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

Table of contents

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

1. Team

Research Scientist

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

Faculty Member

Jean-Louis Roch [Assistant Professor, INPG]
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]

Technical Staff

Fabrice Salpetrier [6 months]
Christophe Laferrière [2008]
Emile Morel [2008]
Antoine Vanel [2008]

PhD Student

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]
Mohamed-Slim Bouguerra [2008, INRIA Cordi]
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]
Yanik N'Goko [2006, co-tutelle Univ. Yaoundé, Cameroon, SARIMA scholarship]
Jonathan E. Pecero-Sanchez [2003, SFERE CONACYT scholarship]
Swann Perarnau [2008, MRNT scholarship]
Benjamin Petit [2007, common to PERCEPTION and MOAIS]
Jean-Noel Quintin [2008, CILOE constract 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]
Haifeng Xu [2007, co-tutelle INPG – Zhejiang University, Hangzhou, China (Guochuan Zhang)]

Visiting Scientist

Alfredo Goldman [USP Sao Paulo Brazil, 12 months]
Nicolas Maillard [UFRGS Porto Alegre, 1 month]
Andreï Tchernykh [CICESE, Ensenada, Mexico, 2 weeks]

Marek Tudruj [Polish Academy of Sciences, Warsaw, 1 week]
Lukasz Masko [Polish Academy of Sciences, Warsaw, 1 week]

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

2. Overall Objectives

2.1. Introduction

The MOAIS project-team is dedicated to end-to-end parallel programming solutions. The objective is to provide parallel programming schemes, interfaces and tools for high performance interactive computing that enable to achieve provable performances on distributed parallel architectures, from multi-processor system on chips to lightweight grids and global computing platforms.

In particular 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-objective, for instance associating precision, fluidity and reactivity in interactive simulations.

Ideally, to achieve portability, the application should be independent of 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. 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 in 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 design**: 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.

- **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 (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.
- 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 practical, centered on applications developed with external partners and experimented on parallel platforms. Significant efforts are made to build, manage and maintain these experimental platforms. We are involved with other teams in four main platforms:

- Grimage (25 cameras, a 30 node PC cluster and a 16 projector display wall - http://www.inrialpes.fr/grimage),
- I-cluster 2 (200 Itanium 2 cluster - http://i-cluster2.inrialpes.fr/)
- a 8-way SMP machine equiped with Itanium processors
- a 8-way machine equipped with dual-core Opteron processors and 2 programmable NVIDIA GPUs
- the Grid'5000 national grid (https://www.grid5000.fr/)

2.2. Highlights of the year

- After the 2007 successful selection at SIGGRAPH Emerging Technologies, the Moais, Perception and Evasion project-teams have presented a portable version of the Grimage platforms at various events: not only scientific international conferences (ECCV'08 in October and VRST'08 in November), but yet successful "grand public" exhibitions and three French TV reports. The top one was in Paris at the event "Ville européenne de la science" (European science city) exposition that took place in Paris in "Grand Palais" from Friday 14th to Sunday 16th of November. The platform was presented in the Inria's "Digital village". At this occasion, we were filmed by the national french TV channel TF1.

- In collaboration between the INRIA Moais team-project (Xavier Besseron, Thierry Gautier, Guillaume Huard) and the MathFi team of the LJK laboratory (Emmanuel Gobet), the "Kaapi/Taktuk" team has won the 5th international challenge GRIDS@WORK "2008 Super Quant Monte Carlo" that took place in Nice, France on October 23th. The goal was to precisely price 1000 contracts with exotic options on high dimension portofolio. For such a classical challenge in financial engineering, "Kaapi/Taktuk" used about 4000 cores during one hour on two distributed grids, one in Japan, the other in France with sustained efficiencies.

3. Scientific Foundations

3.1. Scheduling

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

Participants: Pierre-François Dutot, Thierry Gautier, 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 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: 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, 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: 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 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: 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 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: 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. 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 Ménier 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: Thierry Gautier, Jean-Louis Roch, Vincent Danjean, Frédéric 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.

- **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: Vincent Danjean, Thierry Gautier, 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. 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. In order 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 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: 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 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: Jean-Denis Lesage, 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 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: Vincent Danjean, Thierry Gautier [correspondant], Frédéric Wagner.

Kaapi is an efficient fine grain multithreaded runtime that runs on more than 1000 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: 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:

- 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. We have exhibited techniques to find good trade-off between criteria. Mainly we achieved two main results. First, we caracterized a multi-user problem with an algorithm achieving a constant approximation to the pareto curve [9]. This part is related to our previous works related to results of the game theory. Secondly, we extended the spectrum of multi-criteria results to include either numerous criteria [9] or new and radically different criteria like reliability or memory consumption versus execution time [13], [39].

6.1.2. Adaptive algorithm

New results, both theoretical and experimental, have been obtained with respect to the bi-criteria work/depth threshold in order to reach asymptotic optimal running time on distributed architectures with processors of heterogeneous frequencies.

Provable work-optimal parallelizations of STL (Standard Template Library) algorithms based on the work-stealing technique has been achieved [42]. Unlike previous approaches where a deque for each processor is typically used to locally store ready tasks and where a processor that runs out of work steals a ready task from the deque of a randomly selected processor, overhead for task creations is reduced based on an original implementation of work-stealing without using any deque but a distributed list. The implementation has been performed on top of Kaapi work stealing implementations and exibits near-optimal performances, improving other libraries such as Intel Thread Building Blocks (TBB) or MCSTL for partial-sum for instance. Among new applications, an in-place 2-way array partitioning has been implemented and used to implement quick sort and variants (sort/unstable sort in the STL) [44]. Most of those results are related to the phD thesis of Daouda Traore (presented on 19/12/2008 [10]). A complementary, yet unpublished, result in this thesis is the building of a static optimal algorithm for parallel prefix, based on the adaptive scheme, that achieves the tight lower bound with an asymptotic optimal number of cache miss. Currently, we are integrating the adaptive schem in a high level library, named Kastl, on top of Kaapi.

Based on our 2007 results on the adaptive coupling of GPU and CPU parallelism for interactive 3D modelling, another perspective is to take benefit of KASTL to provide fine grain adaptive parallelization of a part of the SOFA library.

In the context of embedded platforms, this on-line adaptive scheme has been proved optimal for simple stream computations [25]. Within our collaboration with ST through the MINALOGIC/EmSoc SCEPTRE project, a specialized implementation for embedded platforms, not relying on Posix nor sockets library, has been designed, resulting in a C library named AWS (Adaptive Work Stealing) which is a deliverable of the SCEPTRE project. AWS has been ported on three platforms: standard multicore architectures with Posix; a STM 8010 platform with 3 ST200 [OS21 and Multicom library]; a cycle accurate architecture simulator [delivered by TIMA lab]. AWS has been applied on a STM multimedia benchmark for HDTV: Temporal Noise R:w eduction (TNR). Most of those results are related to the phd thesis of Julien Bernard (preented on 8/12/2008 [6]), and co-funded by STM and CNRS.

6.1.3. Safe distributed computation

In the fail-silent model, an efficient coordinated checkpoint mechanism of the dataflow that described the computation at coarse-grain has been developed and integrated into Kaapi (Xavier Besseron phD Thesis). It extends the IEEE TDSC paper published this year[14] in case of iterative applications [26] in order to take into account the knowledge of the dependencies among processors to speedup restart time after a failure.

With respect to malicious faults (byzantine errors), a probabilistic certification platform has been designed that includes hardware crypto-processors [43]. This work has been performed within the ANR SAFESCALE project. Using a macro-data flow representation of the program execution, a complementary work, jointly developed with Paris team-project, is based on work-stealing scheduling to dynamically adapt the execution to sabotage while keeping a reasonable slowdown rate. Unlike static adaptation or adaptation at the source code level, a dynamic adaptation at the middleware level is proposed, enforcing separation of concepts and programming transparency. We are still extending algorithm-based fault-tolerance schemes for probabilistic certification in a more general context.

Finally, considering cryptographic primitives, we have proposed [37] a new way to bound the probability of occurrence of an n-round differential in the context of differential cryptanalysis. Hence this new model allows us to claim proof of resistance against impossible differential cryptanalysis, as initially defined by Bi- ham in 1999. This work is applied to CS-Cipher, to which, assuming some non-trivial hypothesis, provable security against impossible differential cryptanalysis is obtained.

6.2. Software

6.2.1. FlowVR

The FlowVR Suite was downloaded 500 in 2008. The main changes for 2008 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 C++ approach based on a hierarchical component oriented 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. This new framework does not led to any overhead at execution time. An iterative process extract from this herarchical appplication description a low level set of commands used to launch the application on a parallel machine. This work was published in the Jornal of SuperComputing.

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 two grid platforms, the French National Grid50000 and the Japanese Intrigger, during the 5th PLUGTEST event organised by ETSI and project OASIS at Sophia-Antipolis, France, Octobre, 20th - October, 24th, 2008. The KAAPI team took part of the Super Quant Monte Carlo contest during the PLUGTEST event and was the winner in front of 7 teams. Runs have shown the ability to fully exploit machines geographically distributed among France and Japan which demonstrates concrete communication between processes behind firewalls.

6.2.4. KAAPI on iPhoneOS

For the purpose of the participation of MOAIS team at the INRIA stand for SuperComputing 2008 at Austin, Texas, USA, the KAAPI software stack has been ported on the iPhoneOS operating system for embedded device such as Apple iPhone or iPodTouch. Several academic applications have been also ported in order to show the ability to schedule program with dynamic set of resources. This is the first step before porting part of Kaapi for ST embedded MPSoC.

6.2.5. 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 in collaboration with MOAIS has cofunded the PhD thesis of Julien Bernard that has been defended on 8/12/2008. 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.
The work keeps on within the SCEPTRE contract.

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, 05-09

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. It funds the phD thesis of Gérald Vaisman, co-advised by Moais and PopArt team projects.

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, ...).
- *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 a phD thesis that will start in 2009.

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 middleware stack used in 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://grimage.inrialpes.fr) 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.

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 CPC, Paris, France (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), Joâo Pessoa, Brazil.
  - Program comittee of IPDPS 2008, 14-18 april, Miami, USA
  - Program comittee of HCW'2008, 14 april, Miami, USA
  - co-chair of New Challenges of Scheduling Theory, 12-16 may, Luminy, France
  - Program comittee of Grid 2008, 29 sept-1 oct, Tsukuba, Japan
  - Program comittee of ISPDC 2008, 1-5 july, Cracow, Poland
  - Program comittee of CARI'2008, 27-30 october, Rabbat, Maroco
  - Program comittee of SBAC-PAD, 29 oct-1 nov 2008, Campo-Grande, 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).

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

[2] S. JAFAR, A. W. KRINGS, T. GAUTIER. *Flexible Rollback Recovery in Dynamic Heterogeneous Grid Computing*, in "IEEE Transactions on Dependable and Secure Computing", 2008.

[3] J.-D. LESAGE, B. RAFFIN. *A Hierarchical Component Model for Large Parallel Interactive Applications*, in "Journal of Supercomputing", Extended version of NPC 2007 article., July 2008, 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", vol. 37, n^{o} 2, 2007, p. 401–412, http://hal.archives-ouvertes.fr/hal-00002166/en/.

[5] D. TRAORE, J.-L. ROCH, N. MAIILARD, 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.

Year Publications

Doctoral Dissertations and Habilitation Theses

[6] J. BERNARD. *Algorithmes parallÃles auto-adaptatifs et applications*, Ph. D. Thesis, INPG, Dec 2008.

[7] J. E. PECERO-SÁNCHEZ. *Local Global scheduling interaction*, Ph. D. Thesis, Grenoble Institute of Technology, Grenoble, 2008.

[8] K. RZADCA. *Des modèles et des algorithmes pour la gestion des ressources dans les grilles de plusieurs organisations*, Ph. D. Thesis, join Grenoble-INP and Polish Japanese Institute of Information Technology, feb 2008.

[9] É. SAULE. *Algorithmes d'approximation pour l'ordonnancement multi-objectif. Applications aux systèmes parallèles et embarqués.*, Ph. D. Thesis, Institut polytechnique de Grenoble, novembre 2008.

[10] D. TRAORE. *Algorithmes parallèles auto-adaptatifs et applications*, Ph. D. Thesis, INPG, Dec 2008.

Articles in International Peer-Reviewed Journal

[11] F. DIEDRICH, R. HARREN, K. JANSEN, R. THÖLE, H. THOMAS. *Approximation Algorithms for 3D Orthogonal Knapsack*, in "J. Comput. Sci. Technol.", vol. 23, n^{o} 5, 2008, p. 749–762, http://dx.doi.org/10.1007/s11390-008-9170-7.

[12] F. DIEDRICH, K. JANSEN, F. PASCUAL, D. TRYSTRAM. *Approximation algorithms for scheduling with reservations*, in "Algorithmica", to appear, 2008.

[13] A. GIRAULT, É. SAULE, D. TRYSTRAM. *Reliability versus performance for critical applications*, in "Journal of Parallel and Distributed Computing, JPDC", to appear, 2009, http://dx.doi.org/10.1016/j.jpdc.2008.11.002.

[14] S. JAFAR, A. W. KRINGS, T. GAUTIER. *Flexible Rollback Recovery in Dynamic Heterogeneous Grid Computing*, in "IEEE Transactions on Dependable and Secure Computing", 2008.

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

[16] A. MAHJOUB, J. E. PECERO-SÁNCHEZ, D. TRYSTRAM. *Scheduling with uncertainties on new computing platforms*, in "Computational Optimization and Applications", to appear, 2008.

[17] W. NASRI, L. A. STEFFENEL, D. TRYSTRAM. *Adaptive approaches for efficient parallel algorithms on cluster-based systems*, in "International Journal of Grid and Utility Computing, IJGUC", vol. 1, n^{o} 2, 2009, http://inderscience.metapress.com/link.asp?id=0767h1286q173447.

[18] F. PASCUAL, K. RZADCA, D. TRYSTRAM. *Cooperation in Multi-Organization Scheduling*, in "Concurency&Computation: Practice and Experience", n° Extended version selected among the best papers of EuroPar 2008 (in press), 2008.

[19] K. RZADCA, D. TRYSTRAM. *Promoting Cooperation in Selfish Computational Grids*, in "European Journal of Operational Research", n° in press, 2008.

[20] K. RZADCA, D. TRYSTRAM. *Promoting cooperation in selfish grids*, in "European Journal of Operational Research", to appear, 2009, http://dx.doi.org/10.1016/j.ejor.2007.06.067.

[21] L. P. SOARES, B. RAFFIN, J. A. JORGE. *PC Clusters for Virtual Reality*, in "The International Journal of Virtual Reality", Extended Version of IEEE VR 20006 survey, vol. 7, n° 1, March 2008, p. 67–80.

[22] L. A. STEFFENEL, M. MARTINASSO, D. TRYSTRAM. *Assessing contention effects of all-to-all communications on clusters and grids*, in "International Journal of Pervasive Computing and Communiation", vol. 4, n° 4, 2008.

[23] L. A. STEFFENEL, G. MOUNIÉ. *A framework for adaptive collective communications for heterogeneous hierarchical computing systems*, in "Journal of Computer and System Sciences", vol. 74, 2008, p. 1082-1093, http://dx.doi.org/10.1016/j.jcss.2007.07.010.

[24] A. TCHERNYKH, D. TRYSTRAM, C. BRIZUELA, I. SCHERSON. *Idle regulation in non-clairvoyant scheduling of parallel jobs*, in "Discrete Applied Maths", n° in press, 2008.

International Peer-Reviewed Conference/Proceedings

[25] J. BERNARD, J.-L. ROCH, D. TRAORE. *Processor-oblivious parallel stream computations*, in "16th Euromicro International Conference on Parallel, Distributed and network-based Processing, Toulouse, France", Feb 2008, http://www.pdp2008.org/.

[26] X. BESSERON, T. GAUTIER. *Optimised recovery with a coordinated checkpoint/rollback protocol for domain decomposition applications.*, in "Modelling, Computation and Optimization in Information Systems and Management Sciences (MCO'08), Metz, France", Springer, Sept 2008, p. 497–506, http://www.lita.univ-metz.fr/~mco08/.

[27] F. DIEDRICH, K. JANSEN. *Improved Approximation Algorithms for Scheduling with Fixed Jobs*, in "Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms", to appear, 2009.

[28] F. DIEDRICH, B. KEHDEN, F. NEUMANN. *Multi-objective Problems in Terms of Relational Algebra*, in "RelMiCS", 2008, p. 84–98, http://dx.doi.org/10.1007/978-3-540-78913-0_8.

[29] F. DIEDRICH, F. NEUMANN. *Using Fast Matrix Multiplication in Bio-inspired Computation for Complex Optimization Problems*, in "Proceedings of the IEEE Congress on Evolutionary Computation 2008", 2008, p. 3828–3833.

[30] A. GOLDMAN, Y. NGOKO. *A Mixed Integer Linear Programming approach to schedule parallel independent tasks*, in "Proceedings of the 7th International Symposium on Parallel and Distributed Computing, Krakow, Poland", july 2008.

[31] E. HERMANN, F. FAURE, B. RAFFIN. *Ray-traced Collision Detection for Deformable Bodies*, in "3rd International Conference on Computer Graphics Theory and Applications (GRAPP), Madeira, Portugal", January 2008, p. 293–299.

[32] E. JEANNOT, É. SAULE, D. TRYSTRAM. *Bi-objective approximation scheme for makespan and reliability optimization on uniform parallel machines*, in "EuroPar 2008, Las Palmas, Spain", LNCS, 2008, p. 877–886.

[33] M. KOLBERG, D. CORDEIRO, G. BOHLENDER, L. G. FERNANDES, A. GOLDMAN. *A Multithreaded Verified Method for Solving Linear Systems in Dual-Core Processors*, in "Workshop on State-of-the-Art in Scientific and Parallel Computing (PARA 2008), Trondheim, Norway", May 2008.

[34] M. KRYSTEK, K. KUROWSKI, A. OLEKSIAK, K. RZADCA. *Comparison of Centralized and Decentralized Scheduling Algorithms using GSSIM Simulation Environment*, in "CoreGRID Integration Workshop Proceedings", 2008.

[35] J.-D. LESAGE, B. RAFFIN. *High Performance Interactive Computing with FlowVR*, in "IEEE VR 2008 SEARIS workshop, Reno, USA", Shaker Verlag, March 2008, p. 13–16.

[36] B. PETIT, J.-D. LESAGE, J.-S. FRANCO, E. BOYER, B. RAFFIN. *Grimage: 3D Modeling for Remote Collaboration and Telepresence*, in "15th ACM Symposium on Virtual Reality Software and Technology (VRST08), Bordeaux, France", On-site Demo, October 2008, p. 299–300.

[37] T. ROCHE, R. GILLARD, J.-L. ROCH. *Provable Security against Impossible Differential Cryptanalysis. Application to CS-Cipher.*, in "Modelling, Computation and Optimization in Information Systems and Management Sciences (MCO'08), Metz, France", Springer, Sept 2008, http://www.lita.univ-metz.fr/~mco08/.

[38] 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, 2008, http://dx.doi.org/10.1007/978-3-540-68111-3_111.

[39] É. SAULE, P.-F. DUTOT, G. MOUNIÉ. *Scheduling With Storage Constraints*, in "Electronic proceedings of IPDPS 2008", April 2008, p. 1–8.

[40] É. SAULE, B. VIDEAU. *PaSTeL. Une implantation parallèle de la STL pour les architectures multi-coeurs : une analyse des performances*, in "Proceedings électronique de RenPar 18", February 2008.

[41] L. M. SCHNORR, G. HUARD, P. O. A. NAVAUX. *3D Approach to the Visualization of Parallel Applications and Grid Monitoring Information*, in "Proceedings of the 9th IEEE/ACM International Conference on Grid Computing", September 2008.

[42] D. TRAORE, J.-L. ROCH, N. MAIILARD, 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.

[43] S. VARRETTE, J.-L. ROCH, G. DUC, R. KERYELL. *Building Secure Resources to Ensure Safe Computations in Distributed and Potentially Corrupted Environments*, in "Euro-Par 2008, Workshop on Secure, Trusted, Manageable and Controllable Grid Services (SGS'08), Las Palmas de Gran Canaria, Spain", Lecture Notes in Computer Science (LNCS), Springer, August 2008, http://www-lipn.univ-paris13.fr/~cerin/SGS.html.

National Peer-Reviewed Conference/Proceedings

[44] D. TRAORE, J.-L. ROCH, C. CERIN. *Algorithmes adaptatifs de tri parallele*, in "Proceedings électronique de RenPar 18, Fribourg, Suisse", RenPar, Feb 2008, http://www.renpar.org/.

Workshops without Proceedings

[45] D. CORDEIRO, A. GOLDMAN, D. TRYSTRAM. *Implicit Cooperation in Multi-Organization Clusters*, in "The 21st Conference of the European Chapter on Combinatorial Optimization (ECCO XXI), Dubrovnik, Croatia", may 2008.

[46] M. TCHIBOUKDJIAN, V. DANJEAN, B. RAFFIN. *A Fast Cache Oblivious Mesh Layout with Theoretical Guarantees*, in "1th International Workshop on Super Visualization, Island of Kos, Aegean Sea, Greece", ACM, June 2008.

[47] D. TRYSTRAM. *Multi-users, multi-organizations, multi-objectives: a single perspective*, in "2nd scheduling workshop, Aussois, France", Organized by L. Marchal et al. ENS Lyon, may 2008.

[48] G. VAISMAN, PIERRE-FRANÇOIS. DUTOT, A. GIRAULT. *Approche multi-critère : garantie par BOBPP*, in "9ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision", Feb 2008.

Scientific Books (or Scientific Book chapters)

[49] F. DIEDRICH, K. JANSEN, U. SCHWARTZ, D. TRYSTRAM. *A survey on approximation algorithms for scheduling with machine unavailability*, LNCS, to appear, Springer, 2008.