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

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 Associate (CR1)]
Bruno Raffin [Research Associate CR1]

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

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

UPMF Staff
Pierre-François Dutot [Assistant Professor]

Invited Scientist
Andreï Tchernyk [CICESE, Ensenada, Mexico, 1 month]
Michael Bender [Stony Brook University, New York, 2 weeks]

Postdoc
Luciano Soares [1 year]
Fanny Pascual [1 year]

Engineers
Serge Guelton [1 year]
Liyun He [1 year]

PhD students
Thomas Arcila [2005, CIFRE Bull]
Julien Bernard [2005, BDI CNRS / ST Microelectronics scholarship]
Florent Blachot [2003, CIFRE ST Micro Electronics scholarship]
Lionel Eyraud [2002, Normalien, MRNT scholarship]
Samir Jafar [2002, Syrian scholarship]
Clément Ménier [2003, Normalien, common to PERCEPTION 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]
Jaroslaw Zola [2002, Polish scholarship, co-tutelle]
Everton Hermann [2006, INRIA Cordi]
Jean-Denis Lesage [2006, MRNT scholarship]
Xavier Besseron [2006, MRNT scholarship]
Floran Dietrich [2006, co-tutelle]
Gerald Veisman [2006, CIFRE DCN]

Adel Essafi [2006, co-tutelle]
Sami Achour [2006, co-tutelle]
Yannick N'Goko [2006, 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: P.F. Dutot, T. Gautier, G. Huard, G. Mounié, J.-L. Roch, D. Trystram, F. Wagner.

The goal of this theme is to determine adequate multi-criteria objectives which are efficient (precision, reactivity, speed) and to study scheduling algorithms to reach these objectives.

In the context of parallel and distributed processing, the term *scheduling* is used with many acceptations. In general, scheduling means assigning tasks of a program (or processes) to the various components of a system (processors, communication links).

Researchers within MOAIS have been working on this subject for 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. Adaptive Parallel and Distributed Algorithms Design

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

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

This theme deals with the analysis and the design of algorithmic schemes that control (statically or dynamically) the grain of interactive applications.

The classical approach consists in setting in advance the number of processors for an application, the execution being limited to the use of these processors. This approach is restricted to a constant number of identical resources and for regular computations. 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 various combinatorial optimization problems (QAP problems with PRISM laboratory, telecommunication application with LIFL, nqueens ProActive Challenge, ...)) 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, P.F. Dutot, T. Gautier, B. Raffin, J.-L. Roch.

The goal of this theme is to develop approaches to tackle interactivity in the context of large scale distributed applications.

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).

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.

3.4. Adaptive middleware for code coupling and data movements

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

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

This theme deals with the design and implementation of programming interfaces in order to achieve an efficient coupling of distributed components.

The implementation of interactive simulation application requires to assemble together various software components and to ensure a semantic on the displayed result. To take into account functional aspects of the computation (inputs, outputs) as well as non functional aspects (bandwidth, latency, persistence), elementary actions (method invocation, communication) have to be coordinated in order to meet some performance objective (precision, quality, fluidity, *etc*). In such a context the scheduling algorithm plays an important role to adapt the computational power of a cluster architecture to the dynamic behavior due to the interactivity. Whatever the scheduling algorithm is, it is fundamental to enable the control of the simulation. The purpose of this research theme is to specify the semantics of the operators that perform components assembling and to develop a prototype to experiment our proposals on real architectures and applications.

3.4.1. Application Programming Interface

The specification of an API to compose interactive simulation application requires to characterize the components and the interaction between components.The respect of causality between elementary events ensures, at the application level, that a reader will see the *last* write with respect to an order. Such a consistency should be defined at the level of the application in order to control the events ordered by a chain of causality. For instance, one of the result of Athapascan was to prove that a data flow consistency is more efficient than other ones because it generates fewer messages. Beyond causality based interactions, new models of interaction should be studied to capture non predictable events (delay of communication, capture of image) while ensuring a semantic.

Our methodology is based on the characterization of interactions required between components in the context of an interactive simulation application. For instance, criteria could be coherency of visualization, degree of interactivity. Beyond such characterization we hope to provide an operational semantic of interactions (at least well suited and understood by usage) and a cost model. Moreover they should be preserved by composition in order to predict the cost of an execution for part of the application.

This work is based on the experience of the APACHE project and the collaborative research actions ARC SIMBIO and ARC COUPLAGE. The main result relies on a computable representation of the future of an execution; representations such as macro data flow are well suited because they explicit which data are required by a task. Such a representation can be built at runtime by an interpretation technique: the execution of a function call is differed by prealably computing at runtime a graph of tasks that represents the (future) calls to execute. Based on this technique, Athapascan, the language developed by the APACHE project, enables to write a single program for both the code to execute and the description of the future of the execution.

3.4.2. Kernel for Asynchronous, Adaptive, Parallel and Interactive Application

Managing the complexity related to fine grain components and reaching high efficiency on a cluster architecture require to consider a dynamic behavior. Also, the runtime kernel is based on a representation of the execution: data flow graph with attributes for each node and efficient operators will be the basis for our software. This kernel has to be specialized for considered applications. The low layer of the kernel has features to transfer data and to perform remote signalization efficiently. Well known techniques and legacy code have to be reused. For instance, multithreading, asynchronous invocation, overlapping of latency by computing,

parallel communication and parallel algorithms for collective operations are fundamental techniques to reach performance. Because the choice of the scheduling algorithm depends on the application and the architecture, the kernel will provide an *causally connected representation* of the system that is running. This allows to specialize the computation of a good schedule of the data flow graph by providing algorithm (scheduling algorithm for instance) that compute on this (causally connected) representation: any modification of the representation is turned into a modification on the system (the parallel program under execution). Moreover, the kernel provides a set of basic operators to manipulate the graph (*e.g.* computes a partition from a schedule, remapping tasks, ...) to allow to control a distributed execution.

4. Application Domains

4.1. Virtual Reality

Participants: T. Arcila, T. Gautier, J.-D. Lesage, B. Raffin, L. Soares, J.-L. Roch.

We are pursuing and extending existing collaborations to develop virtual reality applications on PC clusters and grid environments. This work mainly relies on FlowVR. Different actions are be considered:

- 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 have been developping approaches for an exact surface based algorithm and a volume base algorithm. This year we also manage to texture the models in real time. A small cluster with 4 cameras was installed at IEEE VR 2006. This demo enabled users to interact in real time with virtual objects using their hands.
- Interactive visualization of AMR (Adaptive Mesh Refinment) Data Sets. This on going projects focuses on developping new algorithms to render large AMR at interactive frme rates. This work is acheived in collaboration with BULL and DEA DAM.

4.2. Code Coupling

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

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

- 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.

4.3. Secure Computations

Participants: T. Gautier, J.-L. Roch, S. Jafar, S. Varette.

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 presented in June 2006);
- 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).

4.4. Embedded Systems

Participants: J.-L. Roch, G. Huard, D. Trystram, V. Danjean, G. Veisman, D. Traore, J. Bernard, F. Blachot.

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 (J. Bernard thesis) obtained on MPEG-4 encoding and Lempel-Zif compression, are very promising. This work recently received additionals fundings (BDI grant 10/2005 - 10/2008 cofunded by ST and CNRS) and is developped in the framework of the contract SCEPTRE inside the regional competitivity pole MINALOGIC/EMSOC.

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 integrate 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).

4.5. Genomic – Multiple Alignments with Tree Construction

Participant: D. Trystram.

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.

5. Software

5.1. FlowVR

Participants: T. Arcila, C. Ménier, J-D. Lesage, B. Raffin [correspondant].

The goal of the **FlowVR** library is to provide users with the necessary tools to develop and run high performance interactive applications on PC clusters and Grids. The main target applications include virtual reality and scientific visualization. FlowVR enforces a modular programming that leverages software engineering issues while enabling high performance executions on distribued and parallel architectures.

The FlowVR software suite has today 3 main components:

- **FlowVR**: The core middleware library. FlowVR relies on the data-flow oriented programming approach that has been successfully used by other scientific visualization tools. Developing a FlowVR application is a two step process. First, modules are developed. Modules encapsulate a piece of code, imported from an existing application or developed from scratch. The code can be a multi-threaded or parallel, as FlowVR enables parallel code coupling. In a second step, modules are mapped on the target architecture and assembled into a network to define how data are exchanged. This netwok can make use of advanced features, from simple routing operations to complex message filtering or synchronization operations.
- **FlowVR Render**: A parallel rendering library. FlowVR Render proposes a framework to take advantage of the power offered by graphics clusters to drive display walls or immersive multi-projector environments like Caves. It relies on an original approach making an intensive use of hardware shaders. FlowVR Render comes with a port of the MPlayer Movie Player. This will enable you to play movies on your favorite 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. By November 2006, it has been downloaded 925 times. The publication [23] gives a good overview of the full Software suite and the different applications it has been used for.

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

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

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

The performance of applications on top of Kaapi scales on clusters and large SMP machines (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.

5.3. TakTuk - Adaptive large scale remote execution deployment

Participant: G. Huard [corespondant].

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

Characteristics:

- adaptivity: efficient work distribution is achieved even on heterogeneous platforms thanks to an adaptive work-stealing algorithm
- scalability TakTuk has been tested to perform large size deployments (hundreds of nodes), either on SMPs, regular clusters or clusters of SMPs
- portability: TakTuk is architecture independent (tested on x86, PPC, IA-64) and distinct instances can communicate whatever the machine they're running on
- configurability: mechanics are configurable (deployment window size, timeouts, ...) and TakTuk outputs can be suppressed/formatted using I/O templates

Outstanding features:

- autopropagation: the engine can spread its own code to remote nodes in order to deploy itself
- communication layer: nodes successfully deployed are numbered and perl scripts executed by TakTuk can send multicast communication to other nodes using this logical number
- informations redirection: I/O and commands status are multiplexed from/to the root node.

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

6. New Results

6.1. Parallel algorithms, complexity and scheduling

6.1.1. Scheduling

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

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

6.1.2. Adaptive algorithm

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

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

The runtime behavior is mainly based on the workstealing scheduling algorithm of Kaapi. Within a collaboration with ST, Kaapi is currently ported on MPSoCs (MultiProcessorS on Chips). During his first year of thesis, Julien Bernard has demonstrated importance of the adaptive scheme for some embedded applications (data compression), at the basis of a new collaboration with ST through the MINALOGIC/EmSoc Sceptre project.

As a result, Moais has particpated to a classification of hybrid algorithms, focusing on adaptive algorithms with provable performances. This classification has been presented in a mini-symposium at the international conference Parallel Processing 2006.

6.1.3. Adaptive Octree for interactive 3D modelling

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

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

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

6.1.4. Adaptive parallel prefix computation

Parallel prefix is a famous folk scheme in parallel programming of great practical importance. For this problem, even if fine grain fast parallel algorithms are known, decreasing the (parallel) time requires to increase the number of operations to perform and thus the load of the system. Based on an on-line coupling of a sequential algorithm and a recursive extraction of parallelism by work-stealing, a near-optimal parallel prefix algorithm has been exhibited on p processors with changing speeds (Europar 2006, PhD thesis of Daouda Traore). Experimentations on SMP and small size distributed architectures have exhibit its good practical performance and interest.

We are currently extending this scheme to provide a near optimal parallel algorithm which does not use the number p of processors. Such algorithms are called *processor oblivious*. We believe that the scheme is generic and could be applied to a wide spectrum of problems, especially in the context of the ANR Safescale and MINALOGIC Spectre contracts.

6.2. Software

6.2.1. FlowVR Suite

A new version of the full FlowVR suite has been released. The main new feature is a graphical user interface for visualizing and handling the graph of a FlowVR application. This new release also includes a OpenGL wrapper enabling to render on a display wall any OpenGL application. Several 2006 plublications are based on FLowVR [23], [22], [21].

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.This work has received the best paper award at the EIT conference this year. 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. we also generalize probabilistic certification of a global computation to any massive malicious attacks, both for detection of forgeryand certification of results. Those results are currently available on Grid5000. and have been used in 2006 within the Rhône-Alpes project Ragtime. Those works are currently extended to certification of computer algebra and sorting applications within the ANR SAFESCALE/BGPR.

6.2.3. Scalability of KAAPI

KAAPI software has been tested on whole Grid50000 during the 3rd PLUGTEST event organised by ETSI and project OASIS at Sophia-Antipolis, 27-30 november 2006. The implemented workstealing algorithm has demonstrated its capacity to scheduling fine grain program on 1422 processors using two level scheduling strategy: a theif tries to steal work first from a thread running on the same process. In case of failure, the theif emits steal request to an other process. The overall efficiency is very good : on a running time of $13305s$, the average inactivity of process is $83s$. The KAAPI team took part of the NQueens contest during the PLUGTEST event and was the winner in front of several teams (China, Japan, Netherlands, Poland, Chile, Brazil).

6.2.4. GRID5000: scheduling algorithm for OAR and authentication

OAR is a batch scheduler developed by Mescal team. The MOAIS team develops the central automata and the scheduling module that includes successive evolutions and improvements of the policy.OAR is used to schedule jobs both on the CiGri (Grenoble region) and Grid50000 (France) grids. CiGri is a production grid that federates about 500 heterogeneous resources of various Grenoble laboratories to perform computations in physics. MOAIS has also developed the distributed authentication for access to Grid5000; in particular, it has been used for the deployment of a medical application within the RAGTIME Rhône-Alpes project.

6.3. Applications

6.3.1. Code coupling and CAPE applications

The thesis of Hamidi-Reza Hamidi proves the ability of Kaapi to support code coupling on standard interfaces, such as CORBA. 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. 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. 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.

7.2. 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.

7.3. 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.

7.4. 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 dynamic 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.

7.5. Contract with DCN, 05-08

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

7.6. Contract SCEPTRE (leader STMicroelectronics, 06-09)

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

8. Other Grants and Activities

8.1. Regional initiatives

- RAGTIME project, 03-06: This project targets management of medical information on the grid. Based on our expertise on macro dataflow scheduling, we have been in charge of data access and computations scheduling and also of security issues for the certification of results from remote execution. A demonstration (mammography analysis based on comparisons with images stored in huge databases such as PACS) has been built on top of Grid5000, in collaboration with CREATIS (J Montagnat, μ-grid software) and LIRIS (JM Pierson, management of right access, and S Miguet, medical image comparison); probabilistic result certification is based on Kaapi software. 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 organized an international symposium at SIAM Parallel Processing'06, in San Francisco (2006,February, 22-24).

8.2. National initiatives

- *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).
- *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. Since october 2006, Serge Guelton has been recruited as an engineer on this contract. A version of Kaapi has been provided with documentation to partners of the contract. 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 (Toulous), MOAIS (Grenoble)
- *GRID'5000*, the french grid platform. MOAIS has participated to the development of the distibuted authentication protocol for Grid5000 (namely deployment with TakTuk, scheduling policies in OAR and distributed authentication based on LDAP).

8.3. International initiatives

8.3.1. 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 grids 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).
- Equipe associée INRIA Diode-A with the universities of the state of Rio Grande do Sul, on the programming tools for grid and clusters for virtual reality (2006-2008).
- LAFMI : CICESE Ensenada, Mexico (A Tcherchnyk) on parallel multiple alignments.

8.4. Hardware Platforms

8.4.1. The GRIMAGE platform

The GrImage platform (http://www.inrialpes.fr/grimage) gathers a 16 projector display wall, a network of cameras and a PC cluster. It is dedicated to interactive applications. GrImage is co-leaded by the Moais and Perception projects (participants are the MOAIS, PERCEPTION, EVASION and ARTIS projects). It is the milestone of a strong and fruitfull collaboration between Moais and Perception (common publications, software and application development).

GrImage (Grid and Image) aggregates commodity components for high performance video acquisition, computation and graphics rendering. Computing power is provided by a PC cluster, with some PCs dedicated to video acquisition and others to graphics rendering. A set of digital cameras enables real time video acquisition. The main goal is to rebuild in real time a 3D model of a scene shot from different points of view. A display wall built around commodity video projectors provides a large and very high resolution display. The main goal is to provide a visualization space for large data sets and real time interaction.

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

8.4.2. SMP Machines

MOAIS and MESCAL have invested in 2005 on two SMP architectures:

- A 8-way SMP machine equiped with Itanium processors.
- A 8-way SMP machine equiped with dual core processors (total of 16 cores) and 2 GPUs. This machine is connected on the 10 Gigabit Ethernet backbone connecting the Icluter-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.

9. Dissemination

9.1. Leadership within scientific community

- Conference Chair:
  - EGPGV 2006 (Eurographics Symposium on Parallel Graphics and Visualization) - Braga Portugal (May 2006)
- Program committees :
  - comité de programme du workshop on Grid and Parallel Computing, American University of Beirut, Libanon (january 2006)
  - comité d'organisation de 12th SIAM Conference on Parallel Processing for Scientific Computing, San Francisco, USA (february 2006)
  - comité de programme de IPDPS 2006 (Seventeenth International Parallel and Distributed Processing Symposium) - Rhodos, Greece (april 2006)
  - comité de programme de HCW'06 (Heterogeneous Computing Workshop) - Rhodos, Greece (april 2006)

- comité de programme de PMAA (Parallel Matrix Algorithms and Applications), Rennes, France (sept. 2006)
- co-responsable du Workshop on Application of high-performance and Grid computing, EuroPar'2006 - Dresden, Germany (August 2006)
- comité de programme de ENC'06 (Mexican International Conference on Computer Science) San Luis Potosi, Mexico (sept. 2006)
- comité de pilotage de HETEROPAR 2006, Barcelona, Spain (sept. 2006)
- comité de programme de RENPAR'17 (17ièmes Rencontres Francophones du Parallélisme), Perpignan, France (octobre 2006)
- comité de programme de SBAC-PAD 2006 (18th Symposium on Computer Architecture and High-Performance Computing) - Ouro Preto, Brazil (oct. 2006)
- comité de programme de IA 2006 (Informatique et Apports Applicatifs), Oujda, Maroco (31 oct - 2 nov 2006)
- comité de programme de ICCES'06 (International Conference on Computer Engineering and Systems) - Cairo, Egypt (november 2006)
- comité de programme de HiPC'06 (13th Annual International Conference on High Performance Computing) - Bangalore, India (december 2006)
- SIAM Parallel Processing, PP'06 conference: organization of the mini-Symposium MS1: Adaptive algorithms for scientific computing. San Francisco, USA (Feb. 2006)
- Transgressive Computing'06 (Fifth International Workshop on Algorithms, Models conference in honor of Jean Della Dora) Granada, Spain (Apr. 2006)
- Europar'06 (European Conference on Parallel Computing) Dresden (Aug. 2006)
- ICON 2006 (14th IEEE International Conference on Networks) - Singapore (Sep. 2006) High Performance Computing, Ouro Pretto, Brazil (Oct. 2006)
- Special Issue of Parallel Computing Journal on Large Scale Grid, Elsevier Parco (Nov. 2006)

- 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).
- Lecturer for the First International Summer School on Emerging Trends in Concurrency (TiC'06) ,Bertinoro, Italia.

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, http://tsi.revuesonline.com/.

[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° 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, http://www.nuengr.unl.edu/eit2005/.

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

Year Publications

Doctoral dissertations and Habilitation theses

[6] L. EYRAUD. *Théorie et pratique de l'ordonnancement d'applications sur les systèmes distribués*, Ph. D. Thesis, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE - INPG, October 2006.

[7] S. JAFAR. *Programmation des systémes parallèles distribuées : tolérance aux pannes, résilience et adaptabilité*, Ph. D. Thesis, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE - INPG, July 2006.

Articles in refereed journals and book chapters

[8] L. A. BARCHET-STEFFENEL, G. MOUNIÉ. *Total Exchange Performance Modelling Under Network Contention*, in "Parallel Processing and Applied Mathematics, 6th International Conference, PPAM 2005, Poznan, Poland, September 11-14, 2005, Revised Selected Papers", R. WYRZYKOWSKI, J. DONGARRA, N. MEYER, J. WASNIEWSKI (editors). , Lecture Notes in Computer Science (LNCS), vol. 0, n° 3911, Springer, 2006, p. 100–107.

[9] C. BARDEL, V. DANJEAN, E. GÉNIN. *ALTree: association detection and localization of susceptibility sites using haplotype phylogenetic trees*, in "Bioinformatics", vol. 11, n° 22, April 2006, http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btl131v1.

[10] J. BLAZEWICZ, M. KOVALYOV, M. MACHOWIAK, D. TRYSTRAM, J. WEGLARZ. *Preemptable malleable task scheduling problem*, in "IEEE Transactions on Computers", vol. 55, n° 4, 2006, p. 486-490.

[11] J. BLAZEWICZ, M. Y. KOVALYOV, M. MACHOWIAK, D. TRYSTRAM, J. WEGLARZ. *Preemptable Malleable Task Scheduling Problem.*, in "IEEE Trans. Computers", vol. 55, n° 4, 2006, p. 486-490.

[12] P. BOUVRY, J.-G. DUMAS, R. GILLARD, J.-L. ROCH, S. VARRETTE. *Chapitre 2 – Cryptographie à clef secréte*, in "La sécurité Multimédia", T. EBRAHIMI, F. LEPRÉVOST, B. WARUSFELD (editors). , Hermes Science, January 2006, p. 23–99.

[13] C. BRIZUELA, L. GONZALEZ-GURROLA, A. TCHERNYKH, D. TRYSTRAM. *Sequencing by hybridization: an enhanced crossover operator for a hybrid genetic algorithm*, in "Journal of Heuristics", 2006.

[14] J. COHEN, E. JEANNOT, N. PADOY, F. WAGNER. *Messages Scheduling for Parallel Data Redistribution between Clusters*, in "IEEE Trans. Parallel Distrib. Syst.", vol. 17, n° 10, 2006, p. 1163–1175, http://dx.doi.org/10.1109/TPDS.2006.141.

[15] J.-G. DUMAS, F. LEPRÉVOST, J.-L. ROCH, V. SAVIN, S. VARRETTE. *Chapitre 4 – Cryptographie à clef publique*, in "La sécurité Multimédia", T. EBRAHIMI, F. LEPRÉVOST, B. WARUSFELD (editors). , Hermes Science, January 2006, p. 103–182.

[16] J.-G. DUMAS, F. LEPRÉVOST, J.-L. ROCH, S. VARRETTE. *Chapitre 5 – Architectures PKI*, in "La sécurité Multimédia", T. EBRAHIMI, F. LEPRÉVOST, B. WARUSFELD (editors). , to appear in Hermes, January 2006, p. 187–208.

[17] L. EYRAUD. *A pragmatic analysis of scheduling environments on new computing platforms*, in "Intl. Journal of High Performance Computing and Applications", 2006.

[18] D. NADDEF, E. SAULE, D. TRYSTRAM. *Le Simplexe : une approche géométrique de la Programmation Linéaire*, in "Tangente Sup", n° 32, juillet-aout 2006, p. 10-12.

[19] W. NASRI, D. TRYSTRAM, S. ACHOUR. *Adaptive algorithms for the parallelization of the dense matrix multiplication on clusters*, in "Internat. J. of Computational Science and Engineering, Special Issue on best selected papers of PDSEC'04", 2006.

[20] G. PARMENTIER, D. TRYSTRAM, J. ZOLA. *Large scale multiple sequence alignment with simultaneous phylogeny inference*, in "Journal of Parallel and distributed computing", 2006.

Publications in Conferences and Workshops

[21] 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", January 2006.

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

[23] T. ARCILA, J. ALLARD, C. MÉNIER, E. BOYER, B. RAFFIN. *FlowVR: A Framework For Distributed Virtual Reality Applications*, in "Première journées de l'Association Francaise de Réalité Virtuelle, Augmentée, Mixte et d'Interaction 3D, Rocquencourt, France", November 2006.

[24] L. A. BARCHET-STEFFENEL, G. MOUNIÉ. *Scheduling Heuristics for Efficient Broadcast Operations on Grid Environments*, in "Proceedings of the International Workshop on Performance Modeling, Evaluation, and Optimisation of Parallel and Distributed Systems (PMEO-PDS'06), in conjunction with IPDPS'06", 2006.

[25] J. BERNARD, J.-L. ROCH, S. DE PAOLI, M. SANTANA. *Adaptive Encoding of Multimedia Streams on MPSoC.*, in "ICCS'06 International Conference on Computational Science (4), workshop Real-Time Systems and Adaptive Applications, Reading, UK", Lecture Notes in Computer Science (LNCS), n° 3994, Springer-Verlag, May 2006, p. 999–1006, http://hpc.csie.thu.edu.tw/gpc2006/.

[26] X. BESSERON, S. JAFAR, T. GAUTIER, J.-L. ROCH. *CCK: An Improved Coordinated Checkpoint/Rollback Protocol for Dataflow Applications in KAAPI*, in "ICTTA'06 IEEE Conference on Information and Communication Technologies: from Theory to Applications, Damascus, Syria", IEEE, April 2006, p. 3353–3358, http://ictta.enst-bretagne.fr/.

[27] X. BESSERON, L. PIGEON, T. GAUTIER, S. JAFAR. *Un protocole de sauvegarde/reprise coordonné pour les applications à flot de données reconfigurable*, in "Proceedings des 17èmes rencontres francophones du parallélisme (RenPar'17), Perpignan, France", October 2006, http://www.renpar.org/.

[28] F. BLACHOT, B. DUPONT DE DINECHINI, G. HUARD. *SCAN: a Heuristic for Near-Optimal Software Pipelining*, in "European conference on Parallel Computing (EuroPar) Proceedings", 2006.

[29] C. CERIN, J.-C. DUBACQ, J.-L. ROCH. *Methods for Partitioning Data and to Improve Parallel Execution Time for Sorting on Heterogeneous Clusters*, in "International conference on Grid and Pervasive Computing, IGC'2006, Tunghia, Taiwa", Lecture Notes in Computer Science (LNCS), n^{o} 3947, Springer-Verlag, May 2006, p. 175-186, http://hpc.csie.thu.edu.tw/gpc2006/.

[30] S. J. COX, T. LIPPERT, G. ERBACCI, D. TRYSTRAM. *Editorial Topic 16: Applications of High-Performance and Grid Computing.*, in "Proceedings of EuroPar 2006, Dresden, Germany", Lecture Notes in Computer Science (LNCS), Springer, 2006, 1041.

[31] S. J. COX, T. LIPPERT, G. ERBACCI, D. TRYSTRAM. *Topic 16: Applications of High-Performance and Grid Computing.*, in "Euro-Par", 2006, 1041.

[32] V. D. CUNG, V. DANJEAN, J.-G. DUMAS, T. GAUTIER, G. HUARD, B. RAFFIN, C. RAPINE, J.-L. ROCH, D. TRYSTRAM. *Adaptive and Hybrid Algorithms: classification and illustration on triangular system solving*, in "Transgressive Computing, Granada, Spain", April 2006, p. 131–148.

[33] V. D. CUNG, V. DANJEAN, J.-G. DUMAS, T. GAUTIER, G. HUARD, B. RAFFIN, C. RAPINE, J.-L. ROCH, D. TRYSTRAM. *Adaptive and Hybrid Algorithms: classification and illustration on triangular system solving*, in "Transgressive Computing, Granada", April 2006, p. 131–148.

[34] V. D. CUNG, J.-G. DUMAS, T. GAUTIER, G. HUARD, B. RAFFIN, C. RAPINE, J.-L. ROCH, D. TRYSTRAM. *Adaptive algorithms: theory and application*, in "SIAM conference on Parallel Processing, San Francisco, USA", February 2006, http://www.siam.org/meetings/pp06/.

[35] V. D. CUNG, J.-G. DUMAS, T. GAUTIER, G. HUARD, B. RAFFIN, C. RAPINE, J.-L. ROCH, D. TRYSTRAM. *Adaptive algorithms: theory and application*, in "SIAM Parallel Processing, San Francisco", February 2006, http://www.siam.org/meetings/pp06/.

[36] J.-G. DUMAS, C. PERNET, J.-L. ROCH. *Adaptive Triangular System Solving*, in "Dagstuhl Seminar Proceedings – Challenges in Symbolic Computation Software, Dagstuhl, Germany", W. DECKER, M. DEWAR, E. KALTOFEN, S. WATT (editors). , July 2006, http://www.dagstuhl.de/de/program/calendar/semhp/?semnr=06271.

[37] G. DUNOYER, T. GAUTIER, G. MOUNIÉ, J.-L. ROCH. *Parallélisation d'un moteur d'inférence bayesienne pour une machine SMP avec Kaapi*, in "Proceedings des 17èmes rencontres francophones du parallélisme (RenPar'17), Perpignan, France", October 2006, http://www.renpar.org/.

[38] S. JAFAR, L. PIGEON, T. GAUTIER, J.-L. ROCH. *Self-Adaptation of Parallel Applications in Heterogeneous and Dynamic Architectures*, in "ICTTA'06 IEEE Conference on Information and Communication Technologies: from Theory to Applications, Damascus, Syria", IEEE, April 2006, p. 3347–3352, http://ictta.enst-bretagne.fr/.

[39] A. LÈBRE, G. HUARD, P. SOWA. *Optimisation des E/S avec QoS dans les environnements multi-applicatif distribués*, in "Proceedings des 17èmes rencontres francophones du parallélisme (RenPar'17)", October 2006.

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

[41] A. LÈBRE, Y. DENNEULIN, G. HUARD, P. SOWA. *Adaptive I/O Scheduling for Distributed Mutli-applications Environments*, in "Fiftenth IEEE International Symposium on High Performance Distributed Computing (HPDC) poster presentations", 2006.

[42] Z. MAHJOUB, W. NASRI, D. TRYSTRAM. *Computing the inverse of a triangular matrix on heterogeneous clusters*, in "Algorithms, Models and Tools for High Performance Computing on Heterogeneous Networks", Nova, Ed. by F. Desprez, E. Fleury, A. Kalinov and A. Lastovetsky, 2006.

[43] L. MASKO, P.-F. DUTOT, G. MOUNIÉ, D. TRYSTRAM, M. TUDRUJ. *Scheduling moldable tasks for dynamic SMP clusters in SoC technology*, in "Parallel Processing and Applied Mathematics, 6th International Conference, PPAM 2005, Revised Selected Papers, Poznan, Poland", WYRZYKOWSKI, DONGARRA, MEYER, WASNIEWSKI (editors). , Lecture Notes in Computer Science (LNCS), n^{o} 3911, Springer-Verlag, june 2006, p. 879-887.

[44] L. PIGEON, B. BRAUNSCHWEIG, T. GAUTIER, P. ROUX. *Simulation Dynamique d'un Réseau de Production sur Cluster de PC*, in "Proceedings de Colloque Systèmes d'Information, Modélisation, Optimisation et Commande en Génie des Procédés (SIMO06), Toulouse, France", 2006.

[45] L. PIGEON, P. ROUX, B. BRAUNSCHWEIG, T. GAUTIER. *Dynamic CAPE-OPEN Simulation Approach on Cluster Oriented Architecture*, in "Proceedings de AIChE 2006 Annual Meeting, San Francisco, USA", 2006.

[46] B. RAFFIN, L. SOARES. *PC Clusters for Virtual Reality*, in "IEEE Virtual Reality Conference, Alexandria, USA", March 2006.

[47] J.-L. ROCH, D. TRAORE, J. BERNARD. *On-line adaptive parallel prefix computation*, in "EUROPAR'2006, Dresden, Germany", Lecture Notes in Computer Science (LNCS), n^{o} 4128, Springer-Verlag, August 2006, p. 843–850, http://www.europar2006.de/.

[48] J.-L. ROCH, D. TRAORE. *Un algorithme adaptatif optimal pour le calcul parallèle des préfixes*, in "CARI'2006, Cotonou, Benin", INRIA, November 2006, http://www.cari-info.org/index_a.php.

[49] K. RZADCA, D. TRYSTRAM. *Promoting cooperation in selfish grids.*, in "SPAA 2006, 18th annual ACM symposium on Parallelism in Algorithms and Architectures, Cambridge, USA", july 30 – august 2 2006, 332.

[50] K. RZADCA, D. TRYSTRAM. *Promoting cooperation in selfish grids.*, in "SPAA", 2006, 332.

[51] J. P. SANCHEZ, D. TRYSTRAM. *A guaranteed clustering algorithm for large communication delay model*, in "European Conference on Combinatorial Optimization, ECCO XIX - CO 2006 Joint Meeting, Porto, Portugal", may 2006.

[52] C. TATON, S. FONTAINE, S. BOUCHENAK, T. GAUTIER. *Administration autonome d'applications réparties sur grilles*, in "Proceedings des 17èmes rencontres francophones du parallélisme (RenPar'17), Perpignan, France", October 2006, http://www.renpar.org/.

[53] D. TRYSTRAM. *Parallélisme, nouvelles architectures, nouveaux paradigmes de calcul*, in "Communication et Connaissance : supports et médiations a l'age de l'information. Coordonné par J.G. Ganascia", Editions du CNRS, collection Sciences et Techniques de l'ingénieur, 2006.

[54] S. VARRETTE, J.-L. ROCH, J. MONTAGNAT, L. SEITZ, J.-M. PIERSON, F. LEPRÉVOST. *Safe Distributed Architecture for Image-based Computer Assisted Diagnosis*, in "ICPS'06, IEEE International Conference on Pervasive Services, workshop on Health Pervasive Systems, Lyon, France", IEEE, June 2006, http://www.icpsconference.org.

[55] J. ZOLA, D. TRYSTRAM, A. TCHERNYKH, C. BRIZUELA. *Parallel multiple alignment with local phylogeny search by Simulated Annealing*, in "Proceedings of HICOMB'06, Rhodos, Greece", 2006.

Miscellaneous

[56] J. DUMAS, D. TRYSTRAM. *Les anniversaires des briseurs de codesCryptographie et codes secrets, l'art de cacher*, n^{o} 16, Pole, 2006.

[57] J. DUMAS, D. TRYSTRAM. *RSA: forces et faiblesses d'un titanCryptographie et codes secrets, l'art de cacher*, n^{o} 16, Pole, 2006.

[58] G. MOUNIÉ, Y. ROBERT, D. TRYSTRAM. *Scheduling on gridsSchool on GRID5000, Grenoble*, 2006.

[59] L. PIGEON. *Exécution CAPE-OPEN distribuée d'un champ de production sur un cluster de PCJournée IEP (Informatique & Procédé), Grenoble, France*, Communication orale, 2006.

[60] L. PIGEON, P. ROUX, B. BRAUNSCHWEIG, T. GAUTIER, D. PAEN. *Distributed Dynamic CAPE-OPEN Simulation of Oilfield Production Networks on Clusters of PCsCO-LaN Annual Meeting, Cannes, France*, Communication orale, 2006.

[61] B. RAFFIN, J. ROCH, D. TRYSTRAM. *Adaptive algorithms for efficient parallel programmingSummer school on concurrency, Bertinoro, Italy*, 2006.

[62] D. TRYSTRAM. *Programming new computing systemsSéminaire de rentrée, ENS de Cachan*, 2006.