INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team moais

Multi-programmation et Ordonnancement pour les Applications Interactives de Simulation

Rhône-Alpes

THEME NUM

Activity Report

2005

Table of contents

1. Team

MOAIS project is a common project supported by CNRS, INPG, UJF and INRIA located in the ID-IMAG labs (UMR 5132).

Head of project team

Jean-Louis Roch [Assistant Professor, INPG]

Administrative staff

Marion Ponsot [INRIA Administrative Assistant, 30%]

INRIA Staff

Thierry Gautier [Research Scientist]
Bruno Raffin [Research Scientist]

INPG Staff

Grégory Mounié [Assistant Professor]
Denis Trystram [Professor]

UJF Staff

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

Project technical staff

Loïck Lecointre [RNTL Geobench - 03/03-05/05]

Invited Scientist

Axels Krings [IDAHO University, Moscow, USA, 1 year]
Andreï Tchernyk [CICESE, Ensenada, Mexico, 1 month]

PhD students

Jérémie Allard [2002, MRNT scholarship]
Thomas Arcila [2005, CIFRE Bull]
Julien Bernard [2005, BDI CNRS / ST Microelectronics scholarship]
Florent Blachot [2003, CIFRE ST Micro Electronics scholarship]
Pierre-Francois Dutot [2000, Normalien, BDI-CNRS MRNT scholarship]
Luiz-Angelo Estefanel [2002, Brasil CAPES scholarship]
Lionel Eyraud [2002, Normalien, MRNT scholarship]
Hamidi Hamid Reza [2001, SFERE scholarship]
Samir Jafar [2002, Syrian scholarship]
Garstecki Lukasz [2001, Polish scholarship, co-tutelle]
Clément Ménier [2003, Normalien, common to MOVI and MOAIS]
Feryal-Kamila Moulay [2003, CIFRE BULL scholarship]
Laurent Pigeon [2003, CIFRE IFP scholarship]
Jonathan Pecero-Sanchez [2003, CONACYT Mexican scholarship]
Krysztof Rzadca [2004, Polish scholarship, co-tutelle]
Eric Saule [2005, MRNT scholarship]
Daouda Traore [2005, Egide France-Mali scholarship]
Sébastien Varette [2004, Luxembourg scholarship]
Jesus Verduzco [2001, Mexican scholarship]
Jaroslaw Zola [2002, Polish scholarship, co-tutelle]

2. Overall Objectives

2.1. Overall Objectives

The goal of the MOAIS project is the programming of applications where performance is a matter of resources: beyond the optimization of the application itself, the effective use of a large number of resources is expected to enhance the performance. Target architectures are scalable ambient computing platforms based on off-the shelf components: input devices (sensors, cameras, ...), output devices (for visual, acoustic or haptic rendering ...) and computing units. Ideally, it should be possible to improve gradually performance by adding (dynamically) resources:

- precision is related to the size of the scheme or order of the model, which directly depends on the computing power (processors and memory space);
- the application control is related to the quality and number of input resources (sensors, cameras, microphones);
- a high quality visualization requires a large display with a high pixel density obtained by stacking multiple projectors or screens. Extra information can be provided to users through sound rendering or haptic systems. Synchronizations are required to ensure data coherency across those multiple outputs.

Those three levels, computations, inputs and outputs, enable users to interact with the application. In this interactive context, performance is a global multi-criteria objective associating precision, fluidity and reactivity.

Then, programming a portable application for such an ambient platform requires to suit to the available resources. Ideally, the application should be independent to the platform and should support any configuration: adaptation to the platform is managed by the scheduling. Thus, fundamental researches undertaken in the MOAIS project are focused on this scheduling problem which manages 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.

Also, 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 the 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, ambient 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 in 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. Application fields are: virtual reality and scientific computing (simulation, combinatorial optimization, biology, computer algebra). For real applications, code coupling is an important issue.

3. Scientific Foundations

3.1. Scheduling

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

Participants: T. Gautier, G. Huard, G. Mounié, J.-L. Roch, D. Trystram.

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 several 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).

3.1.1. Parallel tasks model and extensions

We have contributed to the definition of modern task models: malleable, moldable. We have developed techniques to derive, from an off-line scheduling algorithm, an efficient on-line one which has a good performance guarantee for rigid or moldable tasks. The method uses a batch framework where jobs are submitted to the cluster by a queue on a dedicated processor and where the whole batch has to be completed before starting a new one. The performance guarantee is no more than twice the optimal (which is the worst case).

3.1.2. Multi-objective Analysis.

A natural question while designing practical scheduling algorithms is "which criterion should we optimize ?". Most existing works have been developed for the objective of *makespan* minimization (time of the latest tasks to be executed). It 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 criteria which may be pertinent for specific use. Some of our work deals with the problem of designing scheduling algorithms that optimize simultaneously several criteria. 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*. Part of those theoretical results and other extensions have been implemented in the **OAR** batch scheduler within ACI CiGri and Grid5000.

3.1.3. 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.

Among applications that require a huge memory space for efficiency, we have studied a bioinformatic application, namely multiple alignment of biological sequences and building of specie trees (called phylogenic trees). Our contribution used our competence in combinatorial optimization and parallel computing. Molecular biologists work with large data sets and state-of-the-art algorithms use large processing power which may be provided by parallel processing. Due to its huge volume of computations and data, multiple alignments with tree construction need large size clusters and grids. We have proposed a new approach to solve the problem of parallel multiple sequence alignment based on the application of caching techniques. It is aimed to solve, with high precision, large alignment instances on heterogeneous computational clusters. A key point is to identify a computation in order to determine if it has already been performed. We solved this problem using a hash of the description of the dataflow graph related to the considered computation.

3.1.4. Coarse-grain scheduling of fine grain multithreaded computations.

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 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. Adaptative Parallel and Distributed Algorithms Design

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

Participants: T. Gautier, G. Huard, B. Raffin, J.-L. Roch, D. Trystram.

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. In order 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. We have implemented this first way of adapting granularity by porting Athapascan, the parallel programming interface developed by the APACHE project, on top of Kaapi. It has been successfully used to solve combinatorial optimization problems (QAP problems with PRISM laboratory and telecommunication application with LIFL within the ACI DOCG) and parallel evaluation of a tree to improve performances of a probabilistic inference engine (contract with the Pixelis company).

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 algorithms, a sequential one and a parallel one, 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. Within the IMAG-INRIA AHA project, we are applying this technique to develop adaptive algorithms for various applications: data compression, combinatorial optimization, iterated and prefix sum computations, 3D image reconstruction, exact computations.

3.3. Interactivity

Keywords: *image wall*, *interactivity*, *multimedia*.

Participants: V. Danjean, 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.

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 wishes of the user.

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).

As a first move, we propose to develop a programming environment 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 will enable the user to loosen coherency where it is acceptable, to improve asynchronism and thus performance. A first implementation, called FlowVR, is currently under development. However 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-adaptative 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, adaptative algorithmic schemes, distributed decision making, global stability and balance of the regulation effort.

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.

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. Panorama

Keywords: *code coupling*, *computer aided process engineering*, *fault-tolerance*, *linear algebra*, *multi-alignment and phylogeny*, *virtual reality*.

Participants: J. Allard, T. Gautier, G. Mounié, B. Raffin, J.-L. Roch, D. Trystram, J. Verduzco.

4.1.1. Virtual Reality

We will pursue and extend existing collaborations to develop virtual reality applications on PC clusters and grid environments. This work will rely on FlowVR but also on previous code developments like NetJuggler and SoftGenLock. Different actions will be considered:

- Multi-modal applications. An ongoing collaboration with the I3D group of INRIA Rhône-Alpes targets at coupling multi-projector visualization on workbench and haptic rendering on a PC cluster.
- Real time 3D modeling. An on going collaboration with the MOVI project focuses on developing solutions to enable real time 3D modeling from multiple cameras using a PC cluster. Clément Ménier, Ph.D. student co-directed by Edmond Boyer (MOVI) and Bruno Raffin, started on this topic in September 2003. We first target visual-hull reconstruction algorithms
- Seismic simulations. The goal is to design an interactive seismic simulation that will take advantage of a PC cluster to execute a parallel seismic simulation as well as a multi-projector rendering. This work is a join collaboration with the INRIA I3D group, the LIFO of the University of Orléans, the CEA, the BRGM and the TGS company, funded by the Geobench RNTL contract.
- 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 will take advantage of the skills and equipments available through the GrImage, I-cluster II, Ciment and Grid 5000 platforms we participate to.

4.1.2. Code Coupling

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.

- Coordination of Legacy Code in Homa. An ongoing project is to study an environment based on CORBA technology to automatically assemble components together with the possibility to extract parallelism between invocations and communications in order to re-schedule them to enable parallel data transfers.
- Cape-OPEN application. An on going collaboration with IFP (Institut Français du Pétrole) to study an high performance CAPE (Computer Aided Process Engineering) runtime for cluster architecture. CAPE-OPEN is an industrial standard of interface of components in process engineering. Some structural property of the application will be considered in order to reduce the computation into loosely coupled sub-computations.

4.1.3. Genomic – Multiple Alignments with Tree Construction

Multiple alignments ultimate aim is to construct ancestral sequences given a sequence set for actual species. To do this, one need to know the relationships between species, the *phylogeny*. This one appears to be a tree, with the common ancestor at the root, and sequences of the dataset at the leaves. Usually, when using a multiple alignment program, we do not have this knowledge, and estimation of the tree must be performed. Due to the huge volume of computations and data, multiple alignments with tree construction need large size clusters and grids.

Recently, we proposed a new approach to solve the problem of parallel multiple sequence alignment. The proposed method is based on the application of caching techniques and is aimed to solve, with high precision, large alignment instances on the heterogeneous computational clusters.

We use CacheFlow to store partial alignment guiding trees. It enables to reuse a result in future computations to eliminate redundancy.

4.1.4. Secure computations

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 or results modifications are part of operation, and applications have to deal directly with repeated failures during program runs. Moreover, security is crucial for some applications, like the medical application developped in the framework of the RAGTIME contract. The MOAIS team considers two kinds of failures:

- *Node failures and disconnections*: To ensure resilience of the application, fault tolerance mechanisms are to be used. For the sake of scalability, a global distributed consistent state is needed; it is obtained from both checkpointing locally each sequential process and logging their dependencies (e.g. communications in MPICH-V or Egida).
- *Task forgery*: the program is executed on a remote resource (also called *worker* in the sequel) and its expected output results may be modified on this resource with no control of the client application.

Our approach to solve those problems is based on the efficient checkpointing of the dataflow that described the computation at coarse-grain.

- a scalable checkpoint/restart protocol based on the stack has been developed (thesis of Samir Jafar);
- a probabilistic algorithm for malicious attack detection is being developed (thesis of Sébastien Varrette) in the framework of collaborations with Université du Luxembourg (F Leprevost) and University of Idaho (A Krings).

4.1.5. Embedded systems

In order 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 a CIFRE contract with ST (thesis F. Blachot), we consider the problem of large scale instruction scheduling within embedded VLIW processors, such as the ST200. We use optimal or near-optimal methods for the computation of the instructions schedule even if they are computationally prohibitive. The challenge here is the study automatic fine grain tuning and adaptation of a program to a given architecture, which is a key point to raise portable performances.

Another way of adressing MPSoCs programming is to use on-line efficient schedules. Kaapi is being transfered on MPSoCs platforms in collaboration with ST (Serge Paoli). First results (joined master thesis of J. Bernard in 2005), obtained on MPEG-4 encoding, are very promising. This work recently received additionals fundings (BDI grant cofunded by ST and CNRS) and is developped in the framework of the contract SCEPTRE inside the regional competitivity pole MINALOGIC/EMSOC (SCEPTRE is a joint project with ST (coordinator), INRIA Rhône-Alpes (MOAIS, MESCAL, ARENAIRE, COMPSYS), IRISA (CAPS), TIMA-IMAG and VERIMAG..

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 build with several dual-core processors and graphical processing units. New architectures, like the Cell processors, also integrates several computing units. First work will focus on balancing work load on multi GPU and CPU architectures for scientific visualization problems (CIFRE Ph.D. grant funded by BULL).

5. Software

5.1. Software

Achieving interoperability between softwares developed within the APACHE project (namely Athapascan and Net Juggler), the MOAIS project has been able to build up interactive application prototypes (distributed cloth simulation coupled with a multi-projector visualization).

Based on this experience, we are currently developing a new generation of softwares, more specifically designed for large scale distributed and interactive applications. These softwares use a representation of the macro dataflow, which is central to the MOAIS project, to compute specific schedules of the application tasks. They are designed to be used either independently or coupled.

The MOAIS softwares rely on standard middlewares or, for grid support in particular, on tools developed by other research groups like the MESCAL project.

5.1.1. FlowVR

FlowVR is a middleware library that eases development and execution of virtual reality applications distributed on clusters and grids.

FlowVR supports coupling of heterogeneous parallel codes to build large scale applications. FlowVR reuses and extends the data flow paradigm commonly used for scientific visualization environments. An application is seen as a set of possibly distributed modules exchanging data. Each module endlessly iterates, consuming and producing data. From the FlowVR point of view, modules are not aware of the existence of other modules, as the FlowVR engine takes care of moving data between producers and consumers. This leads to a simple application programming interface that greatly eases porting of an existing program to a FlowVR module (or several modules in case of a parallel code). For data exchange between modules, FlowVR defines an abstract network with advanced features, from simple routing operations to complex message filtering or synchronization operations. Each message is associated with a list of stamps. Stamps are lightweight data used to route or filter messages. This list can also be routed separately from its message to special network nodes in charge of synchronization policies. Different FlowVR networks can be designed without modification or recompilation of the modules.

The FlowVR version 1.2 has been released. It includes several new features like a GUI, support for multiple networks. We also developed on top of FlowVR a distributed rendering framework, called FlowVR-Render, based on shaders that shows a high performance improvement in comparison to existing approaches. This work has been published at IEEE Vis 2005 and released with an Open Source License (library called Flowvr-Render). FlowVR and FlowVR-Render have been used to develop various applications, like real-time 3D modeling (visual-hull and texture) from multiple cameras, ultra-high resolution movie player, 3D visualizer of grid monitoring data (Ganglia and OAR).

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

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 (Symetric 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.

5.2. ClusterVR

5.2.1. The GRIMAGE platform

The MOAIS project is the leader of designing and managing a cluster dedicated to virtual reality applications, the GrImage platform (http://www.inrialpes.fr/grimage). FlowVR is the reference tool for development of applications. The platform will gradually evoluate. A 8-way SMP Itanium base machine is scheduled to be installed on the 10 Gigabit Ethernet Network during Summer 2005. A 8-way SMP machine equipped with dual-core Opteron processors should be installed on the GrImage platform by spring 2006.

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 [20], [36].

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 incertainty and disturbance in classical Combinatorial Optimization problems [53], [23].

6.1.2. Adaptative algorithm

The main results concern the performance prediction of parallel adaptive algorithms [18], [42], [16]; 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 resarch teams in Grenoble and Lyon within the IMAG-INRIA project AHA:

- 3D vision (D Vernizzi, E Boyer, C Menier, B Raffin, JL Roch);
- computer algebra (C Bouillaguet, JG Dumas, T Gautier, JL Roch);
- quadratic assignment problem (VD Cung, T Gautier, S Jafar, JL Roch) [37];
- dynamic deployment on network (G Huard, T Gautier) [26], [38].

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). A master thesis has demonstrate interesting performances on some embedded applications, at the basis of a new collaboration with ST through the MINALOGIC/EmSoc Sceptre project.

6.2. Software

Participants: B. Raffin, T. Gautier.

6.2.1. FlowVR and FlowVR-Render

The FlowVR version 1.2 has been released. It includes several new features like a GUI, support for multiple networks. We also developed on top of FlowVR a distributed rendering framework based on shaders that shows a high performance improvement in comparison to existing approaches. This work has been published at IEEE Vis 200 [29] and released with an Open Source License (library called Flowvr-Render). FlowVR and FlowVR-Render have been used to develop various applications, like real-time 3D modeling (visual-hull and texture) from multiple cameras, ultra-high resolution movie player, 3D visualizer of grid monitoring data (Ganglia and OAR) [27]. The FlowVR technology has also been used to support X-windows on multiple screens [12]. FlowVR is the major contribution of the thesis of Jérémie Allard [9] and enables running large virtual reality applications on the GRIMAGE platform [28], [30].

6.2.2. Fault-tolerancy in KAAPI

We have developed a new algorithm to have a high performance fault tolerant mechanism in KAAPI, with provable performances [38]. This work has received the best paper award at the EIT conference this year [39] . It is based on specializing a communication induced checkpointing algorithm for the work stealing algorithm. The resulting algorithm has a small overhead that can be amortized by adjusting the periodicity of checkpointing a process [37]. Taking benefit of the collaboration with Axel Krings, we also generalize probabilistic certification [52] of a global computation to any massive malicious attacks, both for detection of forgery [41] and certification of results [40]. Those results are currently available on Grid5000.

6.2.3. 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 [20]. 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 [50]; in particular, it has been used for the deployment of a medical application within the RAGTIME Rhône-Alpes project [49] .

6.3. Applications

6.3.1. Code coupling and CAPE applications

The thesis of Hamidi-Reza Hamidi [11] proves the ability of Kaapi to support code coupling on standard interfaces, such as CORBA [21]. With the collaboration of the IFP (Institut Français du Pétrole), we have proposed a prototype that allows CAPE-OPEN compliant application to be executed on cluster. This work has been published [46] 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.

7. Contracts and Grants with Industry

7.1. RNTL project GEOBENCH, 03-05

The RNTL project GEOBENCH associates the MOAIS project, the INRIA action I3D, the LIFO of Université d'Orléans, the CEA, the BRGM and the TGS company. The goal is to develop solutions running on PC clusters for the visualization of (geo)scientific data. Data distribution and computations are supported by Net Juggler. The Amira software from TGS will provide a visualization oriented library for scientific data processing (iso-surfaces extraction for instance). Visualization, more specifically targeting the workbench virtual reality environment (2 L-shaped visualization surfaces), will be associated with haptics interaction. Two classes of applications are considered: applications handling large data sets (CEA application) and applications based on geo-referenced data (BRGM application).

This project is providing funding for 24 months of engineer as well as equipment and travelling.

7.2. RNTL project OCETRE, 04-05

The RNTL project OCETRE associates the MOAIS and MOVI project, the companies Total Immersion and Thalès ST. The goal is to develop solutions for real time 3D reconstruction with a PC cluster and multiple camera acquisition.

This project is providing funding for 24 months of engineer (managed by MOVI) as well as equipment and travelling.

7.3. CIFRE with IFP, 03-06

A collaboration with the company IFP (Institut Français du Pétrole) and the MOAIS project funds a PhD student on code coupling of software components for high performance computing. IFP has worked on the standard CAPE-OPEN which allows to build an application by coupling components. In order to decrease the execution time it should be able to use parallel architectures: the goal of the thesis is to study code coupling methods and scheduling algorithms for these components using the experience of Kaapi.

7.4. CIFRE with ST Microelectronics, 03-06

A PhD thesis involving MOAIS and ST Microelectronics under the terms of a Cifre contract has started in October 2003. The topic of this thesis deals with the problem of large scale instruction scheduling within embedded VLIW processors such as the ST200 model developed by ST Microelectronics. In this context the code produced by the compiler is destined to be directly integrated into some mass-producted embedded device. Thus, the compilation time is negligible compared to the expected performance of the final code. This justify the use of optimal or near-optimal methods for the computation of the instructions schedule even if they are computationally prohibitive. The main issue of this approach is that no current machine can compute an exact resolution of instructions schedule for more than a few hundred instructions. The goal of this thesis is to perform a deep work on the improvement of exact methods as well as a to propose near-optimal approximations of the problem when exact methods cannot be used anymore.

7.5. 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 efficient sequential degeneration). It consists in the specification and implementation in SystemC of a dedicated version of Kaapi software. The validation will be performed based on available emulations of MPSoC platforms.

7.6. CIFRE with Bull, 05-08

Bull is funding a PhD these in collaboration with MOAIS. This PhD is focused on high performance visualization. The work will address performance issues encountered when computing images from large data set to be rendered on display walls. In particular, we will study approaches based on dynamics routing and adpative algorithms. Software developments will be focused on FlowVR, FlowVR Render and VTK. Experiments will mainly use the GrImage platform. We will also considere solutions that are able to take advantage of large SMP and multi-core architectures.

7.7. 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

- RAGTIME project, 03-06: This project targets management of medical informations on the grid. Based on our expertise on macro dataflow scheduling, we are in charge of data access and computations scheduling and also of security issues for the certification of results from remote execution. Partners: LIRIS (Lyon), CREATIS (Lyon), IBCP (Lyon), IN2P3 (Lyon) LIP (Lyon), TIMC-IMAG (Grenoble), ID-IMAG (Grenoble).
- IMAG-INRIA AHA project (05-06). This project targets the design and development of adaptive and hybrid algorithms on parallel computing platforms for applications in arithmetics (finite fields and intervals), exact linear algebra, 3D-reconstruction, combinatorial optimization. Partners: ARENAIRE (Lyon), MOVI, LMC-IMAG and GILCO (Grenoble). We organize an international symposium at SIAM Parallel Processing'06, in San Francisco (2006,February, 22-24).

8.2. National initiatives

- *CYBER II*, 04-06, ACI Masse de Données: the project deals with real time capture, 3D reconstruction and inclusion of a character in a virtual world. Partners : projects MOVI, MOAIS and ARTIS (INRIA Rhône-Alpes) and the LIRIS (Lyon).
- *OTAPHE*, 05-06, ARC INRIA. The project deals with the scheduling of tasks on grid platforms. Partners : GRAAL (Lyon), ALGORILLE (Nancy), MOAIS (Grenoble).
- *BGPR/SAFESCALE*, 05-08, ARA Sécurité: the projects deals with adaptive and safe computations on global computing platforms. Partners: LIPN (Paris XIII), IRISA (Rennes), ENST (Brest), VASCO team (LSR Grenoble), LMC-IMAG and Institut Fourier (Grenoble).
- *GRID'5000*, the french grid platform. MOAIS has participated to the development of the distibuted authentication protocol for Grid5000.

8.3. International initiatives

8.3.1. Foreign office action (MAE and MENESR):

- **Europe:**

  CoreGrid: The project MOAIS participates to the proposal of a Network Of Excellence CoreGrid: workpackages 6 (scheduling) and 4 (fault-tolerance).

  Poland, PAE Polonium: with TU Poznan (J Blazewicz) and Institute of Computer and Information Sciences at Czestochowa University of Technology (R Wyrzykowski). about parallel computation of multiple alignments of genomic sequences and distributed management of caches.

- **Africa:**

  Morocco : with the university of Oujda (Prof. M. Daoudi) about cluster computing (AI MA/01/19 of French-Morocco Committee).

  Morocco : with the university of Rabat (Prof S El Hajji) about security and scientific computing.

  Tunisia : AI Franco-Tunisienne INRIA-DGRSRT with the university of Tunis (Prof M. Jemni) about large scale parallel systems.

8.3.2. North America

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

USA : University of Idaho, Moscow, USA. Axel Krings has been hosted in MOAIS team (09/2004, 08/2005) with support of CNRS and BAC Région Rhône-Alpes.

8.3.3. South America

- 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.
- PICS CNRS CADIGE project with the university federal of Rio Grande do Sul, Brazil (UFRGS) on the programming tools for grid and clusters for virtual reality (2005-2007).
- Capes/cofecub grant obtained with the university federal of Rio Grande do Sup,Brazil (UFRGS), on programming tools forgrid and clusters (2006-2008).
- LAFMI : CICESE Ensenada, Mexico (A Tcherchnyk) on parallel multiple alignments.

8.4. Cluster computing center

8.4.1. The GrImage Platform.

The MOAIS, MOVI, EVASION and ARTIS projects are collaborating to install and operate at the INRIA Rhône-Alpes an experimental plateform for high performance interactive applications (the GrImage platform).

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. This display wall is built to enable stereoscopic projection using passive stereo. The main goal is to provide a visualization space for large models and real time interaction.

GrImage will enable to perform research in the following areas: Real time 3d reconstruction; High performance graphics rendering; Virtual and augmented reality; Distributed resource allocation for interactive applications; Scientific visualization; Interaction and visualization for the grid; Calibration and low level synchronizations.

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

9. Dissemination

9.1. Leadership within scientific community

- Program committees :

  - PARELEC'04 (Fourth International Conference on Parallel Computing in Electrical Engineering) - Dresden, Germany (sept. 2004)
  - ENC'2004 (Mexican International Conference on Computer Science) - Mexico (sept. 2004)
  - SBAC-PAD 2004 (16th Symposium on Computer Architecture and High-Performance Computing) - Sao Paulo, Brazil (Oct. 2004)
  - AICCSA-05 (ACS/IEEE International Conference on Computer Systems and Applications) - Cairo, Egypt (january 2005)
  - HCW'05 (Heterogeneous Computing Workshop) - Denver, USA (april 2005)
  - MISTA'05 (2nd Multidisciplinary International Conference on Scheduling : Theory and Applications) - New York, USA (july 2005)
  - RENPAR'16, Le Croisic, France (Avril 2005)
  - Workshop on scheduling and load-balancing EuroPar'2005 - Lisboa, Portugal (Aout 2003)
  - PARCO'2005 (Parallel Computing) - Malaga, Spain (september 2005)
  - PPAM'05 (Sixth International Conference on Parallel Processing and Applied Mathematics) - Poznan, Poland (sept. 2005)
  - HPCC'05 (International Conference on High Performance Computing and Communications) - Sorento, Italy (sept. 2005)
  - ENC'05 (Mexican International Conference on Computer Science) Puebla, Mexico (sept. 2005)

  - SBAC-PAD 2005 (17th Symposium on Computer Architecture and High-Performance Computing) - Rio de Janeiro, Brazil (Oct. 2005)
  - HiPC'05 (12th Annual International Conference on High Performance Computing) - Goa, India (december 2005)

- 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).

10. Bibliography

Major publications by the team in recent years

[1] J. ALLARD, C. MÉNIER, E. BOYER, B. RAFFIN. *Running Large VR Applications on a PC Cluster: the FlowVR Experience*, in "Proceedings of EGVE/IPT 05, Denmark", October 2005.

[2] E.-M. DAOUDI, T. GAUTIER, A. KERFALI, R. REVIRE, J.-L. ROCH. *Algorithmes parallèles à grain adaptatif et applications*, in "Technique et Science Informatiques (TSI)", vol. 24, 2005, p. 1–20.

[3] 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.

[4] 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.

[5] C. MARTIN, O. RICHARD, G. HUARD. *Déploiement adaptatif d'applications parallèles*, in "Technique et Science Informatiques (TSI)", vol. 24, 2005.

Books and Monographs

[6] J. BLAZEWICZ, K. ECKER, D. TRYSTRAM. *Feature cluster on recent advances on scheduling in computer and manufacturing systems*, vol. 64, n^o 3, Elsevier, 2005.

[7] S. EL HAJJI, A. HILALI, J.-L. ROCH. *Actes de la conférence "Cryptologie et Applications" - 2003*, Université Mohamed II, Rabbat, Maroc, September 2004.

[8] D. TRYSTRAM, Y. SLIMANI, M. JEMNI. *Informatique répartie*, Hors serie de la revue des Sciences et Technologies de l'Information, Hermes, Lavoisier, 2005.

Doctoral dissertations and Habilitation theses

[9] J. ALLARD. *FlowVR: calculs interactifs et visualisation sur grappe*, Thèse de Doctorat, spécialité informatique, Institut National Polytechnique de Grenoble, Novembre 2005.

[10] L.-A. Estefanel. *LaPIe: Communications Collectives Adaptées aux Grilles de Calcul*, Thèse de Doctorat, spécialité informatique, Institut National Polytechnique de Grenoble, December 2005.

[11] H.-R. Hamidi. *Couplage à hautes performances de codes parallèles et distribués*, Thèse de Doctorat, spécialité informatique, Institut National Polytechnique de Grenoble, November 2005.

[12] J. Verduzco. *Environnement X Window pour mur d'images*, Thèse de Doctorat, spécialité informatique, Institut National Polytechnique de Grenoble, June 2005.

[13] J. Zola. *Parallel Server for Multiple Sequence Alignment*, Thèse de Doctorat, spécialité informatique, Institut National Polytechnique de Grenoble, December 2005.

Articles in refereed journals and book chapters

[14] C. Bardel, V. Danjean, J.-P. Hugot, P. Darlu, E. Génin. *On the use of haplotype phylogeny to detect disease susceptibility loci*, in "BMC Genetics", vol. 6, n^{o} 24, 2005.

[15] O. Beaumont, V. Boudet, P. Dutot, Y. Robert, D. Trystram. *Chapitre 3. Gestion de ressources*, D. Trystram, Y. Slimani, M. Jemni (editors). , Hors serie TSI, chap. 3, Hermes, 2005.

[16] J. Blazewicz, M. Kovalyov, M. Machowiak, D. Trystram, J. Weglarz. *Preemptable malleable task scheduling problem*, in "IEEE Transactions on Computers", to appear 2005.

[17] V. Danjean, P.-A. Wacrenier. *Mécanismes de traces efficaces pour programmes multithreadés*, in "TSI", To appear, 2005.

[18] E.-M. Daoudi, T. Gautier, A. Kerfali, R. Revire, J.-L. Roch. *Algorithmes parallèles à grain adaptatif et applications*, in "Technique et Science Informatiques", vol. 24, 2005, 1—20.

[19] A. Darte, G. Huard. *New Complexity Results on Array Contraction and Related Problems*, in "Journal on VLSI of Signal Processing-Systems for Signal, Image and Video Technology", vol. 40, n^{o} 1, 2005, p. 35-55.

[20] 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.

[21] T. Gautier, H. Hamidi. *Re-scheduling invocations of services on RPC-based Grid*, in "In International Journal Computer Languages, Systems and Structures (CLSS)", à paraître, 2005.

[22] A. Goldman, J. Peters, D. Trystram. *Exchanging messages of different size*, in "Journal of Parallel and Distributed Computing", to appear 2005.

[23] A. Goldman, D. Trystram. *An efficient parallel algorithm for solving the Knapsack problem on hypercubes*, in "Journal of Parallel and Distributed Computing", n^{o} 64, 2004, p. 1213-1222.

[24] D. HAGIMONT, D. LITAIZE, Z. MAHJOUB, D. TRYSTRAM. *Chapitre 1. Introduction*, D. TRYSTRAM, Y. SLIMANI, M. JEMNI (editors). , Hors serie TSI, Hermes, 2005.

[25] A. MAHJOUB, D. TRYSTRAM. *Stabilisation pour des applications parallèles*, J. BILLAUT, A. MOUKRIM, E. SANLAVILLE (editors). , Informatique et Systèmes d'Information, Hermes, Lavoisier, 2005.

[26] C. MARTIN, O. RICHARD, G. HUARD. *Déploiement adaptatif d'applications parallèles*, in "Technique et Science Informatiques", vol. 24, 2005.

Publications in Conferences and Workshops

[27] J. ALLARD, J.-S. FRANCO, C. MÉNIER, E. BOYER, B. RAFFIN. *The GrImage Platform: A Mixed Reality Environment for Interactions*, in "IEEE International Conference on Computer Vision Systems, New York", to appear, January 2006.

[28] J. ALLARD, C. MÉNIER, E. BOYER, B. RAFFIN. *Running Large VR Applications on a PC Cluster: the FlowVR Experience*, in "Proceedings of EGVE/IPT 05, Denmark", October 2005.

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

[30] J. ALLARD, B. RAFFIN. *Distributed Physical Based Simulations for Large VR Applications*, in "IEEE IEEE Virtual Reality Conference, Alexandria, USA", to appear, March 2006.

[31] L. A. BARCHET-ESTEFANEL, G. MOUNIÉ. *Performance Characterisation of Intra-Cluster Collective Communications*, in "Proc. of the 16th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2004), Foz do Iguaçu, Brazil, 27-29 October 2004", IEEE Computer Society/Brazilian Computer Society, 2004, p. 254–261.

[32] L. A. BARCHET-ESTEFANEL, G. MOUNIÉ. *Prédiction de Performances pour les Communications Collectives*, in "Proceedings du 16ème Rencontre Francophone du Parallélisme (RenPar'16), Le Croisic, France, 6-8 April 2005", ACM SIGOPS France, 2005, p. 101–112.

[33] L. A. BARCHET-ESTEFANEL, G. MOUNIÉ. *Total Exchange Performance Modelling under Network Contention*, in "Proceedings du 6th International Conference on Parallel Processing and Applied Mathematics (PPAM 2005), Poznan, Poland, 11-14 September 2005", Springer Verlag, 2005, To be published.

[34] O. BEAUMONT, E. DAOUDI, N. MAILLARD, P. MANNEBACK, J.-L. ROCH. *Tradeoff to minimize extra-computations and stopping criterion tests for parallel iterative schemes*, in "3rd International Workshop on Parallel Matrix Algorithms and Applications (PMAA04), CIRM, Marseille, France", October 2004.

[35] N. CAPIT, G. DA-COSTA, Y. GEORGIOU, G. HUARD, C. MARTIN, G. MOUNIÉ, P. NEYRON, O. RICHARD. *A batch scheduler with high level components*, in "Cluster Computing and Grid Proceedings", 2005, 143.

[36] P. DUTOT, D. TRYSTRAM. *A best-compromise bicriteria scheduling algorithm for parallel tasks*, in "Proceedings of WEA'05 (4th International Workshop on Efficient and Experimental Algorithms, Santorini Island, Greece", Poster, 2005.

[37] S. JAFAR, T. GAUTIER, A. W. KRINGS, J.-L. ROCH. *A Checkpoint/Recovery Model for Heterogeneous Dataflow Computations Using Work-Stealing*, in "EUROPAR'2005, Lisboa, Portogal", Springer-Verlag, LNCS, August 2005.

[38] S. JAFAR, T. GAUTIER, J.-L. ROCH. *Modèle de Coût Algorithmique Intégrant des Mécanismes de Tolérance aux Pannes et Expérimentations*, in "Proceedings des 16emes rencontres francophones du parallélisme (RenPar'16), Le Croisic, France", April 2005, p. 125-136.

[39] 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.

[40] A. W. KRINGS, J.-L. ROCH, S. JAFAR. *Certification of Large Distributed Computations with Task Dependencies in Hostile Environments*, in "IEEE Electro/Information Technology Conference , (EIT 2005), Lincoln, Nebraska", IEEE, May 2005.

[41] A. W. KRINGS, J.-L. ROCH, S. JAFAR, S. VARRETTE. *A Probabilistic Approach for Task and Result Certification of Large-scale Distributed Applications in Hostile Environments*, in "European Grid Conference EGC'2005, Amsterdam, The Netherlands", Springer-Verlag, LNCS, February 2005.

[42] L. MASKO, G. MOUNIÉ, M. TUDRUJ, D. TRYSTRAM. *Moldable Tasks scheduling in dynamic SMP clusters with communications on the fly*, in "PARELEC, International Conference on Parallel Computing for Electrical Engineering, Dresden, Germany", september 2004.

[43] G. PARMENTIER, D. TRYSTRAM, J. ZOLA. *Cache-based parallelization of multiple alignment problem*, in "10th international EUROPAR conference, Pisa, Italy", M. DANELUTTO, D. LAFORENZA, M. VANNESCHI (editors). , LNCS, n^{o} 3149, Springer Verlag, september 2004.

[44] J. PECERO-SANCHEZ, D. TRYSTRAM. *A new Genetic Convex Clustering algorithm for parallel time minimization with large communication delays*, in "International Conference Parallel Computing, ParCo'2005, Malaga, Spain", september 2005.

[45] J. PECERO-SANCHEZ, D. TRYSTRAM. *Convex Clustering algorithm for makespan minimization with large communication delays*, in "6th workshop on Models and Algorithms for Planning and Scheduling Problems, MAPSP'2005, Siena, Italy", June 2005.

[46] L. PIGEON, L. TESTARD, T. GAUTIER, P. ROUX. *Towards A Distributed High Performance Computing Environment For CAPE-OPEN Standard*, in "In Proceedings of 7th World Congress of Chemical Engineering, Glasgow, Scotland", 2005.

[47] D. TRYSTRAM. *Efficient algorithms for scheduling the tasks of Parallel Programs*, in "CSC05 (second workshop on Combinatorial Scientific Computing), Toulouse, France", invited talk, june 2005.

[48] D. TRYSTRAM, J. ZOLA. *Parallel multiple sequence algnment with decrentralized Cache support*, in "EUROPAR'05, Lisboa, Portugal", LNCS, n^{o} 3648, Springer, august 2005, p. 1217-1226.

[49] S. VARRETTE, S. GEORGET, J. MONTAGNAT, J.-L. ROCH, F. LEPRÉVOST. *Distributed Authentication in*

GRID5000, in "LNCS OnTheMove Federated Conferences - Workshop "Grid Computing and its Application to Data Analysis (GADA'05)", Agia Napa, Cyprus", LNCS, Springer Verlag, November 2005.

[50] S. VARRETTE, S. GEORGET, J.-L. ROCH, F. LEPRÉVOST. *Authentification Distribuée sur Grille de Grappes basée sur LDAP*, in "Proceedings des 16èmes rencontres francophones du parallélisme (RenPar'16), Le Croisic, France", ASF - Ecole des Mines de Nantes, April 2005.

[51] S. VARRETTE, J.-L. ROCH, Y. DENNEULIN, F. LEPRÉVOST. *Secure architectures for clusters and grids*, in "CRIS 2004, Grenoble, France", IEEE, October 2004.

[52] S. VARRETTE, J.-L. ROCH, F. LEPRÉVOST. *FlowCert: Probabilistic Certification for Peer-to-Peer Computations*, in "16th Symposium on Computer Architecture and High Performance Computing, IEEE SBAC-PAD 2004, Foz do Iguacu, Brazil", IEEE, October 2004, p. 108–115.

Miscellaneous

[53] G. MOUNIÉ, D. TRYSTRAM. *Dealing with uncertainty in scheduling algorithmsSpring school on design and analysis of algorithms, Hangzhou, China*, 2005.

[54] D. TRYSTRAM. *Le calcul à hautes performances*, n^o 26, février 2005, Tangente-sup.