A (2006-2008).
- 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.

8.3.4. USA

LINBOX project with the university of Delaware (Dave Saunders) LMC-IMAG (Grenoble) et ARENAIRE (LIP-ENSL, Lyon).

8.4. Hardware Platforms

8.4.1. The GRIMAGE platform

The GrImage platform (http://www.inrialpes.fr/grimage) gathers a 16 projector display wall, a network of cameras and a PC cluster. It is dedicated to interactive applications. GrImage is co-leaded by the Moais and Perception projects (participants are the MOAIS, PERCEPTION, EVASION and ARTIS projects). It is the milestone of a strong and fruitfull 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. A display wall built around commodity video projectors provides a large and very high resolution display. The main goal is to provide a visualization space for large data sets and real time interaction.

The first part of GrImage (75 Keuros) was funded in 2003 by the INRIA and the Ministère de la Recherche (via INPG). The second part (50 Keuros) was funded by the INRIA. Some equipments are directly funded by the MOAIS and PERCEPTION projects through different contracts.

The Moais, Perception and Evasion project-teams brought to the Siggraph 2007 Conférence, San Diego, a small scale of the GrImage platform. During 5 days we demonstrated a camera based approach for markerless 3D interactions. This demo was selected with 22 others amongst 75 submissions. The demo relies on the FlowVR middelware, developped by Moais, to reach the real-time performance. See http://www.inrialpes.fr/grimage/#siggraph vor pictures, videos and some press articles.

8.4.2. SMP Machines

MOAIS invested in 2006 on two SMP architectures:

- A 8-way SMP machine equipped with Itanium processors.
- A 8-way SMP machine equipped with dual core processors (total of 16 cores) and 2 GPUs. This machine is connected on the 10 Gigabit Ethernet backbone connecting the Icluster-2, GrImage and Id-Pot clusters.

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 new generations of parallel adaptive algorithms taking advantage of the processing power provided by the multiple CPUs and GPUs available.

8.4.3. MPSoC

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

9. Dissemination

9.1. Leadership within scientific community

- Program committees :
  - Program committee HCW'07 (16th IEEE Heterogeneous Computing Workshop), Long Beach, California (march 2007)
  - Program committee ESCAPE, Hangzhou, China (april 2007)
  - Program committee workshop PMGC (Advances on programing models both for grid and cluster computing), Rio de Janeiro, Brazil (may 2007)
  - comité de programme de la conférence FUN'07 (the Fourth Conference on Fun with Algorithms) Isola d'Elba, Italy (2007)
  - Program committee MISTA'07, Paris, France (august 2007)
  - Program committee PPAM 2007 (the seventh international conference on Parallel Processing and Applied Mathematics), Gdansk, Poland, (sept. 2007)
  - Program committee PBC'07 (second workshop on Parallel Computational Biology, Gdansk, Poland (september 2007)

  - Program committee HeteroPar 07 (the sixth International Workshop on Algorithms, Models and Tools for parallel computing on heterogeneous networks), Austin, USA (september 2007)
  - Program committee SBAC-PAD 2007 (the 19th International Symposium on Computer Architecture and High Performance Computing, Brazil (november 2007)
  - Program committee ParCo2007 (Parallel Computing), Germany (september 2007)
  - Program committee IEEE AINA2008, Okinawa, Japan (march 2008)
  - Program committee HCW'08 (17th IEEE Heterogeneous Computing Workshop), Miami, USA (april 2008)
  - Program committee RENPAR'08 (18-ièmes Rencontres francophones du parallélisme), Fribourg, Suisse (february 2008)
  - Program committee PMAA'08 (the 5th International workshop on parallel matrix algorithms and applications), Neuchatel, Switzerland (june 2008)
  - Program committee IPDPS (the 19th International Parallel and Distributed Processing Symposium) , Miami, USA (april 2008)
  - Program committee of IEEE VR 2008 (Virtual Reality), Reno, Nevada.
  - Program comittee of EGPGV 2008 (Eurographics Symposium on Parallel Graphics and Visualization), Crete, Grece.
  - Program comittee of SVR 2008 (Symposium on Virtual and Augmented Reality), João Pessoa, Brazil.

- Members of editorial board : Calculateurs Parallèles, collection *Studies in Computer and Communications Systems*-IOS Press;*Handbook on Parallel and Distributed Processing, Springer Verlag*; *Parallel Computing Journal, series Advances in parallel processing,Elsevier Press*; ARIMA Journal; Parallel Computing Journal. IEEE Transactions on Parallel and Distributed Systems (TPDS).
- Member of the steering board of the EGPGV workshop (Eurographics Symposium on Parallel Graphics and Visualization).

9.2. Invited Talks

- Bruno Raffin, *Grimage: MarkerLess 3D Interactions*, Game Developer Conference, Lyon, December 2007.
- Jean-Louis Roch, *Processor oblivious parallel algorithms with provable performances*, Workshop Interactive Parallel Computation in Support of Research in Algebra, Geometry and Number Theory, MSRI Berkeley, USA Janvuary 2007.
- Denis Trystram, *Scheduling in Computational grids*, AEOLUS Workshop, Nice, May 2007
- Denis Trystram, *Multi-objective scheduling*, Ecole de Printemps d'Informatique Théorique, Frejus, June 2007

10. Bibliography

Major publications by the team in recent years

[1] J. ALLARD, B. RAFFIN. *A Shader-Based Parallel Rendering Framework*, in "IEEE Visualization Conference, Minneapolis, USA", October 2005.

[2] P. 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", vol. 16, n^{o} 2, april 2005, p. 217-237.

[3] S. JAFAR, A. W. KRINGS, T. GAUTIER, J.-L. ROCH. *Theft-Induced Checkpointing for Reconfigurable Dataflow Applications*, in "IEEE Electro/Information Technology Conference , (EIT 2005), Lincoln, Nebraska", This paper received the EIT'05 Best Paper Award, IEEE, May 2005.

[4] A. LÈBRE, Y. DENNEULIN, G. HUARD. *Cluster-Wide Adaptive I/O Scheduling for Concurent Parallel Applications*, in "IEEE international conference on Cluster Computing Proceedings", 2006.

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

Year Publications

Books and Monographs

[6] B. CHEN, M. PATERSON, G. ZHANG (editors). *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, First International Symposium, ESCAPE 2007, Hangzhou, China, April 7-9, 2007, Revised Selected Papers*, Lecture Notes in Computer Science, vol. 4614, Springer, 2007.

[7] J.-G. DUMAS, J.-L. ROCH, E. TANNIER, S. VARRETTE. *Théorie des Codes — Compression Cryptage Correction*, Dunod Sciences-Sup, Paris, France, February 2007, http://www.dunod.com/pages/ouvrages/ficheouvrage.asp?id=50692.

Doctoral dissertations and Habilitation theses

[8] C. MÉNIER. *Système de vision temps-réel pour les interactions*, Ph. D. Thesis, INPG, 2007.

[9] L. PIGEON. *Environnement interopérable distribué pour les simulations numériques avec composants CAPE-OPEN*, Ph. D. Thesis, Institut National Polytechnique de Grenoble (INPG), September 2007.

[10] S. VARRETTE. *Sécurité des Architectures de Calcul Distribué: Authentification et Certification de Résultats*, Ph. D. Thesis, INP Grenoble et Université du Luxembourg, September 2007.

Articles in refereed journals and book chapters

[11] C. BRIZUELA, L. GONZALEZ-GURROLA, A. TCHERNYKH, D. TRYSTRAM. *Sequencing by hybridization: an enhanced crossover operator for a hybrid genetic algorithm*, in "Journal of Heuristics", vol. 13, n^{o} 3, June 2007, p. 20ç-225.

[12] F. DIEDRICH, K. JANSEN. *Faster and simpler approximation algorithms for mixed packing and covering problems*, in "Theor. Comput. Sci.", vol. 377, n^{o} 1-3, 2007, p. 181-204.

[13] S. JAFAR, A. KRINGS, T. GAUTIER. *Flexible Rollback Recovery in Dynamic Heterogeneous Grid Computing*, in "IEEE Transactions on Dependable and Secure Computing", 2007.

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

[15] K. RZADCA, D. TRYSTRAM. *Promoting cooperation in selfish grids*, in "European Journal of Operational Research", to appear 2007.

Publications in Conferences and Workshops

[16] J. ALLARD, C. MÉNIER, B. RAFFIN, E. BOYER, F. FAURE. *Grimage: Markerless 3D Interactions*, in "Proceedings of ACM SIGGRAPH 07, San Diego, USA", Emerging Technology, August 2007.

[17] X. BESSERON, L. PIGEON, T. GAUTIER, S. JAFAR. *Un protocole de sauvegarde / reprise coordonné pour les applications à flot de données reconfigurables*, in "TSI", Hermès, 2007.

[18] F. BLACHOT, G. HUARD, J. PECERO, E. SAULE, D. TRYSTRAM. *Scheduling instructions on processors with incomplete bypass*, in "Proceedings of the eight workshop on models and algorithms for planning and scheduling problems", July 2007.

[19] V. DANJEAN, R. GILLARD, S. GUELTON, J.-L. ROCH, T. ROCHE. *Adaptive Loops with Kaapi on Multicore and Grid: Applications in Symmetric Cryptography*, in "Parallel Symbolic Computation'07 (PASCO'07), London, Ontario, Canada", ACM, July 2007.

[20] F. DIEDRICH, K. JANSEN. *An Approximation Algorithm for the General Mixed Packing and Covering Problem*, in "ESCAPE", 2007, p. 128-139.

[21] F. DIEDRICH, F. PASCUAL, D. TRYSTRAM. *Approximation Algorithms for Scheduling with Reservations*, in "HiPC 2007, the 14th Annual IEEE International Conference on High Performance Computing, Goa, India", December 2007.

[22] J. J. DONGARRA, E. JEANNOT, E. SAULE, Z. SHI. *Bi-objective Scheduling Algorithms for Optimizing Makespan and Reliability on Heterogeneous Systems*, in "SPAA '07: Proceedings of the nineteenth annual ACM Symposium on Parallelism in Algorithms and Architectures", ACM press, June 2007, p. 280-288.

[23] L. EYRAUD-DUBOIS, G. MOUNIÉ, D. TRYSTRAM. *Analysis of Scheduling Algorithms with Reservations*, in "International Parallel and Distributed Processing Symposium, IPDPS", IEEE, 2007, p. 1-8.

[24] T. GAUTIER, X. BESSERON, L. PIGEON. *KAAPI: A Thread Scheduling Runtime System for Data Flow Computations on Cluster of Multi-Processors*, in "Parallel Symbolic Computation'07 (PASCO'07), London, Ontario, Canada", n^{o} 15–23, ACM, 2007.

[25] T. GAUTIER, J.-L. ROCH, F. WAGNER. *Fine Grain Distributed Implementation of a Dataflow Language with Provable Performances*, in "Workshop PAPP 2007 - Practical Aspects of High-Level Parallel Programming in International Conference on Computational Science 2007 (ICCS2007), Beijing, China", IEEE, May 2007.

[26] J.-D. LESAGE, B. RAFFIN. *A Hierarchical Programming Model for Large Parallel Interactive Applications*, in "IFIP International Conference on Network and Parallel Computing, Dalian, China", Lecture Notes in Computer Science, Excellent Student Paper Award, vol. 4672, Springer, September 2007, p. 516-525.

[27] F.-K. MOULAÏ, G. MOUNIÉ. *Bi-criteria Scheduling Algorithm with Deployment in Cluster*, in "Proceedings of 16th Heterogeneity in Computing Workshop, HCW, in conjonction with IPDPS", 2007, p. 1-7.

[28] F. PASCUAL, K. RZADCA, D. TRYSTRAM. *Cooperation in Multi-Organization Scheduling*, in "EUROPAR 2007, Rennes, France", LNCS, n^{o} 4641, Springer Verlag, August 2007.

[29] J.-L. ROCH. *Processor-oblivious parallel algorithms with provable performances*, in "Workshop Interactive Parallel Computation in Support of Research in Algebra, Geometry and Number Theory, Math. Science Research Inst., Berkeley, USA", January 2007.

[30] J.-L. ROCH, S. VARRETTE. *Probabilistic Certification of Divide & Conquer Algorithms on Global Computing Platforms. Application to Fault-Tolerant Exact Matrix-Vector Product*, in "Parallel Symbolic Computation'07 (PASCO'07), London, Ontario, Canada", ACM, July 2007.

[31] K. RZADCA. *Scheduling in Multi-Organization Grids: Measur- ing the Inefficiency of Decentralization*, in "SPC'07, Workshop on Scheduling for Parallel Computing of Seventh International Conference On Parallel Processing And Applied Mathematics (PPAM 07)", Lecture Notes in Computer Science, Springer, to appear 2007.

[32] K. RZADCA, D. TRYSTRAM, A. WIERZBICKI. *Fair Game-Theoretic Resource Management in Dedicated Grids*, in "CCGrid 2007, the 7th IEEE International Symposium on Cluster Computing and the Grid, Rio de Janeiro, Brazil", May 2007.

[33] L. SOARES, C. MÉNIER, B. RAFFIN, J.-L. ROCH. *Parallel Adaptive Octree Carving for Real-time 3D Modeling*, in "IEEE Virtual Reality Conference, Charlotte, USA", Poster, March 2007.

[34] L. SOARES, C. MÉNIER, B. RAFFIN, J.-L. ROCH. *Work Stealing for Time-constrained Octree Exploration: Application to Real-time 3D Modeling*, in "EGPGV, Lugano,Switzerland", May 2007.

[35] L.-A. STEFFENEL, M. MARTINASSO, D. TRYSTRAM. *Assessing contention effects on MPI-Alltoall communications*, in "GPC'07, International Conference on Grid and Pervasive Computing, Paris, France", May 2007.

[36] D. TRYSTRAM. *Cooperation in multi-organization grids*, in "Workshop EuMedGrid. Grid computing: e-infrastructure, applications and research, Tunis, Tunisia", invited talk, November 2007.

[37] D. TRYSTRAM, J. ZOLA. *Multiple sequence alignment and phylogenetic inference*, in "Grid Computing for Bioinformatics and Computational Biology, edited by E.G. Talbi and A.Y. Zomaya", John Wiley and Sons, 2007.