INRIA Resarch project Proposal

"Multi-programmation et Ordonnancement pour les Applications Interactives de Simulation"

MOAIS

Programming and Scheduling Design of Interactive Simulation Applications on Distributed Ressources

Thème 1a
INRIA Rhône-Alpes

January 18, 2005

Abstract

The MOAIS research project is a joined project CNRS–INPG–INRIA–UJF located at ENSIMAG, Montbonnot Saint-Martin site and hosted in the Laboratoire ID-IMAG Informatique et Distribution, UMR CNRS/INPG/INRIA/UJF 5132.

Contents

Preamble

Notice that throughout all this document the term *project* usually refers to an INRIA project; such a project is a research group funded by INRIA to pursue research on a given topic.

MOAIS is a joined *CNRS–INPG–INRIA–UJF project*; it is jointly funded by both national institutions CNRS and INRIA and both Grenoble universities INPG and UJF.

Like the members of the Mescal project, the members of the MOAIS project are all former members of the APACHE project. They are also all members of the "Laboratoire Informatique et Distribution" (ID-IMAG lab, UMR CNRS/INPG/INRIA/UJF 5132).

1 Team

- **Project Leader**
  - Jean-Louis Roch (INPG MdC[1])
- **Project Administrative Assistant**
  - Marion Ponsot (30 %)
- **Researchers**
  - Thierry Gautier (INRIA CR[2])
  - Guillaume Huard (UJF MdC)
  - Grégory Mounié (INPG MdC)
  - Bruno Raffin (INRIA CR)
  - Denis Trystram (INPG Pr)
- **Ph.D. Students**
  - Jérémie Allard (third year)
  - Florent Blachot (second year)
  - Pierre-François Dutot
  - Luiz-Angelo Estefanel (third year)
  - Lionnel Eyraud (second year)
  - Feryal-Kamila Moulai (second year)
  - Hamid Reza Hamidi (fourth year)
  - Samir Jafar (third year)
  - Krzysztof Rzadca (first year)
  - Clément Menier (second year)

[1] Equivalent to Assitant Professor
[2] Full time researcher

  - Jonathan Pecero-Sanchez (second year)
  - Laurent Pigeon (first year)
  - Sebastien Varrette (second year)
  - Jesus-Alberto Verduzco-Ramirez (fourth year year)
  - Jaroslaw Zola (second year)

- **Engineers**
  - Loick Lecointre (INRIA, temporary position until December 2004)

2 Presentation

Taking benefit of numerous operating resources is a key point to push up the limits of applications.

Parallel and distributed computers, from super-scalar sequential machines to large size clusters of symmetrical multi-processor nodes and now grid architectures, are being successfully used to increase the computational power to improve quality and precision of scientific simulations. Immersive environments like Cave systems, take advantage of multiple projectors, sound channels, haptic feedback systems to enforce the user's sense of immersion in a synthetic world. Multiplying input devices has been used for various applications to improve the system responsiveness and to enlarge the amount, quality and diversity of input data (parallel file systems for scientific applications, clusters for web search engines, multiple cameras for motion capture).

Highly specialized architectures have been proposed to provide powerful systems dedicated to such applications, the SGI Onyx machine being among the most famous example for virtual reality applications. In the field of computer vision, efficient sequential algorithms for 3D modeling are proposed to obtain from several cameras a 3D representation of an object. For acoustics, mainly sequential algorithms, relying on DSP processing units, are used to compute spatialized sound for multi-channels output systems from stereo to home theater 5.1 sound systems.

Nevertheless, beyond the optimization of one – software or hardware – component, the effective availability of low-price computational input and output devices (cameras, sensors, projectors, etc.) raises a new challenge: how to build scalable platforms based on off-the-shelf components (input, output and computational units), to drastically improve the performance of applications in terms of output data quality and interactivity. Ideally, it should be possible to improve gradually the performance by adding (dynamically) resources.

Scheduling is one of the key problems to be solved to bridge the gap between the real and ideal worlds. It manages the distribution of the application on the architecture. The complexity here is mainly related to the large number, heterogeneity and dynamicity of hardware and software resources:

- resources are distributed and heterogeneous, some being possibly added or suppressed at any time;
- the application is a complex assembly of various heterogeneous components;
- the performance objective depends on multiple criteria: precision, latency, coherency, refresh rate, fairness, economy. Its formalization is an open problem.

Thus, fundamental researches undertaken in the MOAIS project are focused on this scheduling problem. The members of the project all have an expertise on this problem. The originality of the MOAIS approach is to use the application's adaptability to enable its control by the scheduling.

3 Scientific Foundations

The MOAIS project focuses on applications where performance is a matter of resources: beyond the optimization of the application itself, the effective use of a larger number of resources is expected to enhance the performance.

This encompasses large scale scientific simulations that have played a prominent role in the development of high performance parallel computing.

Today as available resources are becoming more numerous and more heterogeneous, it is critical to develop environments that enable the applications to efficiently adapt to the availability of these resources.

The MOAIS project proposes to address some of these issues, related to two strategic research directions of the INRIA: to couple data and models to simulate and control complex systems; to combine simulation, visualization and interaction.

Applications of scientific computations involve coupling of complex models with different time and space discretization scales. For instance, multi-physics problems require the coupling of several codes: underground fluid flow models with surface fluid dynamics flow models, chemistry and transport model coupling. Beyond simulation computations, other codes may have to be coupled to provide input data or to compute output data like images or statistics to control the application. The resulting application is a complex coupling of various codes distributed on heterogeneous resources (including several input and output devices).

Each code is itself a component, usually parallelized (with MPI for instance) on a cluster or grid to obtain the computational power required by the simulation. Such a component includes its own local scheduling rules. Coupling is implemented by communications between the components. Although this coupling is critical, there is no high level support to express it. As such, the global application scheduling is reduced to the juxtaposition of the local scheduling of each distributed component. A performance loss on one component may affect the performance of the whole application. This is especially critical in case of complex interactions between components.

Simulation results being more and more complex, scientific visualization is becoming an essential tool for their analysis. Future scientific applications will take advantage of virtual reality technologies to provide an intuitive and interactive work space, with possible collaborative work across distant sites. For such applications, increasing the number of resources is a key to improve performance:

- 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). Usually, computing resources are partially abstracted. Most applications rely on MPI (Scalapplix,...) where the number of resources, usually assumed identical, is a parameter. Other environments (Charm++, Athapascan, Smarts) are based on a higher level of abstraction: resources may be heterogeneous and their number may vary during execution. However, coupling codes and integrating multiple input and output interfaces in

these frameworks is difficult.

- the application control is related to the quality and number of input resources (sensors, cameras, microphones). Software tools (e.g. COVIZE, VR Juggler) provide standardized interfaces for most input resources. But having a dynamical number of input devices requires distributed algorithms that are able to efficiently adapt to the available data. For instance real time 3D modeling can require tens of cameras. The algorithm should be able to remove/add cameras smoothly without a global synchronization barrier that would affect the real-time performance.

- 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 the data coherency across those multiple outputs. Existing softwares like SoftGenLock, Net Juggler, Chromium or Zysygy, provide simple synchronization schemes, typically a global synchronization barrier. But when scaling to large systems the cost of such approaches becomes prohibitive. New approaches that enable a loosen coherency, for the objects that are outside or far from the user point of view for instance, must be developed.

These 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.

Indeed, those performance criterion are antagonists. On the one hand, the less the synchronization and computation steps between input sensors and output peripherals, the better the reactivity. On the other hand, precision and fluidity are improved by increasing the degree of parallelism, leading to extra synchronization costs. Three kind of synchronizations can be identified: data dependencies synchronization for the simulation level; low level synchronizations to guarantee a time-coherent data acquisition; output device synchronization to ensure the user does not experience inconsistencies (swap-buffer synchronization on a display wall for instance). Then the scientific problem is to control those various synchronizations in order to improve scalability and performance.

To solve this problem, MOAIS focuses on scheduling. Scheduling bridges between an application and its execution:

- 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 programming (BSP, Nesl, Earth, Jade, Cilk, Athapascan, Smarts, ...) 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.

Then, core research in MOAIS is the scheduling. Entry of the scheduling is not a graph as classically encountered in the literature but -dynamic- tasks and synchronization relations of various kinds. Also, the scheduling plays the role of an interpreter of the macro data flow related to the application. For the sake of scalability, this data flow and its scheduling has to be distributed. The scheduling controls the data flow to reach a global performance objective. This control is based on the adaptation of the application. Since the multi-criterion performance objective mixes reactivity, fluidity and precision, the adaptation has to be possible on the various components that are coupled in an application of interactive simulation.

Therefore, focused on those applications, research axis of MOAIS are directed towards:

- **Scheduling.** To formalize and study the related scheduling problem. The critical points are: the modelization 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 the algorithm is 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.
- **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.

The research is not only theoretical but also practical, centered on applications developed with external partners. Softwares developed in the MOAIS project are all based on an efficient management of the data flow related to the execution.

4 Research Themes

4.1 Scheduling

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 (participants D. Trystram, G. Huard, J.L. Roch).

The area of scheduling is already quite well established [16,13,26] as the first significant results have been established during the seventies for manufacturing systems. In the context of parallel and distributed processing, the term *scheduling* is used with many acceptions. 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 the 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). Current challenges in scheduling are mainly how to take into account the increasing complexity of the execution parameters of new systems that are both distant and heterogeneous.

Context and Problem Positioning. The MOAIS project deals with the efficient and transparent programming of irregular applications on parallel and distributed systems. In this context, scheduling is the key issue when looking for performance. Once potential parallelism has been exhibited within one or several applications, one has to find an allocation of tasks to available resources as well as a date of execution. For highly dynamic applications, it might be necessary to perform this task at runtime.

Currently, research in this field aims at characterizing and designing scheduling algorithms for realistic applications that will be executed on the new systems. Unfortunately, there is no common acceptance of an universal model. Past research results dealing with simple classes of parallel systems (with respect to the current view of distributed computing!) such as shared memory machines or homogeneous clusters of workstations cannot be used anymore. These works, based on the *delay model* have reached their limits despite some interesting theoretical attempts of extension (for instance, the extension of delays on a set of identical processors to a set of uniform processors). The new execution platforms are composed of a large number of processors, most of the time geographically distant, hierarchically organized and heterogeneous.

Recently, two models have emerged that try to get closer to these new systems: divisible tasks and parallel tasks [26]. In both cases, underlying execution models hide the complexity associated to the explicit handling of communications.

Divisible tasks gather a large class of applications of parametric nature, for which the computation work is spread among a very large number of elementary sequential tasks; then the problem is to limit the overhead due to the management of this huge number of tasks on the relatively few available resources; this

[16] H. El-Rewini, T.G. Lewis, and H.H. Ali. *Task Scheduling in Parallel and Distributed Systems.* Prentice Hall, New Jersey, 1994.

[13] P. Chretienne, E.G. Jr Coffman, J. K. Lenstra, and Z. Liu. *Scheduling Theory and its Applications.* John Wiley and Sons, England, 1995.

[26] J. Leung, editor. *Handbook on Scheduling algorithms: Algorithms, Models and Performance Analysis.* CRC press, 2004.

is the basis of work-stealing scheduling algorithms [7,3].

On the contrary, parallel tasks might be viewed as a dual approach in which tasks are themselves small parallel programs that will executed on several processors. A distinction is made between rigid tasks, for which the number of processors allocated to the task is fixed once for all before the scheduling, moldable tasks, for which this number is computed by the scheduler once for each task and malleable tasks, for which this number might evolve dynamically during the execution of a task. For these models, major results have been established that will serve as a basis for most of the future works.

We detail below the current research directions aiming at building efficient and operational scheduling algorithms for managing the resources while running parallel programs on new execution supports. Our approach is both theoretical and experimental; studied scheduling algorithms are implemented and experimentally evaluated. Furthermore, we will integrate those scheduling algorithms in prototypes dedicated to the management of effective execution supports.

Extensions of the Parallel Tasks model. The characteristics of grids and global computing are intrinsically dynamic. We know how to derive, from an off-line scheduling algorithm, an efficient on-line one which has a good performance guarantee for rigid or moldable tasks [15]. The method is to use 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 twice the optimal (which is the worst case).

One problem which remains to be solved is how to mix rigid and moldable tasks. Most of the actual codes are rigid ones, mainly due to historical inertia (the users just look at the available CPU resources before submitting a job). This problem is strongly related to the management of reservation periods which should be taken into account for a practical use.

Stability and Sensitivity. A crucial factor for implementing the scheduling algorithms on new execution supports is *versatility* (temporary addition – or removing – of machines, bad estimations of times for transferring data, internal cache effects that affect the computation speed, etc.). Thus, it is necessary to develop mechanisms for reacting to the disturbances of the parameters from which the schedule has been determined.

The first possibility (called sensitivity) is to analyze the scheduling algorithms in regard to their ability to react to disturbances. The second possibility (the stability) is to develop adequate correction mechanisms to control these effects [25]. For instance, work-stealing scheduling may be considered as a way to control stability of an on-line greedy schedule, not stable in the general case [6].

[7] R.D. Blumofe and C.E. Leiserson. Space-efficient scheduling of multithreaded computations. *SIAM Journal on Computing*, 27(1):202–229, 1998.

[3] U. Acar and G. Blelloch. The data locality of work stealing. In *Proceedings of the ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, june 2000.

[15] P-F. Dutot, G. Mounie, and D. Trystram. *Scheduling Parallel Tasks: Approximation algorithms*, chapter 26. CRC Press, to appear april 2004.

[25] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications.* Kluwer Academic Publishers, 1996.

[6] P. Berenbrink, T. Friedetzky, and L.A. Goldberg. The Natural Work-Stealing Algorithm

Our work focuses on this last topic. Our idea is to propose generic algorithmic schemes for correcting the solution determined without disturbances.

Multi-objective Analysis. A natural question while designing practical scheduling algorithms is "which criterion should we optimize ?". Most of 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. Recently, we proposed an algorithm which is guaranteed for both *makespan* and *minsum*. This algorithm is being implemented and will be used to manage the resources of the Icluster-2.

In the case of parallel interactive applications, performance is a mix of three antagonist objectives: latency, precision and rendering. Even if interactive objective is intuitive, its formalization is challenging. Nevertheless, it may be possible to design scheduling which guarantees performance for antagonist objectives. As a crucial example, minimizing memory space and parallel time are two antagonist objectives [7]; however, scheduling algorithms have been proposed that achieve provable bounds with respect to both objectives [29]. We study such algorithms in the framework of macro dataflow computations.

4.2 Adaptative Parallel and Distributed Algorithms Design

This theme deals with the analysis and the design of algorithmic schemes that control (statically or dynamically) the grain of interactive applications for simulation (participants T. Gautier, B. Raffin, J.-L. Roch, D. Trystram).

The current trend in the evolution of parallel machines is an exponential growth in the number of nodes allocated to some computation. Clusters of hundreds to thousands of workstations have replaced multiprocessor machines and the clusters themselves are in the process of being interconnected into massive grids (as proposed in the project grid 5000).

Nevertheless, the data exchange and the coordination are significantly more complex problems when synchronizing thousands of nodes than when just a dozen of workstations are implied in the process. In such a context, not only the capabilities of network links between nodes is quickly insufficient, but the synchronization of a large number of machines implies that many of them are waiting for the slowest ones. In other words, a strong synchronization during

---

is Stable. *SIAM Journal on Computing*, 32(5):1260–1279, 2003.

[7] R.D. Blumofe and C.E. Leiserson. Space-efficient scheduling of multithreaded computations. *SIAM Journal on Computing*, 27(1):202–229, 1998.

[29] G.J. Narlikar. Scheduling threads for low space requirement and good locality. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 83–95, 1999.

the computation creates both a global workload imbalance between resources and a slowdown due to communications.

Ideally, a good parallel execution can be viewed as a flow of computations flowing through resources with no control overhead. To minimize control overhead, the application has to be adapted: a parallel algorithm on p resources is not efficient on $q < p$ resources. On one processor, the scheduler should execute a sequential algorithm instead of emulating a parallel one. Then, the scheduler should adapt to resource availability by changing its underlying algorithm.

This 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. This idea is the basis upon which Athapascan, the parallel programming interface developed by the APACHE project, has been built [20].

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. To deal with heterogeneous and/or dynamical resources (e.g. grid architectures), 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 [19], Hood [8], Athapascan [21]) 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.

However, the algorithm may adapt itself at runtime in order to improve performance by decreasing synchronization overheads [30]. For instance, the evaluation of a stopping condition usually involve communications. Then, the frequency of this evaluation should depend on its cost and on the degree of parallelism. A distributed 3D modeling algorithm should be able to integrate/remove

[20] F. Galilée, J.-L. Roch, G. Cavalheiro, and M. Doreille. Athapascan-1: On-line Building Data Flow Graph in a Parallel Language. In IEEE, editor, *International Conference on Parallel Architectures and Compilation Techniques, PACT'98*, pages 88–95, Paris, France, October 1998.

[19] M. Frigo, C.E. Leiserson, and K.H. Randall. The Implementation of the Cilk-5 Multithreaded Language. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'98)*, June 1998.

[8] R.D. Blumofe and D. Papadopoulos. HOOD: A User-level Threads Library for Multiprogrammed Multiprocessors. Technical Report http://www.cs.utexas.edu/users/hood/, The University of Texas at Austin, October 1998.

[21] T. Gautier, R. Revire, and J.-L. Roch. Athapascan: Api for asynchronous parallel programming. Technical Report RR-0276, APACHE, INRIA Rhône-Alpes, February 2003.

[30] M.C. Rinard and P. C. Diniz. Eliminating synchronization bottlenecks using adaptive replication. *ACM Transactions on Programming Languages and Systems*, 25(3):316–359, may 2003.

cameras with a minimal cost, avoiding global synchronizations as much as possible, to keep a sustained frame rate.

Furthermore, an adaptative algorithm not only fits to resource availability, but also meets the multicriterion performance objectives (latency, precision, level of detail and refresh rate constraints [27]):

- enrolling/unrolling some parts of the computation may be used to tune the dicretization step (time, space,...) and meet the precision constraints of the numerical simulation [28].

- latency is affected by the number of input and output resources. The greater this number, the longer the latency. Thus, if the latency constraint is no more met, a way to decrease latency is to ignore some of those resources in the application. This requires the application to be dynamically maleable. For instance, the application might be asked to decrease its latency or to improve other performance objectives.

Adaptivity to resources and constraints relies on the underlying algorithm. It is a difficult problem, that may have deep implications on its structure. A 3D modeling adaptable algorithm should take a variable number of camera views as its inputs.

The main approach consists in finding a good mix of various algorithms, what is often called a "poly-algorithm". For instance, the Atlas library [33] uses performance benchmark scores to decide at compile time the best block size and instruction interleaving for sequential matrix product. At run-time FFTW algorithm [18] considers the effective dimension of the input vector to find by dynamic programming the best split factors and thresholds from nested recursive FFTs. Both cases rely on pre-benchmarking of the algorithms. To suit the granularity of dynamic and heterogeneous resources, we will study a generic algorithmic scheme based on the recursive coupling of two different algorithms, a sequential one and a parallel one. We have used this scheme to build an adaptative compression algorithm [24].

An other class of adaptive algorithms, called "anytime algorithms", adapts the quality of the result computed to the amount of time available [9,22]. Those

[27] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of Detail for 3D Graphics*. Morgan Kaufmann, 2002.

[28] S. Moore and V. Eijkhout. Workshop on adaptive algorithms for parallel and distributed computing environments, June 2003. www.cs.utk.edu/shirley/iccs2003-adaptalg/, to appear in Springer-Verlag LNCS.

[33] R.C. Whaley, A. Petitet, and J. Dongarra. Automatically tuned linear algebra software (ATLAS), http://math-atlas.sourceforge.net/, 2000.

[18] M. Frigo and S.G. Johnson. Fftw: An adaptive software architecture for the fft. In *(*`www.fftw.org`*)ICASSP Conference Proceedings*, volume 3, pages 1381–1384, octobre 1998.

[24] A. Kerfali, J.-L. Roch, and M. Daoudi. Algorithmes parallèles à grain adaptatif - Application à la parallélisation de gzip. In *RENPAR'15*, pages 18–26, Nice, France, octobre 2003.

[9] M. Boddy and T. Dean. Solving time-dependent planning problems. In *Proceedings of IJCAI-89*, pages 979–984, 1989.

[22] J. Grass and S. Zilberstein. Anytime algorithm development tools. *SIGART Bulletin*

algorithms have been particularly developped in the field of artificial intelligence [34]. Today, anytime algorithms are mainly sequential algorithms; the degree of parallelism could be taken into account has an extra parameter to improve the quality of the result given a certain amount of completion time.

4.3 Interactivity

The goal of this theme is to develop approaches to tackle interactivity in the context of large scale distributed applications (participants T. Gautier, B. Raffin, J.-L. Roch).

Today interactivity is often limited to applications of relatively reduced complexity that can be executed on a workstation (CAD system for example) or on dedicated mid size multi-processor machines like SGI Onyx. For complex applications, visualization is often limited to a post-mortem processing of the application results [4]. Interaction is them reduced to manipulating a "static" data set with no interactive feedback on the application execution. Even in this case, interaction can be difficult due to the amount of data that may have to be processed (several gigabytes for oil reservoir visualization for instance). Distributed virtual environments (distributed battle fields or video games) succeed to provide interactivity based on a high degree of data replication and loosely coupled systems exchanging lightweight data like other gamers position.

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 [14], allows a high resolution rendering on a large surface. Stereoscopic visualization gives a information of depth. Sound and haptic systems (force feedback) can provide extra information in addition to visualized data. However to drive 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 tow different approaches to handle data exchange between

---

Special Issue on Anytime Algorithms and Deliberation Scheduling, 1995.

[34] Shlomo Zilberstein and Stuart Russell. Optimal composition of real-time systems. *Artif. Intell.*, 82(1-2):181–213, 1996.

[4] J. Ahrens, C. Law, W. Schroeder, K. Martin, and M. Papka. A parallel approach for efficiently visualizing extremely large, time-varying datasets. http://www.acl.lanl.gov/Viz/papers/pvtk/pvtkpreprint/.

[14] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart. The Cave Audio VIsual Experience Automatic Virtual Environement. *Communication of the ACM*, 35(6):64–72, 1992.

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 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 [23,12]. 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 make 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

[23] G. Humphreys, M. Eldridge, I. Buck, G. Stoll, M. Everett, and P. Hanrahan. WireGL: A Scalable Graphics System for Clusters. In *Proceedings of SIGGRAPH 2001*, 2001.

[12] H. Chen, Y. Chen, A. Finkelstein, T. Funkhouser, K. Li, Z. Liu, R. Samanta, and G. Wallace. Data Distribution Strategies for High-Resolution Displays. *Computers & Graphics, Special Issue on Mixed Realities*, 2001.

to reach a tradeoff given the user wishes and the resources available. Issues include multicriteriom optimizations, adaptative algorithmic schemes, distributed decision making, global stability and balance of the regulation effort.

4.4 Coupling and Data Movements

This theme deals with the design and implementation of programming interfaces in order to achieve an efficient coupling of distributed components (participants T. Gautier, B. Raffin, J.-L. Roch).

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[11] 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 make possible 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.

Application Programming Interface. The specification of an API to compose interactive simulation application requires to characterize the components and the interaction between components[31]. 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 few 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 semantics.

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 semantics[31] 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

[11] N. Carriero and D Gelernter. Coordination languages and their significance. *Communication of the ACM*, 35(2):97–107, 1992.

[31] M. Schumacher, editor. *Objective Coordination in Multi-Agent System Engineering: Design and Implementation*, volume 2039. Springer-Verlag Heidelberg, 2001.

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.

Kernel for Asynchronous, Adaptive, Parallel and Interactive Application. To manage the complexity related to fine grain components and to reach 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[32,5,17], the kernel will provide an interface to specialize the computation of a good schedule of the data flow graph. Moreover, the kernel will provide operators on the graph (*e.g.* computes a partition from a schedule, remapping tasks, ...) to allow to control a distributed execution.

5 Context and Positioning

This section positions the research themes of the INRIA MOAIS project in the following contexts:

- inside INRIA: the other INRIA projects that address similar problems are listed, precising for each the originality of MOAIS;
- international: the main related research areas and the work in progress are described;
- national: we give an overview of the MOAIS actions and collaborations at a national level.
- inside the ID laboratory: we focus on the origin of MOAIS and its relation ship with the MESCAL project that co-emerges from APACHE.

[32] J. Tao, M. Schulz, and W. Karl. Ars: an adaptive runtime system for locality optimization. *Future Generation Computer Systems*, 19(5):761–776, 2003.

[5] K. Barker, A. Chernikov, N. Chrisochoides, and K. Pingali. Mapping applications to machines - a load balancing framework for adaptive and asynchronous applications. *IEEE Transactions on Parallel and Distributed Systems*, 15(2):183, 2004.

[17] C. T. H. Everaars, F. Arbab, and B. Koren. Dynamic process composition and communication patterns in irregularly structured applications. *Concurrency Practice and Experience*, 12(2-3):157–174, 2000.

5.1 Positioning inside INRIA

The MOAIS project is related to two strategic research axis of the INRIA: to couple data and models to simulate and control comlex systems; to combine simulation, visualization and interaction.

ALCOVE (`www.lifl.fr/ALCOVE/`) The Alcove project focuses on collaborative and interactive virtual environments. In particular they develop the Spin3D software for collaborative work based on a Corba technology.

While interactivity is an important performance factor for collaborative work, MOAIS rather focuses on finer grain applications.

CALVI (`math.u-strasbg.fr/calvi/`) The CALVI project focuses on developing algorithms for numerical simulation of complex phenomena, their execution on large clusters and grid environments, as well as the execution of degraded versions of these algorithms for interactive visualization.

MOAIS and CALVI have close research topics, CALVI being more concerned with some specific complex simulation like fluid an plasma dynamics, while MOAIS focus on interactivity related issues. A collaboration is currently emerging, helped by the presence at CALVI of Florence Zara, a former Ph.D. student from ID/MOAIS.

GRAAL (`graal.ens-lyon.fr/`) The GRAAL project focuses on algorithmic design, middleware libraries and applications over large scale heterogeneous architectures and the grid. While the GRAAL and the MOAIS projects share some common research directions such as the definition of new task models and the design of scheduling algorithms for heterogeneous and dynamic platforms, their final objectives are different.

The GRAAL project proposes to cope with new grid constraints by allowing more flexibility in the tasks model : this is the divisible task model. The view of the MOAIS project is rather to hide the complexity of grid scheduling problems by hiding the complexity of inter-task communication within the hierarchical organization of the parallel task model.

The GRAAL project also focuses on some kind of absolute efficiency of scheduling algorithms through asymptotic optimality. Following this idea, they propose a new trend in scheduling : the steady state scheduling. The goal of MOAIS project is rather to cope with unpredictable execution support by developping stable scheduling algorithms (almost insensitive to variations in network or cpu performance) and to fullfill the complex needs of grid users by developping multicriteria scheduling algorithms (with performance guaranteed for several objectives simultaneously).

OASIS (`www-sop.inria.fr/oasis/`) The OASIS project focuses on developping fundamental principles, techniques and tools for the building, analysis, validation, verification and maintenance of reliable systems in the domain of distributed applications, networks (Internet and intranets), smartcards, and termi-

nals. The MOAIS approach is complementary as it targets scheduling problems and interactivity.

PARIS (`www.irisa.fr/paris/`) The PARIS project focuses on distributed and parallel programming environments for large scale numerical simulations. While their work on parallel code coupling is of high interest for MOAIS, they do not target interactive applications.

SCALAPPLIX (`www.labri.fr/Recherche/PARADIS/Scalapplix/`) The Scalapplix project focuses on high performance distributed algorithms for complex scientific applications. Large scale application steering using a virtual reality environment is also considered.

The MOAIS research activity is complementary as it focuses on interactivity, scheduling and coupling for parallel applications rather than on algorithms for scientific applications.

SIAMES (`www.irisa.fr/siames/`) The SIAMES project focuses on image synthesis, animation, modeling and simulation. SIAMES develops the OpenMask library for distributed interactive simulations. It differ from FlowVR as OpenMask does not support parallel code coupling.

5.2 International Positioning

The MOAIS project concerns mainly five research areas:

- **Scheduling for today's platforms.** The international research on scheduling has made constant progress for many years. The main results established in the context of manufacturing systems have been adapted to Parallel Processing. Nevertheless, most studies are theoretically oriented and only few software development are available. One originality of the MOAIS project is to cover the whole spectrum from well-founded mathematical backgroud to actual implementations.

  Parallel Tasks model is a recent but very active topic today which has been proposed in the late eighties. The researchers of MOAIS promoted the feature of moldability within this model and have established the best guaranteed approximation known at this time for independent jobs in both off-line and on-line cases. On another hand, multi-criteria analysis is a subject which is more and more studied in the context of scheduling. Our view is to focus on determining analytically Pareto curves (which represent the best trade-off between criteria). Finally, robustness of scheduling (defined as the ability to react to disturbances on inputs) is envisaged through the design of new policy that are intrinsically stable. No existing scheduling algorithms have this feature to our knowledge.

  We are also building a practical scheduler for grids (included in the OAR batch scheduler developed in collaboration with the MESCAL project). Although similar tools already exist (MAUI scheduler, Condor, XtremeWeb),

they are at best based on classical models (rigid tasks) and do not address new needs in grid scheduling such as massive and parametrized task submissions, opportunist computation and multicriteria fullfilment.

- **Programming and coupling environments for clusters and grids.** Developing parallel programming environments has been a preoccupation of numerous researh labs for a long time (OCCAM, PVM, MPI, HPF, OpenMP, CHARM++, CILK, etc.). Today attention is focused onto large clusters and grids where the ability to couple several parallel codes becomes critical. In that field the most known initiative is certainly the Globus project at NCSA. As grid technologies improves, it becomes clear that visualization and interaction should be an important element in grid architectures. Based on the Athapascan and Net Juggler experience, MOAIS proposes to develop a new generation of tools adapted to interactive applications in a large scale context (FlowVR, FlowMD, CacheFlow).

- **Distributed virtual environments.** This research area focuses on large scale virtual environements distributed at a internet level and using P2P-like approaches. The main applications are for network gamming, battlefield simulations, virtual communities (DIS/MMA, DIVE, Cavern). Interactivity in a large scale context is of main concern but coupling of parallel computations is not addressed.

- **Virtual reality.** This research area focuses on developping highly interactive applications using multi-sensorial input and ouput devices to provide users with a sense of immersion in a synthetic world (CaveLib, VR Juggler, Covise, OpenSG, Syzygy, Dice). Classically, these multi-projector environments like CAVEs or Workbenches, are driven by mid-size dedicated computers equiped with the required low level synchronization mechanisms (SGI ONYX). Today, such computers are being replaced by PC clusters. But commodity components lack of the hardware for low level synchronizations. It must be compensated by software solutions. MOAIS did contributed to that change with sofwares like Net Juggler and Soft-GenLock. Our goal is now to provide solutions for the next generation of virtual reality application that will require an important computing power using large clusters or grids. On these topics MOAIS has close connections with the VRAC (`vrac.iasate.edu`) that develops the VR Juggler platform.

- **Scientific visualization.** Visualization is an efficient way to evaluate results from complex simulations. Multi-display environments and clusters are used today to travers large (possibly distributed) data sets to build high resolution images (volume rendering, iso-surfaces). Rather than working on static data sets, another approach proposes to couple the visualization with the simulation. In both cases it involves coupling parallel codes, multiples output interfaces ans sometimes multiple inputs (storage disks). At an international level , the Los Alamos National Labs (`www.ccs.lanl.gov/ccs1/projects/Viz`), the HLR Stuttgart (`www.hlrs.de`) and the GEOFEM (`geofem.tokyo.rist.or.jp`) are important actors. MOAIS col-

laborates with the BRGM and CEA/DAM on coupling sismic simulations and visualization. MOAIS also experimented coupling of parallel codes and multi-projector rendering in an interactive contexte, using Net Juggler and PETSC for a fluid simulation, Net Juggler and Athapascan for a cloth simulation. Next step is to move to larger clusters and grids taking advantage of our close collboration with the MESCAL project.

5.3 National Positioning

At a national level, the research projects listed in section 5.1 encompass most of the research activities performed on the topics MOAIS is focused on. We also actively collaborate with non INRIA research laboratories, like IMAG (federation of computer science and applied mathematics laboratories of Grenoble) and LIFO (computer science laboratory of the University of Orléans), or research institutes, like IFP (petroleum research insitute), CEA (atomic energy research center), BRGM (geological research institute). Most of these collaborations are funded by national programs (see section 8.3). MOAIS also collaborates with private companies like ST Microelectronics, Bull, TGS (see section 8.3).

MOAIS national visibility is also enforced by its implication in different experimental platforms, like CIMENT, grid 5000 and Grimage (see section 8.2).

5.4 Positioning inside of the ID laboratory

The research themes of MOAIS emerged through different activities of the APACHE project:

- the Athapascan-1 portable parallel programming environment, based on a macro dataflow interpreted at run-time. Athapascan has been mainly used for non interactive parallel simulations. A Ph.D. work (Florence Zara) has however shown the benefits and limits of Athapascan-1 in the context of interactive cloth simulation.

- the study of scheduling strategies, that led to the specification, classification and resolution of different issues of load balancing.

- the NetJuggler library that enables a coherent distributed rendering across a display wall. NetJuggler also allows a synchronous coupling between a distributed simulation and a distributed image rendering.

- the development of applications, in particular molecular dynamics (Symbio and Takakaw codes), phylogeny and cloth simulation (Sappe code), that has exhibited issues of introducing interactivity in parallel executions.

On these themes, the APACHE project brought effective contributions, practical and theoretical, up to the development of applications. However these results obtained are only partially related to interactive distributed applications, which require an adaptative coupling of sensors, actuators and simulations in a distributed context.

This observation has motivated a new team dynamics in the ID laboratory that led to the MOAIS project. The co-emergence of the MESCAL project focused on distributed execution supports for large clusters and grids, helped to center the MOAIS research topics at the application and scheduling level. The cooperation between MOAIS and MESCAL is materialized through the co-development of the OAR batch scheduler and the Inuktitut communication library.

6 Softwares

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.

All softwares will be available as open source.

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

6.1 FlowVR

FlowVR is a middleware dedicated to large scale virtual reality applications. FlowVR supports coupling of heterogeneous parallel codes and is component oriented to favor code reuse. While classical communication paradigms focus on either a synchronous approach (FIFO channels) or an asynchronous one (sampling), FlowVR enables a large range of intermediate policies to better balance the application performance between level of details, latencies and refresh rates.

FlowVR reuses and extends the data flow paradigm commonly used for scientific visualization environments (SCIRun [2], COVISE [1]). A VR 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, the FlowVR engine taking care of moving data between producers and consumers. This favors code reuse and enables to keep a simple FlowVR application programming interface (API) to ease turning an existing code into a FlowVR module. In case of a parallel code, several or all processes or threads can be ported to become FlowVR modules. For module data exchange, FlowVR defines an abstract network featuring from simple routing operations to complex message handling operations.

---

[2] Scirun: A scientific computing problem solving environment. Scientific Computing and Imaging Institute, http://software.sci.utah.edu/scirun.html.

[1] COVISE. http://www.hlrs.de/organization/vis/covise/.

Each message is associated with a *list of stamps*, a 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. Besides predefined FlowVR stamps, others, like a time or a 3D bounding box for instance, may be added to extend the network routing, filtering or synchronization abilities. The FlowVR network enables to build complex collective communications, a desirable feature for efficient parallel code coupling. It is also possible to go beyond the classical synchronization barrier, designing synchronizations waiting for the resolution of complex constraints based on stamps (a data semantically richer than a signal).

FlowVR is currently under development and will be ported to Linux for IA32 and IA64 architectures as well as Mac OS X for G5. FlowVR is co-developed with the LIFO (Laboratoire d'Informatique Fondamentale d'Orléans). All FlowVR related materials (code, documentation, mailing lists) are available at `http://flowvr.sf.net`. Source code will be available under GPL and LGPL licences.

6.2 Kaapi - Kernel for Asynchronous, Adaptive, Parallel and Interactive Application

Kaapi is a runtime support using 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. The Athapascan software, developed by the APACHE project, provides a C++ API, which 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 the 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, have been used to compute combinatorial optimization problems on the French Grid Etoile.

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. Moreover, Kaapi generates traces of execution that could be displayed by the Paje software.

Kaapi is developed for UNIX platform and has been ported on most of the UNIX (LINUX, IRIX, Mac OS X) and is compliant with both 32 bits and 64 bits architectures (IA32, G4, IA64, G5, MIPS). All Kaapi related material are available at `http://www-apache.imag.fr/software` under GPL and LGPL licences.

6.3 CacheFLow: Cache–based approaches for speeding-up applications

Web–caching is well known method for reducing network latency, network traffic and server load. The caching technique in general has been recognized as an effective method for reducing computation. Currently, cache applications vary from Internet technologies, including for example *point–to–point* systems, to data bases and large computational systems.

CacheFlow is a library that will use a memorization mechanism based on cache techniques in order to eliminate redundancy of computations in a parallel environment.

The fundamental problem is to identify a computation in order to determine if it has already been performed. We solve this problem using a description of the macro dataflow graph.

We perform the identification at a procedure call level, identifying the logical name of the procedure and the value of its actual parameters. When a procedure completes, his arguments and results are stored in the cache. When a procedure call becomes ready, his identification is searched in a hashtable that stores all elements in the cache. If an occurrence is found, then results in the cache are assigned to the result of the new procedure call, eliminating this redundant call.

7 Applications

7.1 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 Net Juggler and SoftGenLock. Different actions will be considered:

- Multi-modal applications. An on going 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 [10].

- 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

[10] E. Boyer and J.-S. Franco. A Hybrid Approach for Computing Visual Hulls of Complex Objects. In *Proceedings of CVPR'03*, volume I, pages 695–701, 2003.

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.

7.2 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 on going 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 make possible parallel data transfers.

- Cape-OPEN application. An on going collaboration with IFP (Institut Franç ais du Pétrole) to study an high performance CAPE (Computer Aided Process Engineering) runtime for cluster architecture. CAPE-OPEN is a 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.

7.3 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 and estimation of the tree must be performed. Due to its 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.

7.4 FlowCert

The RAGTIME project (Région Rhône-Alpes) uses a grid architecture for management of medical information that involves huge distributed databases. In the context of medical applications, integrity of the results is critical.

FlowCert uses the representation of the execution by its related macro dataflow (such as the one provided by Kaapi) to tolerate node resilience while providing probabilistic certification of correctness.

On large scale architectures, such as global computing platforms or grid systems, node failures or disconnections are frequent events. To prevent from resilience, FlowCert implements a checkpoint/restart mechanism. To encompass parallel computations with dependencies, FlowCert consists in an asynchronous distributed systematic storage of the macro dataflow graph that represents tasks (identifier and parameters) and their data dependencies.

Furthermore, the use of remote resources raises the issue of the certification of the output results. If grid middlewares, such as Globus, provide some security services (authentication, integrity of communication), they do not supply a certificate of the execution, i.e. a certification of the result integrity. Then, using the macro dataflow as a certification track, FlowCert performs reexecution of randomly chosen tasks to provide a probabilistic Monte Carlo test of integrity.

8 Collaborations, Platforms and Contracts

8.1 Collaborations with the INRIA project MESCAL

MOAIS has close relations with the INRIA MESCAL project that focuses on middleware systems for clusters and grids. Within MOAIS, we take an active part in the development of the scheduler of the OAR batch system developed by the INRIA project MESCAL.

OAR emphasizes on simplicity, extensibility, modularity, efficiency, robustness and scalability. It is based on an unprecedented high level conception that reduces drastically its software complexity. Its internal architecture is built on top of two main components: a generic and scalable tool for the administration of the cluster (launch, nodes administration, ...) and a database as the only way to share information between its internal modules. Written in Perl, OAR is also extremely modular making it straightforward to extend. Thus, it constitutes a privileged platform to develop and evaluate several scheduling algorithms and new kinds of services.

MOAIS contributes to the scheduler of OAR: it is a batch scheduler that should meet classical requirement such as priority based scheduling, management of jobs by queues, node reservation and backfilling. Furthermore, current development in OAR focuses on its extension to Grids; then, the scheduler is getting more complete and also more complex. Examples of required improvements are the support for best effort jobs, advanced policies, heterogenous resources and multilevel scheduling. Our expertise in batch scheduling, guaranteed approximation methods and new tasks models is a key point in the success of the

OAR scheduling. This is also a privileged context for real world experimentations on theoretical advances.

8.2 Experimental Platforms

8.2.1 GrImage

The MOAIS, Movi, Evasion and Artis projects are collaborating to install and operate at the INRIA Rhône-Alpes an experimental platform for high performance interactive applications, called **GrImage**.

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 with 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 modeling
- 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 was funded in 2003 by the INRIA and the Ministère de la Recherche (via INPG). We expect a second funding in 2004 from the INRIA and the Ministère de la Recherche. GrImage will eventually include 16 projectors, 20 digital cameras and 28 PCs.

8.2.2 Clusters and Grids

MOAIS is also involved in the experimental Platform Icluster2 (Itanium cluster located at INRIA Rhône-Alpes), the CIMENT regional grid and the project of national grid called grid 5000.

8.3 Contracts

- RNTL GEOBENCH (2003-2004).

  The goal is to Study PC cluster oriented solutions for immersive visualization and hatpic rendering applied to geo-scientific data. Partners : the

INRIA i3D group, the LIFO of the university of Orléans, the CEA, the BRGM and the TGS company.

- ProBayes and Pixellis 2003–2004. The company ProBayes distributes a probabilistic inference engine called PL. The kernel of the system consists in the recursive evaluation of a real function on the nodes of a tree. For most applications, this tree is of large size. To decrease computation times, MOAIS participates to the optimization of the code within a contract with the society Pixellis. Taking benefit of Pentium extended arithmetic co-processor, Pixellis realizes the sequential optimization of the arithmetic part; MOAIS develops a fine grain parallelization. This parallelization uses FlowStack software to schedule recursive tree computations on a cluster of symmetrical multi-processors.

- ACI Grid *GRID2* (2002-2003).

  MOAIS is in charge of an animation project which aims to organize collaboration between research teams, and workshops on the following research topics: "architecture of softwares and languages", "runtime support and middleware", "models and algorithmic", "algorithmic and application". Partners are CCH (Nancy), IRISA (Rennes), LaBRI (Bordeaux), LAMI (Evry), LIFL (Lille), LIP6 (Lyon), LIRMM (Montpellier), and LRI (Paris).

- Action ACI Grid *Projet DOC-G* 2002-2004.

  The MOAIS project is involved in the action DOC-G funded by ACI-GRID to exploit a grid architecture to solve challenging problems in combinatorial optimization. Partners are PRISM (Versailles) and LIFL (Lille). Two main applications are considered: quadratic assignment (PRISM) and telecommunications antenna mapping (LIFL). MOAIS contributes to efficiently schedule the computation tasks on a grid while ensuring fault-tolerance.

- European *Network of Excellence CoreGRID* started in 2004 and supported by *ERCIM*.

  This Network of Excellence aims at building a European-wide research laboratory that will achieve scientific and technological excellence in the domain of large scale distributed, GRID, and Peer-to-Peer computing. It is the primary objective of the CoreGRID Network of Excellence to build solid foundations for GRID and Peer-to-Peer (P2P) computing both on a methodological basis and a technological basis. This will be achieved by structuring research in the area, leading to integrated research among experts from the relevant fields, and more specifically distributed systems and middleware, programming models, knowledge discovery, intelligent tools, and environments.

- MOAIS researchers are involved in the CIGRI project (ACI GRID).

This project started in 2002 and aim at designing and developing tools for managing Monte carlo applications that are studied within the regional interdisciplinary group CIMENT.

- Project RAGTIME (Région Rhône-Alpes) 2003–2006.

  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 of the certification of results from remote execution.

- Cifre contract with ST Microelectronics 2003–2006.

  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 deal 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.

- Cifre contract INRIA-IFP 2003–2006.

  A collaboration with the company IFP (Institut Français du Pétrole) and APACHE project funds a PhD student on the research area of code coupling of software components for high performance computing. IFP has work of the standard CAPE-OPEN which allows to build application by coupling components. In order to decrease the runtime of the execution it should be able to use parallel architecture. The goal of the thesis is to study code coupling methods and scheduling algorithm for these components using the experience of Athapascan and Homa tools.

- RNTL OCETRE, (2004-2005).

  The RNTL project GEOBENCH associates the APACHE and MOVI project, the companies Total Immersion and Thalès ST. The goal is to develop solutions for real time 3D modeling with a PC cluster and multiple cameras for acquisition.

- ACI Masse de Données CYBER II (2004-2006).

  Real time motion capture, 3D reconstruction and inclusion of a character in a virtual world. Partners : the projets MOVI and ARTIS (INRIA Rhône-Alpes) and the LIRIS laboratory, Lyon.

- ARC OTaPHe (2005-2006), directed by Frédéric Sutter (GRAAL). To federate theoretical and experimental research on parallel tasks scheduling on hjeterogeneous platforms. Experimentations will be performed on DIET and OAR.

8.4 Collaborations

Distributed and interactive simulation applications are developed in partnership with research teams specialist of the field:

- **Regionally.** physical models within the CIMENT project; clothes simulation on clusters with EVASION; real-time 3D-reconstruction from a network of cameras with MOVI; bioinformatics simulation and phylogeny trees classification with HELIX and the Rhône-Alpes Genopole; security on the grid within the CryptAlpes research group; participation to the GROG group in the IPI (Institut de la Production Industrielle);
- **Nationally.** molecular dynamics with SCALAPPLIX; participation to the GRD ARP through ORDO and Hiperf working groups; AS CNRS on Grid computing; Middleware for large scale VR applications with LIFO, universté d'Orléans.
- **Internationally.**
  - coordination of a bilateral franco-marocco AI-MA01-19 project (2001-2004);
  - coordination of a bilateral franco-german PROCOPE project (2002-2004);
  - coordination of a franco-brazilian project with UPSP (Sao Paulo) (2002-2004);
  - security certification problems on the grid are considered in collaboration with Franck Leprévost and Pascal Bouvry at Université du Luxembourg (joined direction of the Ph-D thesis of Sébastien Varrette).
  - coordination of the bilateral Franco-Tunisian INRIA-DGRSRT project (2005-2006) on "Modelization and deployment of large scale parallel systems" (joinde direction by Denis Trystram from France and Mohamed Jemni for Tunia). French participants: Paul Featrier, Nahid Emad, Christophe Cérin.

8.5 Animation of Academic Community

8.5.1 Event Organization

- **EGPGV04.** Bruno Raffin is co-chair and local organizer of the Eurographics Symposium on Parallel Graphics and Visualization that will take place in Grenoble, June 8-9th, 2004.

- **PCS'04 workshop.** Denis Trystram is co-chair of the next edition of this workshop to be held in Colima, Mexico in September 2004.
- **Workshop on Scheduling for Computer and Manufacturing Systems** Denis Trystram is co-chair of the next edition of this workshop, dedicated to the 70ty birthday of Ed Coffman in May 2004.
- **HETEROPAR'04** Denis Trystram is co-chair of the next edition of the workshop on heterogeneous aspects of computing in Dublin in july 2004.
- **PAAM'04** Denis Trystram is co-chair of the next edition of the worshop on Parallel algorithms and Applications in Mathematics in october 2004.
- **WASC'04** Denis Trystram is co-chair of the 1st workshop on Algorithms and Scheduling in Bertinoro in july 2004.

8.5.2 Teaching

- **Master ISC Grenoble.** Jean-Louis Roch gives the course entitled *Parallel Computation: Algorithmic and Fundamental Techniques* at the Ecole Doctorale Mathématiques&Informatique of Grenoble.
- **Master CSCI (joined INPG-UJF) Grenoble.** Jean-Louis Roch is co-director (director for INPG) of the joined INPG-UJF Master *Cryptology, Security and Information Coding* at the Ecole Doctorale Mathématiques & Informatique of Grenoble.
- **DEA Informatique fondamentale d'Orléans.** Bruno Raffin give every year several lectures about parallel architectures and virtual realities at the DEA Informatique Fondamentale at the university of Orléans.
- **Siggraph Tutorial.** Bruno Raffin co-roganized with Hank Kaczmarski, University of Illinois, and Marcelo Knorich Zuffo, University of São Paulo, a course about Virtual Reality Clusters at the Siggraph conferences 2002 and 2003. A third edtion of this course has been submitted for Siggraph 2004.

9 Recent Publications [2002–2004]

PhD and "Habilitation" Theses

[1] G. Parmentier, *Une approche générique pour l'alignement multiple et la reconstruction de phylogénies*, Thèse de doctorat, Institut National Polytechnique de Grenoble, décembre 2003.

[2] E. Romagnoli, *Exploitation efficace de grappes dynamiques de PC indifférenciés pour le calcul paralléle*, Thèse de doctorat, Institut National Polytechnique de Grenoble, décembre 2003.

[3] F. Zara, *Algorithmes paralléles de simulation physique pour la synthése d'images: application à l'animation de textiles*, Thèse de doctorat, Institut National Polytechnique de Grenoble, décembre 2003.

Articles

[4] A. Darte, G. Huard, « New Complexity Results on Array Contraction and Related Problems », *Journal on VLSI Signal Processing*, 2003.

[5] J.-G. Dumas, J.-L. Roch, « On parallel block algorithms for exact triangularization. », *Parallel Computing 28*, 2002, p. 1531–1548.

[6] P.-F. Dutot, « Complexity of Master-slave Tasking on Heterogeneous Trees », *European Journal on Operationnal Research*, 2003, To appear.

[7] A. Goldman, G. Mounie, D. Trystram, « 1-Optimality of static BSP computations: scheduling independent chains as a case study », *Theoretical Computer Science*, 290, 2003, p. 1331–1359.

[8] A. Goldman, D. Trystram, « Efficient parallel algorithm for solving the Knapsack problem on hypercube », *Journal of Parallel and Distributed Computing - JPDC*, to appear.

[9] A. Gupta, G. Parmentier, D. Trystram, « Scheduling precedence task graphs with disturbances », *RAIRO Operational Research*, 2003, to appear.

[10] R. Lepere, G. Mounie, D. Trystram, « An Approximation Algorithm for Scheduling Trees of Malleable Tasks », *European Journal of Operational Research*, 142, 2002, p. 242–249.

[11] R. Lepere, D. Trystram, G. Woeginger, « Approximation Scheduling For Malleable Tasks under Precedence constraints », *International Journal of Foundation in Computer Science 13*, 4, 2002, p. 613–627.

[12] F. Zara, F. Faure, J.-M. Vincent, « Parallel Simulation of Large Dynamic System on a PCs Cluster: Application to Cloth Simulation », *Special issue on cluster/grid computing in International Journal of Computers and Applications (IJCA)*, March 2004.

Book Chapters

[13] P.-F. Dutot, G. Mounie, D. Trystram, *Scheduling Parallel Tasks: Approximation algorithms*, to appear april 2004.

[14] T. Gautier, H. Hong, J.-L. Roch, W. Schreiner, *Handbook of Computer Algebra – Foundations, Applications, Systems*, Springer Verlag, Heidelberg, 2002, ch. Parallel implementation.

Conference and Workshop Publications, etc.

[15] J. Allard, M. C. Cabral, C. Goudeseune, H. Kaczmarski, B. Raffin, B. Schaeffer, L. Soares, M. K. Zuffo, « Commodity Clusters for Immersive Projection Environments », California, July 2003.

[16] J. Allard, V. Gouranton, G. Lamarque, E. Melin, B. Raffin, « Softgenlock: Active Stereo and Genlock for PC Cluster », *in : Proceedings of the Joint IPT/EGVE'03 Workshop*, Zurich, Switzerland, May 2003.

[17] J. Allard, B. Raffin, F. Zara, « Coupling Parallel Simulation and Multi-display Visualization on a PC Cluster », *in : Euro-par 2003*, Klagenfurt, Austria, August 2003.

[18] J. Blazewicz, M. Kovalyov, M. Machowiak, D. Trystram, J. Weglarz, « Exact algorithms for scheduling Malleable Tasks », *in : EURO - INFORMS*, Istanbul, Turkey, july 2003.

[19] J.-G. Dumas, T. Gautier, M. Giesbrecht, P. Giorgi, B. Hovinen, E. Kaltofen, B. Saunders, W. Turner, G. Villard, « Linbox: a Generic Library for Exact Linear Algebra », *in : Proceedings of ICMS'2002 : International Congress of Mathematical Software*, Beijing, China, août 2002.

[20] J.-G. Dumas, T. Gautier, C. Pernet, « Finite Field Linear Algebra Subroutines », *in : Proceedings of ISSAC'2002: International Symposium on Symbolic and Algebraic Computations*, Lille, France, juillet 2002.

[21] P.-F. Dutot, « Ordonnancement de tâches identiques sur réseau hétérogène », *in : École thématique sur la globalisation de ressources informatiques et des données*, INRIA, p. 375–384, December 2002.

[22] P.-F. Dutot, « Master-slave Tasking on Heterogeneous Processors », *in : International Parallel and Distributed Processing Symposium*, IEEE Computer Society Press, April 2003.

[23] Ł. Garstecki, « Generation of conformance test suites for parallel and distributed languages and APIs », *in : Eleventh Euromicro Conference on Parallel, Distributed and Network-Based Processing*, IEEE, p. 308–315, 2003.

[24] T. Gautier, H. Hamidi, « HOMA: un compilateur IDL optimisant les communications des données pour la composition d'invocations de méthodes CORBA », *in : Proceedings des Rencontres Francophones du Parallélisme (RenPar'15)*, p. 127–134, La Colle sur Loup, France, 2003.

[25] S. Jafar, J.-L. Roch, « Fault-Tolerance for Macro Dataflow Parallel Computations on Grid », *in : ICCTA'04 IEEE Conference on Information&Communication Technologies: from Theory to Applications*, Damascus, Syria, april 2004.

[26] A. Kerfali, J.-L. Roch, E. M. Daoudi, « Algorithmes parallèles à grain adaptatif - Application à la parallélisation de gzip », *in : RENPAR'15*, p. 18–26, Nice, France, octobre 2003.

[27] A. Mahjoub, C. Rapine, D. Trystram, « Influence of starting solutions on the stabilization of scheduling algorithms », *in : EURO - INFORMS*, Istanbul, Turkey, july 2003.

[28] N. Maillard, E. M. Daoudi, P. Manneback, J.-L. Roch, « Contrôle amorti des synchronisations pour le test d'arrêt des méthodes itératives. », *in : RENPAR'14*, p. 177–182, Hammmamet, Tunisie, Avril 2002.

[29] R. Revire, F. Zara, T. Gautier, « Efficient and Easy Parallel Implementation of Large Numerical Simulation », *in : Proceedings of ParSim03 of EuroPVM/MPI03*, Springer (réd.), p. 663–666, Venice, Italy, 2003.

[30] A. Tchernykh, D. Trystram, « On-line scheduling of multi-processor jobs with idle regulation », *in : PPAM, fifth International Conference on Parallel Processing and Applied Mathematics*, Czestochowa, Poland, 7-10 september 2003.

[31] S. Varrette, J.-L. Roch, « Certification logicielle de Calcul Global avec dépendances sur grille », *in : Proceedings des Rencontres Francophones du Parallélisme (RenPar'15)*, M. Auguin, F. Baude, D. Lavenier, M. Riveill (réd.), p. 169–176, La-Colle-Sur-Loup, France, 15–17 Octobre 2003.

[32] F. Zara, F. Faure, J.-M. Vincent, « Physical cloth simulation on a PC cluster », *in: Fourth Eurographics Workshop on Parallel Graphics and Visualization 2002*, X. P. D. Bartz, E. Reinhard (réd.), p. 105–112, Blaubeuren, Germany, September 2002.

[33] F. Zara, J.-M. Vincent, F. Faure, « Coupling Parallel Simulation and Parallel Visualization on PC Clusters », *in: Commodity Cluster for Virtual Reality 2003, VR 2003 Workshop*, Los Angeles, USA, March 2003.

Technical Reports

[34] Ł. Garstecki, « CTS–Designer tool for effective functional conformance testing of parallel and distributed programming libraries », rapport de recherche, Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Department of Knowledge Engineering, 2003.

[35] T. Gautier, R. Revire, J.-L. Roch, « Athapascan: API for Asynchronous Parallel Programming », rapport de recherche n° RR-0276, APACHE, INRIA Rhône-Alpes, February 2003.