INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team apache

Algorithmique Parallèle, Programmation et Répartition de Charge

Rhône-Alpes

THEME 1A

Table of contents

1. Team

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

Head of project team

Brigitte Plateau [Professor]

Administrative staff

Anne-Laure Binder [INPG Administrative Assistant INPG, half-time]
Valérie Fené [INPG Administrative Assistant INPG, half-time]
Marion Ponsot [INRIA Administrative Assistant, half-time]
Annie-Claude Vial d'Allais [CNRS Administrative Assistant, half-time]

INRIA Staff

Thierry Gautier [Research Scientist]
Bruno Raffin [Research Scientist]
Said Oulahal [Engineer, half-time]

CNRS Staff

Philippe Augerat [Engineer, leaving 1/1/2004]
Joëlle Prévost [Engineer, leaving 1/10/2003]

INPG Staff

Yves Denneulin [Assistant Professor]
Grégory Mounié [Assistant Professor]
Brigitte Plateau [Professor]
Jean-Louis Roch [Assistant Professor]
Denis Trystram [Professor]

UJF Staff

Jacques Briat [Assistant Professor]
Guillaume Huard [Assistant Professor]
Jean-François Méhaut [Professor]
Olivier Richard [Assistant Professor]
Jean-Marc Vincent [Assistant Professor]

Project technical staff

Wilfried Billot [INRIA, 9/01-9/03]
Nicolas Capit [ACI Grid- 12/03-12/05]
Aurélien Dumez [INRIA, 9/03-9/05]
Christian Guinet [RNTL E-Toile, 10/02-11/03]
Loïck Lecointre [RNTL Geobench - 03/03-05/05]
Julien Leduc [RNTL CLIC, 9/02-11/03]
Olivier Lobri [RNTL CLIC, 9/02-9/03]
Stéphane Martin [HP, then CLIC, 02/01-11/03]
Pierre Neyron [RNTL CLIC, 09/02-11/03]
Stéphane Perret [RNRT SIDRAH, 02/03-08/03]

Invited Scientist

Paulo Fernandes [PUC University, Porto Alegre, Brazil, 1 month]
Philippe Navaux [UFRGS University, Porto Alegre, Brazil, 2 weeks]
William Stewart [North Carolina University, Raleigh, USA, 4 weeks]
Andreï Tchernyk [CICESE, Ensenada, Mexico, 1 month]

PhD students

Jérémie Allard [2002, MRNT scholarship]

Anne Benoit [2000, MRNT scholarship]
Florent Blachot [2003, CIFRE ST Micro Electronics scholarship]
Damien Croizet [2003, France Telecom scholarship]
Georges DaCosta [2001, Normalien, MRNT scholarship]
Pierre-Francois Dutot [2000, Normalien, BDI-CNRS MRNT scholarship]
Luis-Angelo Estefanel [2002, Brazilian CAPES scholarship]
Euloge Edi [2000, Ivorian scholarship]
Lionel Eyraud [2002, Normalien, MRNT scholarship]
Estelle Gabarron [2003, CIFRE BULL scholarship]
Cyril Guilloud [2000, INRIA, contrat BULL MRNT scholarship]
Hamidi Hamid Reza [2001, SFERE scholarship]
Samir Jafar [2002, Syrian scholarship]
Sirak Kaewjamnong [2003, Thai scholarship]
Adrien Lebre [2002, INRIA scholarship, Bull contract]
Nhien-An Le-Khac [2000, EGIDE, co-tutelle]
Pierre Lombard [2000, BDI-CNRS scholarship]
Garstecki Lukasz [2001, Polish scholarship, co-tutelle]
Corinne Marchand [2001, INRIA scholarship - France Telecom contract]
Cyrille Martin [2000, CIFRE BULL scholarship]
Maxime Martinasso [2003, CIFRE BULL scholarship]
Clément Ménier [2003, Normalien, common to MOVI and APACHE]
Jean-Michel Nlong [2002, INRIA scholarship , HP contract]
Gilles Parmentier [1999, MRNT scholarship]
Laurent Pigeon [2003, CIFRE IFP scholarship]
Jonathan Pecero-Sanchez [2003, CONACYT Mexican scholarship]
Mauricio Pillonl [2000, Brazilian CAPES scholarship]
Rémi Revire [2000, MRNT scholarship]
Bruno Richard [2000, HP engineer]
Emmanuel Romagnoli [2000, CIFRE HP scholarship]
Ihab Sbeity [2003, MRNT scholarship]
Olivier Valentin [2003, MRNT scholarship]
Jesus Verduzco [2001, Mexican scholarship]
Eiad Sulaiman [2002, Syrian scholarship]
Florence Zara [2000, MRNT scholarship]
Jaroslaw Zola [2002, Polish scholarship, co-tutelle]

2. Overall Objectives

Hardware and software technologies allow to aggregate an apparently illimited number of computing elements, storage devices, and peripherals. These technological steps make accessible anywhere the mondial stock of textual, visual and acoustic information via intelligent and efficient indexing services through a hierarchy of cache servers. A geographical taxonomy is frequently used to distinguish the different systems available for intensive computation:

cluster : the cluster issue is to use hundreds to thousands of interconnected PCs to deliver a computing power comparable to a supercomputer at a much lower cost. Technical relevant problems are basically related to distributed systems middleware (initialization, configuration, protection and sharing, etc.), to parallel programming and algorithms and to scheduling of tasks and resource allocation. A slight extension of a cluster is to consider that the various hardware is nomadic within an Intranet: the infrastructure is no more homogeneous nor static and the challenge is to benefit from temporarily unused hardware for large computations.

grid : the grid challenge is to use as a single cluster several distributed clusters available within an organization. Mastering grids requires to solve critical security problems and to manage the efficient connexion of heterogeneous hardware. Another aspect is the intelligent coupling of applications and the interactive visualization of large amount of data.

global computing or peer to peer computing : Based on the observation that the computer utilization on the internet is very low, the challenge is then to use this potentially illimited power for solving very large problems. For example, the *seti@home* project gathers millions of computers for the analysis of stars radiations. The idea is that these resources should be available on demand, like electricity. A key point to run such computing systems is automatic plug and run protocols and efficient algorithms for resources discovery, localisation and acquisition. At last, peer to peer applications should be resistant to a large variety of failures.

In the APACHE project, we are interested by management and programming tools for such high performance computing infrastructures: clusters, cluster on an Intranet, grids, global computing and peer to peer computing. The following paragraph gives a short glance at major achievements of the project.

In what concerns parallel programming, our approach is to promote a programming model based on asynchronous tasks with shared data access rules. The programming environment ATHAPASCAN, based on these ideas, allows the automatic computation of an abstract representation of the program (macro-dataflow graph) and an automatic load distribution of computing tasks and data placement, usable on a wide variety of computing platforms (cluster and grids). Load sharing is based on scheduling and mapping algorithms on which the project has a thorough expertise. The task scheduling manager of ATHAPASCAN has been used in industrial applications (program verification) to run parallel programs defined by a dynamic script-shell. Key applications are interactive visualization applications: NETJUGGLER allows to distribute the visualization on several tens of screens of video projector and to interact with the visualization. Following this line of thought, SAPPE allows to visualize an interactively moving piece of material. The objective is to combine image capture, computing (analysis of the image), and interactive visualization of the results. The main difficulty is the handling of various synchronizations under heavy computing load and with human interaction.

A runtime kernel INUKTITUT, able to cope with heterogeneous standards, allows to deploy dynamic sets of lightweight communicating processes and an object memory. The originality of this kernel is to be scalable and based on a high level API. It is used to implement ATHAPASCAN as well as HOMA, a platform for the parallel execution of CORBA computing components where research is conducted to obtain efficiency through interleaved and parallel communications. All these environments have been ported (and used) on various computing plaforms (Cray T3E, IBM SPx and SGI Origin 2000 and 3800, PC clusters) and on a new Itanium cluster.

In what concerns management tools, the project focuses on the scalability aspects: the KA TOOLS allow to deploy OS, code, files on hundreds of PCs by using appropriate application, dependent spanning trees and neighbour-to-neighbour protocols. These tools are now commercialized by Mandrake with the CLIC cluster suite. Another area of interest is the discovery and allocation of resources within an Intranet with a dynamic set of resources and the usage of a cluster as a storage device in order to allow high speed transfer between cluster by using communication interleaving and node parallelism. All these algorithms are based on thorough performance evaluations and measurements.

These research directions and results are part of collaborations with the companies BULL, HP, Mandrake, BRGM, France Telecom.

3. Scientific Foundations

3.1. Parallel algorithms, complexity and scheduling

Key words: *algorithms*, *complexity*, *modelling*, *scheduling*, *performance evaluation*.

In the field of parallel algorithms, there is no concensus on efficiency and there exists numerous models of complexity. On a given problem, once the parallelism is extracted and the execution is described as a dynamic task graph, the problem is to schedule this task graph on the resources of the parallel architecture. This motivates theoretical studies: to design algorithms whose related task graphs can be scheduled on various architectures with proved bounds is a main research axis; to provide scheduling algorithms suited to machine models; to develop quantitative models of parallel executions.

3.1.1. Algorithms and complexity

Since the 80's, the design of efficient parallel algorithms is an active field. Various algorithmic techniques have been developed in order to provide successful parallelization of challenging problems, from pipelining to cascading. In order to reach portability, various models have been proposed to abstract the underlying resources.

Primitive models (such as PRAM or the uniform circuits) were based on an arbitrary number of tightly synchronized identical processors, leading to characterized algorithms in terms of total number of operations T_1 and critical path T_∞. In order to take into account communications, extensions (such as delay model, BSP or LogP) have been proposed in order to provide a more realistic abstraction of a distributed architecture.

However, those models are difficult to emulate on realistic large scale parallel architectures from a cluster including several hundreds of symmetrical multi-processors nodes to a grid of clusters. On such architectures, memory accesses are non-uniform and the memory hierarchy plays a crucial role.

Also, the algorithmic model which remains the most successful for the design of portable algorithms is the uniform task graph; it abstracts the execution of the algorithm on a given input by representing both computations and their dataflow dependencies. >From basic representations such as precedence or dependency graphs, we have contributed to popularize more general representations: the dynamic dataflow graph and the malleable tasks with implicit communications models. Various cost annotations, such as number of operations, synchronizations, communications and remote memory access volumes, have been introduced in order to lead to more realistic complexity analysis on distributed architectures.

The emulation of the algorithm on a given architecture model is then reduced to the scheduling of its related task graph. Thus, algorithmic research focuses on the design of algorithms whose related task graphs may be provably efficiently scheduled on various architecture models.

3.1.2. Scheduling

Today, the theoretical studies in the field of parallel processing aim at characterizing the scheduling algorithms (worst case bounds for the execution time or the memory, optimality result, etc.) for realistic types of applications or parallel and distributed supports. We are working on the determination of adequate models, able to take into account the new characteristics of these supports (namely, hierarchy, heterogeneity, unbalance between communications and computations, etc). Many interesting results have been developed for the malleable tasks model and effective implementations have been proposed. Recently, extensions for taking into account simultaneously multiple objectives have been studied. As an example of multicriteria constraints, we consider the scheduling to optimize time completion while bounding the required memory space. By considering programs for which a sequential schedule is implicitly defined, it is possible to bound the memory space with respect to the one of this sequential schedule. Such a multi-objective scheduling has then a generic solution that we have applied to the scheduling of dynamic dataflow graphs.

For the problem of implementing parallel applications, it is possible to design very good scheduling algorithms by specializing general techniques to a given class of applications with particular properties. In the context of on-line scheduling, list algorithms lead to asymptotically optimal results for regular applications. In the context of more realistic use, more sophisticated algorithms have to be designed for efficient implementations.

3.2. Runtime for parallel and distributed applications

Key words: *programming model*, *runtime*, *threads*, *message passing*, *remote memory access*.

Runtime softwares implement a parallel architecture model or a parallel virtual machine. They aim at offering an easy to use interface to access computing, storage and communication resources. A required capability for a runtime is its ability to efficiently exploit the parallelism of a parallel or distributed architecture. The widely adopted approach consists in using threads to exploit both physical parallelism, as in a SMP and a fully distributed architecture.

Another important characteristic required from a runtime is its ability to scale from clusters of hundreds of nodes to grids of thousands of nodes or millions over the Internet. Another aspect is the robustness regarding the heterogeneity of computing and network resources and their availability. Scaling to such a large number and the high heterogeneity of resources imply using efficient tools to broadcast files, start applications or collect results on thousands of nodes. Implementing global computing on the Internet requires proper tools to discover available computing resources.

The two most frequently used tools are:

MPI simulates a network of uniprocessors communicating using operators suited to scientific computing. It works well on distributed architectures.

Posix threads simulates a multiprocessor with shared memory and is limited to architectures with shared physical memories.

The commonly used approach is to simulate a network of multiprocessors. The interactions between the nodes can simply be seen as remote procedure calls, active messages, and remote memory accesses. Close to the communicating processes paradigm, this method inherits of its advantages (programming model, portability) while offering an improved efficiency and more flexibility in scheduling communications and computations. Indeed a key element in parallelization is *l*ocality, i.e. bringing a couple computation-data closer to a couple processor-memory. Dynamic remote creation of threads gives the possibility to implement dynamic load balancing.

Operating such kernels started on homogeneous architectures like clusters. Nowadays the challenges are in scaling, mastering heterogeneity and dynamic configurations. Scaling highlights critical points in the design of such kernels. They concern the efficiency of launching a parallel application on hundreds of nodes as well as broadcasting code and data. These operations have to be parallelized in a way that is well suited to the architecture. Passing from clusters to grids or the Internet requires to manage qualitative (nodes, languages, protocols) and quantitative (CPU, bandwidth, ...) heterogeneity. The runtime kernel has to adapt to dynamic resources configurations when using idle nodes inside an intranet or temporally available resources on the Internet.

3.2.1. Dynamic network of communicating threads

The technological context enforces distributed memory architecture whose nodes are shared memory multiprocessors, SMP. Memory access times are the same on a given node, but communication latencies between nodes are high compared to memory access time. Runtime kernels based on communicating threads favor the structuring of a parallel computation as a dynamic network of communicating processes where data and process placement is explicit. Processes on the same node communicate using shared memory. Between nodes they communicate using messages. Efficient use of such architectures requires to parallelize communication and computation, i.e. overlapping communications by computations. Using only asynchronous operators for communication enables the designer to improve overlap inside a process. Programming with threads allows to correct imperfect overlaps inside processes. An essential property of a kernel is its reactivity in front of unpredictable communication latency and its ability to avoid data copying.

3.2.2. Resource management

Currently, trends for high performance infrastructures mainly concern architectures of clusters and grids with a high number of resources. Resources and nodes can appear and disappear due to node failure or network connectivity problems. During a day, the risk of failure is not negligible with a cluster of several hundred of nodes. Inside a grid with a complex network topology, router and switch may fail and sites become

unreachable. At this level, the scientific challenge concerns the management of dynamic resources. We are experimenting such algorithms within a platform called OAR.

According to the new requirements of HPC users, the data storage is also an exciting challenge. Grids and clusters have a huge distributed space for data storage. As the disk capacity increases, most of the disk space remains unused for user data files. The new requirements for high performance applications concern huge amount of data, coherency level, fine-grained sharing. We study new approaches to improve disk throughput and to solve scalability issues. Rather designing a new protocol, we have chosen to enhance the NFS distributed file system protocol. The aim is to be able to quickly broadcast file to parallel application nodes in transparent manner. Technics like file stripping and replication will be included. Another goal is to compare the general push and pull approaches in file distribution. Push method can be view as selective broadcast, and pull one as transfer on demand. There will be an interest to determine situations where each one is the best method.

3.2.3. Scalability

The use of a cluster composed of hundreds of nodes shows the need for efficient global operations such as: application launcher, file transfert tools, and exploitation and operating commands. Fast installation (low level deployment) appears also as an important issue. All these operations are based on broadcast or diffusion communication algorithms. The key elements are a spanning tree which permits the network bandwidth exploitation (saturation), disjoint edges use as network parallelism and overlapping between communication and computing phases. We have observed that the best configuration mainly depends on the global operation which must be executed and network properties, characteristics and topology. The issue is to obtain a set of efficient solutions and a profiling tool which can determine the best strategy according to a particular context. In grids, at each level, software is concerned by scalability and thus can use the above techniques. The system installation of large-sized cluster may be a very long process if no facilities are provided to system administrators. The user also needs efficient middlewares to run parallel programs with their data files.

3.2.4. Virtual clusters

Many statistical analysis of Intranet networks show a very low ratio for resource usage. The outstanding point is that workstations and personal computers are mainly idle during the night (or week-end) and slightly used during the days. Users typically spend just a few hours a day using their machines, so there are idle periods where the resources such as processing power, memory and connectivity remain unused. The exciting challenge is to be able to use this huge set of resources for high performance computations. One of the key points is the design of an automatic protocol (plug&play) to discover and locate available resources and services in the Intranet. Another key point is the design of a protocol to reserve and allocate resources for computing jobs as resources may appear or disappear according to user behaviors or network connectivity. If it is clear that middlewares have to manage this dynamic behavior, this requires also new programming models with stopping and restarting mechanisms. The ATHAPASCAN model based on task graph should be suitable to support volatile resources.

3.3. Model and language for parallel and distributed computing

Key words: *parallel algorithmics*, *programming model*, *tasks graph*, *scheduling*, *mapping*, *load-sharing*.

Defining an application programming interface may have various objectives: portability of existing codes, automatic parallelization or scheduling automatization. Our effort targets this last scheduling objective, the user expressing the parallelism of his application in such a way to avoid any complex code analysis.

The effective availability of various parallel and distributed architectures (from super-scalar sequential machines to large size clusters of symmetrical multi-processor nodes and grids) motivates the definition of programming models that abstract the underlying architecture and its features. Of course, providing a semantic independent of the architecture enables code reutilisability and coupling (this is the case for coordination, skeleton or also component languages). This also enables efficient executions by specializing the scheduling of the application for the architecture. This is sometimes made by code annotations, like in HPF or Open-MP.

In this research axis, we study programming models for which scheduling can achieve provable bounds while making easier the development and the coupling of codes.

Since automatic parallelization of a sequential code is a difficult problem, most models are based on explicit parallelism; parallelism extraction then does not require a complex code analysis. This is for instance the case for Cilk [5] or Jade [9]. Like in logic programming language, an abstract interpretation technique is used to control the execution in order to ensure the semantics. The difficulty is then to minimize the overhead of this interpretation, especially on distributed architectures.

3.3.1. Programming model

Following the theoretical models presented in the section *Parallel algorithmics, complexity and scheduling*, we developed a programming model (ATHAPASCAN) based on the representation of an execution by an embedded bipartite graph which describes computational tasks and data dependencies between tasks.

While parallelizing compilers compute such a representation from sequential code analysis, an alternative approach is to precompute the graph by on-line interpretation of explicit parallel instructions. Parallelism is explicited at an arbitrary granularity and data dependencies precomputed by an abstract interpretation which unfolds at runtime the dataflow. Then, based on an analysis of this dataflow, a fine scheduling can be computed, taking into account not only resources idleness but also data locality. Furthermore the semantics is sequential: this is mainly used for efficient sequential degeneration on a bounded number of resources and to avoid memory space exhaustion. The programming model allows to express the coupling of codes with no loss of parallelism while ensuring data dependencies. The prototype ATHAPASCAN implements these idea. The current development aims at better performance.

3.3.2. Efficiency by specialization of the scheduling

The efficiency of a scheduling depends not only on the application (i.e. the graph related to the execution) but also on the architecture. The system is open: various scheduling strategies, which respects the interface of the graph management, may be plugged in, by code annotation.

The independence between the building of the graph and the scheduling strategy introduces an overhead. Moreover, some strategies are using additional information on the graph (e.g. estimated cost of a task, priority, locality) increases this overhead. This has motivated the specification and implementation of a default optimized work-stealing strategy in order to decrease the overhead.

3.3.3. Flow Control for Interactive Application

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 an application on a distributed architecture requires to develop approaches to balance the workload and the activation frequency of the different tasks. The goal is to optimize CPU and network resources to get as close as possible to the reactivity or level of details defined by the user. We are currently working on a extension of the Net Juggler software. The goal is to integrate parallel code coupling functionalities and flow control algorithms.

3.4. Tools for performance and debugging

Tuning parallel programs is one of the most important part in the development of parallel/distributed applications. In the context of the Apache project, research focuses on tools that help developers to optimize application performance. The aim is to provide a view of the parallel execution as precise as possible. Moreover, the tool should be dynamic such that one could navigate inside the visualization and get different "points of view" of the execution. To achieve this goal, our approach is to study software tracers and to analyze traces `post mortem`, through a model of execution, provides information on the causality of events, resources usage, distributed patterns,...

3.4.1. Modelling and performance

Quantitative evaluation of the behavior of parallel applications is known to be a difficult task. The difficulties are experimental for the observation and data analysis of complex traces, and theoretical for the prediction of behaviors according to parameters of the application. These systems are characterized by a large number of entities (processors, threads, tasks, messages,...) and complex interactions between these entities (synchronizations, communications,...). To observe a global behavior of the system, a huge number of events should be observed.

For program behaviour, the project focuses on the design and the development of software tracing tools. For prediction of behavior, modelling techniques have been applied and combined with numerical or simulation methods. Because of the temporal non deterministic aspect of parallel application and the lack of knowledge on execution times and latencies, models are expressed as multi-dimensional stochastic processes in continuous time with discrete state spaces. These states spaces are very large and have a complex structure that requires specific techniques to be solved numerically.

The main approach is to build Markov models [1] and to take into account the distributed structure of the system. Efficient numerical algorithms, based on the structure of the model, reduce the amount of memory and decrease the complexity of the computation.

Another approach that has been developed recently is based on the properties of trajectories of the system in the state space. The evolution of the system is described by stochastic evolution equations. Because the evolution of a parallel system is based on sequences of synchronizations between the entities, evolution equations are usually expressed by few operators like $(\max, +)$ [10]. Then a spectral analysis of these operators allows the computation of typical performance indexes like speedup, utilization of resources, conflict probability,...

3.4.2. Traces and performance

A software tracer [7], in the context of an environment based on threads and communications should identify all objects that are manipulated at the run-time level: manipulation of threads (creation and deletion), communication ports, synchronization variables,... On the other hand, these events should be related to resources that are used by the system. In particular, tracers should observe the origin of thread scheduling events and the history of communications (to capture the causality). Problems are then to identify the objects and the interactions, for example which thread is executed on which processor. Several difficulties remain such as the instrumentation of the scheduling of users threads on kernel threads. Finally, because traces are recorded in a distributed way, a tool for trace integration is developed. In this case, a global datation of events is built taking into account drifts and offsets of physical clocks.

3.4.3. Visualization and analysis

The source of a parallel application misbehavior could be various because of the integration of several levels that have been implemented and tested separately. So several levels of abstraction are needed and correspondence between these levels is of great importance. For example, a message latency at the application level could be produced by the application itself, by the running environment, the operating system or the hardware (processor/network,...). A good visualization tool should provide simultaneously a global and a very detailed vision of the execution. Indeed, three concepts should be emphasized: The extensibility of the tool should allow the modification of the behavior model (by adding another type of resource, synchronization,...); The interactivity should permit to explore traces by zooming in time, by investigating objects, by making statistics,..; The scalability of the tool should allow to increase the model size without modification of the visualization code.

4. Application Domains

4.1. Panorama

Participants: J. Allard, J. Briat, E. Euloge, T. Gautier, G. Parmentier, G. Mounié, B. Raffin, J.-L. Roch, D. Trystram, J. Verduzco, J.-M. Vincent, F. Zara.

Key words: *linear algebra*, *multi-alignment and phylogeny*, *cloth simulation*, *virtual reality*, *on-demand geographical map*.

Applications in the field of numerical simulation and image synthesis and processing are typical of the user demand for high performance computing. In order to confront our proposed solutions for parallel computing to real applications, the projet is involved in collaborations with end-users to help them to parallelize their applications.

4.2. Bio informatics

The project studies data processing applications (multi-alignment and phylogeny). Applications to phylogeny have complexity that comes from the combinatorial approach of the multi-alignment problem when processing very large data bases. As for numerical simulations, parallel computing is required to reduce computation time.

4.2.1. Multi-alignment and phylogeny

Research is conducted since October 1999 on new bioinformatics applications, namely multiple alignment of biological sequences and building of specie trees (called phylogenic trees). Our contribution uses 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.

Our objective is to propose an efficient solution to the multi-alignment problem. It is a basic problem in biology which is used in many applications like the construction of phylogenic trees. In the context of the PhD work of Gilles Parmentier, we developed a sequential code that is currently parallelized by Jeroslaw Zola. We will study the mechanisms for avoiding to recompute parts of the genetic code that have been already computed on the same subsets of sequences.

4.3. Images

Image synthesis and image processing are computer intensive tasks. The Apache project collaborates with the MOVI and EVASION projects of INRIA-RA, as well as with researchers in geography, to test distributed computation solutions with these applications. These applications impose strong execution time constraints to enable real time user interactions. The goal is to use parallelism to speed up computations, and to develop scheduling strategies taking into account task dependencies, tasks priorities and deadline execution constraints. Depending on the application, different mathematical solutions are used : statistical methods for image matching, differential equations for image synthesis or statistical smoothing for interactive construction of statistical data based maps.

4.3.1. Image Matching

This joint work involves the MOVI project (R. Horaud), Gravir-IMAG and INRIA-RA.

Dense image matching techniques are used for augmented reality applications. Let a set of images covering all the points of a real scene., then dense matching of spatially close images allows to build images for any given view-point by interpolation. The computation bottleneck is image matching that must be performed for 40 images per second in average. The goal is to study parallelization approaches for these algorithms to reach a real time image matching.

Different approaches have been considered for SMP and cluster architectures. Results underline the importance of the quality of the result obtained. It appears that parallelism enables to rapidly reach a good partial solution, compared to complete solutions obtained with sequential algorithms. Another important aspect that appeared through the different tests performed, is the application high irregularity. This feature leads us to develop dynamic load balancing strategies. Results about this work are expected shortly.

4.3.2. 3D Reconstruction

This joint work involves the MOVI project, Gravir-IMAG and INRIA-RA.

One approach for augmented reality consists in using multiple fixed cameras to capture a scene from different points of view. The goal of 3D reconstruction is to use these data to build a 3D model of the observed objects.

This 3D model can next be injected in a 3D application. 3D reconstruction is a highly intensive computation task that must be performed in real time (30 times per second). The goal of this collaboration is to develop parallel visual hull reconstruction algorithms that can efficiently run on a PC cluster. The GrImage platform will be used for testing.

4.3.3. On-demand Geographical Maps

This joint work involves the UMR 8504 Géographie-Cités and the Maisons de l'Homme et de la Société.
Geo-statistical data representation, like demographical or economical data, are computation intensive tasks. A single map represents a synthesis of statistical data. If, for instance, one wants to represent the population density on earth from geo-referenced data, the technique used consists in choosing a smoothing radius R and in computing for any point on earth the population at a distance of R km around each point. The geographers would like to interactively tune up the R parameter to observe the influence of the smoothing radius on the representation to identify properties about the observed population. These operations involve highly intensive computations and could benefit from parallel execution approaches.

4.3.4. Cloth Simulation

This joint work involves François Faure (Evasion-GRAVIR and INRIA-RA).
3D real time rendering of Cloth simulation is a a challenging problem for 3D real time rendering of dressed animated characters. For a realistic rendering, fundamental elements of physics, like velocity, forces (gravitation, etc.), are used to model the movements of several interacting objects. The goal of our work is to decrease the computation time to reach real time animations.

4.3.5. Virtual Reality

This joint work involves Emmanuel Melin and Valérie Gouranton (LIFO - Université d'Orléans)
Multi-projectors virtual reality environments, like Cave or workbench systems, are usually driven by dedicated multi-processor computers (SGI Onyx for instance). The goal of this work is to develop solutions to switch from high end machines to low cost PC clusters. Beside a cost advantage, PC clusters are scalable, modular and support a very large range of interfaces. In a second step, the goal is to take advantage of PC cluster scalability to run large virtual reality applications that require a processing power that today's dedicated machines cannot provide.

5. Software

5.1. Inuktitut kernel

The use of clusters, grid or unused personal computers connected to Internet for intensive computing imply extension and adaptation of communication kernel and low-level middleware. The transition from cluster to grid and from grid to Internet leads to address two kinds of heterogeneity: qualitative (computer, language, protocol) and quantitative (computing performance, network capacity...). Our approach consists in developing a generic C++ library based on communicating multithread model for all protocols. INUKTITUT can be viewed as a simplification of the previous ATHAPASCAN-0 kernel interface (with generic aspect) and an extension to support common protocols and low-level middlwares as TCP/IP, MPI and CORBA. This kernel will support also high performance networks (with high bandwidth and low latency) such as Myrinet, SCI or InfinyBand. Finally, all or a subset of these protocols could be supported simultaneously within a parallel application.

INUKTITUT is a generic C++ library with the following features. Its originality lies in its ability to cope with heterogeneity and scalability and is opened to manage dynamic architectures.

Multithread We have not designed a new genuine API (*Application Program Interface*) for threads. We designed a Java's thread like an interface in C++ language. It includes synchronized objects, objects threads and a minimal garbage collector. It allows to exploit multiprocessor platforms and overlapping between communications and computing phases.

Communication It is based on the active message model, which is known to be a simple interface to implement on different networks and protocols and be easy to extend to support application protocols. Supporting a new protocol consists in transforming its actions in services triggered by active messages.

On top of INUKTITUT is built a parallel launcher, called Taktuk, and a library of fundamental algorithms, called ICS:

Taktuk: is a tool to launch or deploy efficiently parallel applications on large clusters, and simple grids. The deployment includes on one hand the launch of the parallel program on all nodes and on the other hand the setting up of a communication layer. Efficiency is obtained thanks to the overlay of all independent steps of the deployment. The good properties, performance and robustness of this tool are demonstrated by its use in several projects like: the Clic Mandrake Cluster Linux distribution, OAR (Job manager) and ATHAPASCAN.

ICS: is a library of fundamental algorithms in parallel and distributed computing on top of INUKTITUT. ICS implements classical parallel algorithms for broadcast and reduction (flat-tree, alpha-tree, chain). The interfaces to these algorithms are non-blocking and non-global: the caller is not blocked until the message buffer could be re-used and the message is received like a normal active message as defined in INUKTITUT, without specific 'wait' instruction on the destination processors. These two features allow to have a better control on the overlapping of computation and communication when using collective communication operators within a computation. This is not true in related work (MPI) where these collective operators are blocking and global operations (all processors should make a call to the operator). Comparison with MPI on cluster with TCP network shows comparable performance for isolated calls and for short message size. The measured latency is better due to the lightweight implementation of INUKTITUT in comparison with MPI/CH or MPI/LAM.
ICS also contains distributed algorithms for the detection of the termination. These algorithms are based on the Chandy/Lamport's algorithm and are lock-free. There are used by ATHAPASCAN at two levels: to detect the termination of a set of threads inside a process and to detect the termination of a set of processes.

A functional prototype of INUKTITUT is available. This implementation is based on Posix Thread to provide multithread support. The active message model is developed on different levels of protocols and low level middleware : TCP, MPI and CORBA. The SCI and Myrinet support are also envisioned. Portability is not the single issue of multi-protocol support, the other one is to be able to use together different protocols and select the best to communicate between all couples of nodes. Moreover, heterogenity of data representation must be potentially addressed in coupling parallel applications. The present version of INUKTITUT has also an efficient scale launcher (Taktuk) and a library of collective communication services (ICS). The aim of INUKTITUT is to be the low-level part of ATHAPASCAN runtime and an implementation of MPI2 standard (Dyade-LIPS project). The complete library will be multi-networks compatible and thread-aware. It can be used on clusters and grids. To date, only static clusters can be exploited, in the near future, INUKTITUT should support dynamic clustering.

5.2. Tools for cluster management and software development

The large-sized clusters and grids show serious limitations in many basic system softwares. Indeed, the launching of a parallel application is a slow and significant operation in an heterogeneous configuration. The broadcast of data and executable files is widely under the control of users. The available tools do not scale because they are implemented in a sequential way. They are mainly based on a single sequence of commands applied over all the cluster nodes. In order to reach a high level of scalability, we propose a new design approach based on a parallel execution. We have implemented a parallelization technique based on spanning trees with a recursive starting of programs on nodes. Many industrial collaborations were carried out with Mandrake, BULL, HP and Microsoft.

KA-TOOLS: tools for cluster management The KA-TOOLS main idea is to extend the standard Unix tools for large-sized clusters. The KA-TOOLS include operating system deployment, launching of processes, file system operations, and system monitoring. Two software packages are available (under licence GPL) on the server http://ka-tools.sourceforge.net. *ka-run* is a library allowing the management of parallel processes and *ka-deploy* is a software package for the automatic installation of Linux and Windows 2000 systems over PC clusters. The KA-TOOLS are the basis of the Mandrake distribution for Linux cluster management and were developed within the RNTL CLIC

NFSP: parallel file system When deploying a cluster of PCs there is a lack of tool to give a global view of the available space on the drives. This leads to a suboptimal use of most of this space. To address this problem NFSP was developed, as an extension to NFS that divides file system handling in two components: one responsible for the data stored and the other for the metadata, like inodes, access permission.... They are handled by a server, fully NFS compliant, which will contact associated data servers to access information inside the files. This approach enables a full compatibility, for client side, with the standard in distributed file system, NFS, while permitting the use of the space available on the clusters nodes. Moreover efficient use of the bandwidth is done because several data servers can send data to the same client node, which is not possible with an usual NFS server. The prototype has now reached a mature state, the sources are available at http://www-id.imag.fr/Laboratoire/Membres/Lombard_Pierre/nfsp/.

5.3. Athapascan

The multiplicity of parallel architectures for high performance computing (symetric multi-processor, cluster, grid) requires to program using a parallel programming model with a semantic independent of the architecture and which allows efficient implementation. The key point is to control the memory usage to respect the dependencies between writers and readers of data and the scheduling should be able to ensure a good locality of data accesses. ATHAPASCAN is such an API and goes with a runtime library of scheduling algorithms allowing to port an application onto different parallel architectures (shared memory CC-NUMA or distributed memory architectures). The most famous related work is OpenMP that allows, by code annotations, to specify the scheduling strategies for the parallelism of (nested) loops. OpenMP only works on shared memory architectures where the hardware ensures the memory consistency.

Programming model In order to deal with data and control flow at a grain defined by the application (macro-data flow), parallelism is expressed through asynchronous remote procedure calls, called *tasks*, that communicate and are only synchronized via access to a shared memory. The ATHAPASCAN semantics relies on shared data access and ensures that the value returned by a read statement is the last written value (or a copy of) according to the lexicographic order defined by the program: statements are lexicographically ordered by ' ; '. The choice of such a sequential semantics is motivated by its direct readability on the program source. This also favors a direct reuse of existing sequential code. The parallelism is explicit (grain of task and grain of shared data) while the detection of the synchronizations is implicit, which makes ATHAPASCAN very easy to use.

Implementation of the scheduling Because the semantic is independent of the architecture, it is possible to choose the best suited scheduling strategy for a given application on a given architecture without any modification of the application. Code annotation allows to pass scheduling information to the ATHAPASCAN runtime system. The choice of the scheduling algorithm is made at runtime when launching the application. The current library of scheduling algorithms is split into three categories:

- sequential scheduling: it allows to execute any ATHAPASCAN program sequentially while respecting the semantics. The objective is to obtain an easy to debug execution.
- work-stealing: this scheduling algorithm is based on work stealing by idle processors. The graph construction and execution are lightly coupled. It works both on shared memory

multiprocessors architecture and distributed memory architecture (cluster or grid). An idle processor selects a victim processor using a strategy among a set of predefined and extendable strategies (random, hierarchical, round robin, ...).

– semi-static: at runtime the construction of the data flow graph part is scheduled as a whole using a certain strategy. Most of the available strategies are based on partitionning the graph according the 'Owner Compute Rule' heuristic used in HPF. The partitionning step is either based on the SCOTCH library developed in LABRI-Bordeaux or uses an orthogonal bisection algorithm using geometric information which comes from the application.

ATHAPASCAN uses INUKTITUT as a runtime. Current implementation is based on TCP and POSIX threads.
Several classes of applications have been used to test ATHAPASCAN: numerical subroutine in linear algebra (Cholesky factorisation of matrices, sparse matrix-vector product); parallelization of sequential code (gzip, PL a library for probabilistic inference calculus); simulation of quantic box, linear algebra in computer algebra, simulation of clothes. For each tested application the performances are good (for instance, they are better than the best know implementation for the Cholesky factorization) if the scheduling strategy is adapted: for symmetric multiprocessors, the performances are equivalent to Cilk (MIT) on serie-parallel classes of applications. On distributed memory architecture, the performances are close to the performances obtained using MPI or ScaLAPACK.

An overall presentation and documentation are available on the web server of the project http://www-apache.imag.fr.

5.4. NetJuggler : PC clusters for Virtual Reality

NETJUGGLER is a joint developpement effort with the LIFO, Université d'Orléans. NETJUGGLER enables the execution on a PC cluster of virtual reality applications developed with VR Juggler (www.vrjuggler.org). In a transparent way for the user, NETJUGGLER duplicates the application on the cluster nodes. It catches and broadcasts the input events to guarantee the coherency between the different copies. NETJUGGLER also integrates a swaplock synchronization to ensure that the images computed on the different nodes are displayed synchronously to form a large single coherent image.

The duplication strategy adopted by NETJUGGLER allows to gather the power of graphics cards distributed on a PC cluster. This approach is well adapted to a large majority of virtual reality applications that are graphically complex, but usually associated with simulations that do not require the power of multiple processors. NETJUGGLER was developed to be performant and scalable. It uses the MPI library and uses techniques of message aggregation and collective communication algorithms to reduce the network load. NETJUGGLER is open source and used in different companies and labs like the BRGM, France, the VRlab, Sweden, the HLR, California, Argonne Futures Lab, Illinois. Net Juggler is available at http://netjuggler.sourceforge.net.

5.5. OAR: simple and scalable batch scheduler for clusters and grids

OAR is a batch scheduler that emphasizes simplicity, extensibility, modularity, efficiency, robustness and scalability. It is based on a 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. Completly written in Perl, OAR is also extremely modular and straightforward to extend. Thus, it constitutes a privileged platform to develop and evaluate several scheduling algorithms and new kinds of services.

Most known batch schedulers (PBS, LSF, Condor, ...) are of old-fashioned conception, built monolithically, with the purpose of fulfilling most of the exploitation needs. This results in systems of high software complexity (150000 lines of code for OpenPBS), offering a growing number of functions that are, most of

the time, not used. In such a context, it becomes hard to control both the robustness and the scalability of the whole system.

The OAR project focuses on robust and highly scalable batch scheduling for clusters and grids. Its main objectives are the validation of grid administration tools such as TAKTUK, the development of new paradigms for grid scheduling and the experimentation of various scheduling algorithms and policies.

The grid development of OAR has already started with the integration of best effort jobs whose purpose is to take advantage of idle times of the resources. Managing such jobs requires a support of the whole system from the highest level (the scheduler has to know which tasks can be cancelled) down to the lowest level (the execution layer has to be able to cancel awkward jobs). The OAR architecture is perfectly suited to such developments thanks to its highly modular architecture.

The OAR system can also be viewed as a platform for the experimentation of new scheduling algorithms. Current developments focus on the integration of theoretical batch scheduling results into the system so that they can be validated experimentally. Future work will address the integration of new theoretical task models such as malleable tasks developed by the project.

6. New Results

6.1. Parallel algorithms, complexity and scheduling

Participants: P.-F. Dutot, L. Eyraud, T. Gautier, G. Huard, G. Mounié, R. Revire, J.-L. Roch, D. Trystram.

6.1.1. Scheduling

Past years, research on scheduling were focused on three main points. First, systematic studies were performed on various task scheduling heuristics on general graph, i.e. without any particular structural properties, conducting complexity analysis and looking for good performance ratio. Second, the studies on the impact of the execution model where performed on recent models like *BSP* or *LogP* in order to, partially, evaluate and compare them. This point leads to the conclusion that competitive scheduling algorithms are not realistically achievable in models where communication is explicit. This leads to the third point. A large part of ongoing work concerns promoting malleable tasks, introduced to simplify communications management in scheduling heuristics. Recent results show that very good approximation ratio are achieved in scheduling independent malleable tasks, with different minimization criterion, namely maximum completion time and average completion time. A trade-off between these two optimization criterion is proven to be efficient in this model. We also propose algorithms for scheduling graphs of malleable tasks with constant performance ratio. These results use a methodology where the allotment problem and the resulting mapping problem are solved one after the other. From a practical point of view, as the problem is difficult and does not have a fully polynomial approximation scheme, the complexity is moved to the graph building phase, where the user may introduce all its knowledge of the problem.

The major challenge is still how to extend traditional results on the new parallel platforms: clusters and lightweight grid (cluster of clusters), with heterogeneous components, hierarchical parallel execution, communication load imbalance, etc.

Another research area is the sensitivity of scheduling algorithms, i.e. the capacity to take into account disturbance on data (computation and communication), and if necessary, stabilize these algorithms with a control of the execution. An approach to solve the question is to mix static scheduling policy and local on-line scheduling. Some classical algorithms in ATHAPASCAN environment were analyzed from this point of view.

6.2. Inuktitut

Participants: J. Briat, T. Gautier, C. Martin, M. Pillon, O. Richard.

6.2.1. Taktuk: parallel launcher

TAKTUK is a tool to launch or deploy efficiently parallel applications on large clusters, and simple grids. Efficiency is obtained thanks to the overlay of all independent steps of the deployment. We have shown that

this problem is equivalent to the well known problem of the single message broadcast. Performance gap between the cost of a network communication and of a remote execution call enables us to use a work stealing algorithm to realize a near-optimal schedule of remote execution calls.

6.3. Administration and exploitation tools for cluster

Participants: P. Augerat, J. Briat, W. Billot, Y. Denneulin, G. Huard, A. Lebre, P. Lombard, C. Martin, S. Martin, B. Richard, O. Valentin.

A cluster is a parallel machine with a high number of PCs and an operating system on each PC. Our research aims at providing tools to help the installation, and in such architecture resources and nodes can appear and disappear due to node failure or network connectivity problems, at managing dynamic resources and finally, middleware to handle huge distributed space for data storage.

6.3.1. Batch scheduler for clusters and grids

A first prototype of OAR is written in a very high level language (Perl) and makes intensive use of high level tools (MySql and TAKTUK a project software), thereby resulting in a concise code (around 5000 lines of code) easy to maintain and extend. This small code as well as the choice of widespread tools (MySql) are essential elements that ensure a strong robustness of the system. Secondly, OAR makes use of Sql requests to perform most of its job management tasks thereby getting advantage of the strong scalability of most database management tools. Such scalability is further improved in OAR by making use of Taktuk to manage the nodes themselves. Current development in OAR focuses on its extension to Grids and advanced scheduling techniques. The extension of OAR to Grids has already started by making it support best effort jobs. OAR is currently deployed on a cluster of 48 machines located into the project, and on a grid architecture between several labs at Grenoble.

The integration of advanced scheduling techniques is in progress and aims at adding both state of the art batch scheduling algorithms and new task models.

6.3.2. Storage and transfer solutions for clusters and grids

The NFSP software gives an unique view of a set of disks on nodes. It addresses two topics: giving an unified interface to a set of storage resources and enabling parallel read operations on a file for better scaling. It is suited to a cluster environment but not really to a grid one. The metadata server can only be used in a local environment, NFS uses the UDP protocol which is almost never used across sites for security reasons, so it is not possible to access the server remotely, like any other NFS servers. If this ware possible, it would be inefficient because NFS assumes a low communication latency and its protocol is optimized for that. On the other hand the distributed storage capability of NFSP can be used across clusters to do simultaneous transfers between sets of storage nodes and, thus, increasing total bandwidth. The Gxfer library was written to validate this idea, and we were able to transfer a 1 gigabyte file in 9 seconds between two clusters of 10 nodes connected through a 10 gigabit link, the network of each cluster being a Fast Ethernet.

Using the NFSP file system and the Gxfer tool we are designing a distributed file system which will keep NFS characteristic but with distributed naming and efficient file transfer. This work was initiated in the master project of Olivier Valentin.

6.4. Athapascan

Participants: T. Gautier, S. Jafar, H. Hamid Reza, R. Revire, J.-L. Roch.

6.4.1. Control of the overhead of execution with work stealing

In order to reach fine grain computation, ATHAPASCAN extends the work-first principle of Cilk to the data flow graph: because most of the tasks of a program are executed sequentially, the construction of the graph should require as few instructions as possible. In ATHAPASCAN runtime, it requires as few instructions as to push a task into a stack and update some pointers. The state of tasks (ready or not ready) is never computed during such a sequential execution. This is possible because the semantics of ATHAPASCAN allows a schedule

which is closed to a pure sequential order of execution. All of the overhead due to the computation of ready-to-execute tasks and detection of synchronisation (communication) is required only by the steal operation when a (virtual) processor becomes idle. The number of such steal operations is correlated to the parallel time (on an infinite number of processors) of the program, which is very small in comparison to the total number of operations if the parallel program can exhibit enough parallelism.

Moreover, the graph can be built in parallel without global synchronisation (to ensure consistency or due to heap allocation where implicit synchronisation is generally implemented by the memory allocator of standard C++ library).

The practical results are: a reduction by a factor of about 1000 in the time of creating a task compared to the previous implementation of ATHAPASCAN. The same factor remains in the comparison the total execution time of the fine grain program when compared with the previous implantation; A 98% of efficiency in the usage of processors on various architectures (measured up to 64 processors on an CC-NUMA Origin38000 and up to 100 processors of a cluster) is obtained on fine grain applications and shows that the runtime kernel for multithreaded and distributed computing is scalable.

6.4.2. Semi-static scheduling strategies

Performance of most of the parallel numerical simulations, such as domain decomposition, rely on computing a good partitioning of the data taking into account the dependencies from the application. A dependency graph models the execution and a graph partitioner allows to compute the mapping of data on the processors. In an ATHAPASCAN application, a data flow graph models the execution: the graph is bipartite (data and tasks are distinct), each access to a data made by a task is known. For such numerical simulations, we have extended the scheduling algorithms library of ATHAPASCAN with a new strategy based on the partitioning of a graph of data dependencies embedded in the data flow graph. We have used SCOTCH library (Labri, Bordeaux) as a graph partitioner. Then, using heuristics such as 'Owner Compute Rule' (used in HPF), we map each task on the processor that holds most of its effective parameters.

To take into account spatial neighbourhood properties of distributed data, we have also developed a well known orthogonal bisection algorithm to partition the data graph of dependency according to their spatial positions. This strategy has been used with success in the specific context of molecular dynamics several years ago.

Both these strategies ensure that "most" of the neighboring data will be mapped onto the same processors. The owner compute rule heuristic ensures that most of the tasks that shared data will be mapped onto the same processors. The communication volume is directly correlated to the quality of the computed partition. First results of these strategies show a good speed up of a small cluster (16 bi-processors) in the application of the simulation of clothes.

6.4.3. Scheduling strategies for multiple invocations in CORBA

Code coupling requires the management of all the parallelism between applications: the parallel execution and the parallel communication should be expressed and then exploited. This is a complex work that should be partially hidden to the end users. In the context of CORBA, we have developed a methodology that allows to automatically exploit all the parallelism between invocations of methods and the communication to transfer the parameters needed to realize each invocation on the objects. The idea is to reschedule the invocations by splitting each blocking invocation in two non blocking invocations. ATHAPASCAN is used to detect the synchronisations and exploit the parallelism. A prototype, called HOMA, is operational: from the description of the IDL interface, our compiler automatically generates an ATHAPASCAN program which respects the original semantic but does the rescheduling. The end user should only recompile its applications with the HOMA IDL compiler. First results have shown that HOMA is able to exploit all the parallelism on a set of independent invocations which model a coupled application in chemistry developed during our past collaboration in the INRIA ARC SIMBIO with NUMATH project, LCTN (Nancy) and CEA-DSV (Grenoble).

6.5. Tools for performance evaluation

Participants: A. Benoit, C. Guilloud, G. Mounié, B. Plateau, I. Sbeity, E. Sulaiman, J.-M. Vincent.

6.5.1. Modelling and performance

The results obtained this year concern parallel simulation methods to certify the reachability of the steady state of large systems. These methods come from statistical mechanics and have been implemented and tested.

Other results have been obtained in the area of numerical solutions for Markov chains. We have proved that when a stochastic automata network presents replications, the chain is lumpable and that the lumped chain also has a generator which can be obtained by a sum of tensor product. This new formulation can be computed automatically. We have proposed a new semantics for stochastic automata with discrete time scale.

All these results are implemented in a software package called PEPS.

6.5.2. Generic trace and visualization

Paje allows applications programmers to define what is visualized and how new objects should be drawn. To achieve such flexibility, the hierarchy of events and the visualization commands may be defined by the programmers inside the applications. The visualization of parallel execution of ATHAPASCAN applications was achieved without any new addition into Paje software. Inserting few events trace into the ATHAPASCAN runtime allows the visualization of different facets of the program: application computation time but also user task graph management and scheduling of these tasks. Paje is also, among others, used to visualize Java program execution and large cluster monitoring. The PhD thesis of C. Guilloud aims at designing and evaluating flexible tools at the trace generation level.

6.6. Applications

Participants: J. Allard, T. Gautier, C. Ménier, G. Parmentier, E. Romagnoli, D. Trystram, J.-L. Roch, J.-M. Vincent, F. Zara.

6.6.1. Dynamic maps on demand

Geographic researchers create and manipulate maps with various data, changing data type, data correlation, smoothing function, etc. Building a map with acceptable resolution requires around one hour on a standard PC. Interactivity is thus impossible. By parallelizing the application with MPI and then ATHAPASCAN, optimizing the code and selecting suitable libraries, this time was reduced to a few seconds on a parallel cluster. A first prototype was build. One of the scientific problems with this architecture is to take into account previous map generation, in the map request flow. This requires a caching policy which is one of the main speedup factor of the application.

6.6.2. Probabilistic inference calculus

The APACHE project has developed the parallel part of a probabilistic inference machine library provided by Pixellis and SHARP project. Currently, this software is commercialized by Probayes (an INRIA startup). The problem is to evaluate a tree of expression. Both parallelisation for symmetric multiprocessors and clusters has been developed using ATHAPASCAN. Several parts of the library have been parallelized, the target architecture is a 2-processors machine. A cluster version is under development. The key point in the parallelisation was to reduce the overhead in task creation. We use a method based on an adaptive algorithm that extracts at runtime the parallelism of the evaluation when processors become idle. Good speedups have been reached for coarse and medium grain instances (between 70% and 99% of efficiency on a 2-processors machine).

6.6.3. Virtual Reality

The duplication strategy implemented with NETJUGGLER fits a large range of virtual reality applications. However, some applications, that will probably become more and more common in a near future, require high performance computations that fails to supply the duplication strategy. For these applications, we developed coupling strategies, using NETJUGGLER, to enable the distribution of the computations that lead to performance bottlenecks. We developed two different such applications, the former using a parallel fluid flow simulation solver programmed with the PETSC library and the latter a cloth simulation parallelized with ATHAPASCAN. The resulting application associates two different levels of parallelism. On one side,

a parallel rendering for display wall and on the other side the parallel simulation. Up to now the coupling is synchronous: each new simulator step corresponds to a new image. But scaling to a large number of processors and introducing different interaction modes (haptics, image, sound), requires a more efficient management of the available resources through a more flexible coupling (loosen coherency coupling). On-going work focuses on a extension of Net Juggler to enable asynchronous module coupling with automatic flow regulation between modules.

7. Contracts and Grants with Industry

7.1. Collaboration INRIA-HP labs, 00-03

- The collaboration with HP-Labs at Grenoble was concerned with the use of idle resources inside an Intranet. HP donated to the INRIA APACHE project a cluster with 225 processors PC and human effort. 3 PhD students were funded.
- Gelato Consortium: INRIA joined the Gelato Federation in February 2003 and Apache has a leadership position for this project within INRIA. Co-founded by HP and seven of the world leading research institutions, Gelato is an open source community initiative designed to foster the development and dissemination of focused computing solutions for researchers and associated IT staffs working on Linux-based Intel¨ Itanium¨ 2 platforms.

7.2. Collaboration INRIA-BULL : action Dyade LIPS, 00-03, 03-06

In the context of a global partnership between BULL and INRIA, BULL and the Apache project collaborate to develop clustering software solutions aimed at very large computing infrastructures. These clusters feature a complete software environment including management tools, efficient storage solutions and resource management. The partnership promotes the cluster architectures based on the Intel Itanium 2 processor which has established new records for floating point processing. This processor provides the 64-bit wide addressing scheme needed by large data sets of scientific applications and has up 6 MB of on-chip cache to give the processor superfast access to data. BULL has developed FAME (Flexible Architecture for Multiple Environment) by using standard component assemblies as the building block of larger systems.

Cyrille Martin was funded by BULL during 3 years to prepare his PhD. He studied the deployment of parallel applications on large clusters, that can be extended to grids. The deployment includes on one hand the launching of the parallel program on all nodes and on the other hand the setting up of a communication layer. The good properties and performance figures of this tool, TAKTUK, are demonstrated by its use in several projects like: KA-TOOLS (included and used by the Clic Mandrake Cluster Linux distribution), OAR (Job manager) and INUKTITUT (Communication layer of the environment ATHAPASCAN). The thesis will be defended in December 2003.

Cyril Guilloud was also funded by BULL during 3 years. The Paje software is used to provide behavioral visualizations of parallel programs. The work on trace recording and trace manipulation facilities can be generated from a description of the events to be traced. and the trace-driven simulator reconstructing the behavior of the observed execution. The Paje simulator is generic and can be specialized by the objects to be visualized. The thesis will be defended in February 2004.

Adrien Lebre is also funded by a BULL grant since April 2003. The scientific area he will work on consists in a study of Input/Output characteristics from HPC Applications and existing Parallel I/O Solutions. Currently, Adrien Lebre is studying the different behaviors of such systems, with a particular interest in the parameters ruling these from a hardware, middleware and application point of view. The next steps will tackle the issues related to MPI/IO.

In January 2004, two new students (Estelle Gabarron and Maxime Martinasso) will start PhD with BULL grants.

7.3. RNTL project CLIC, 02-04

The APACHE project collaborates with MandrakeSoft and Bull to build a Linux distribution for cluster. APACHE contributes to the tools for exploitation (deployment, parallel commands, parallel file system) as well as with the parallel programming environment ATHAPASCAN. This project is providing funding for 48 months of expert engineer as well as equipment.

7.4. RNTL project E-Toile, 02-04

The APACHE project is, among other labs and the CS, Sun, EDF and CEA companies, part of the RNTL ETOILE project whose goal is to a build a production grid testbed based on clusters and to use it on significant applications. The APACHE project has two years of funding for an engineer and also funds to buy hardware and to travel. The GXfer product, to transfer efficiently files across a grid, was developed for this project.

7.5. RNRT project SIDRAH, 02-04

The RNTL project SIDRAH associates the APACHE project, HP and France Telecom. The goal is to study a research infrastructure for ubiquitous computing where communicating objects have to share information. The development will be based on an extension of existing software. The funding provides equipment and travelling and 15 months of engineer.

7.6. RNTL project GEOBENCH, 03-04

The RNTL project GEOBENCH associates the APACHE project, the INRIA action i3D, the LIFO of Université d'Orléans, the CEA, the BRGM and the TGS company. The goal is to develop solutions running on PC clusters for the visualization of (geo)scientific data. Data distribution and computations are supported by Net Juggler. The Amira software from TGS will provide a visualization oriented library for scientific data processing (iso-surfaces extraction for instance). Visualization, more specifically targeting the workbench virtual reality environment (2 L-shaped visualization surfaces), will be associated with haptics interaction. Two classes of applications are considered: applications handling large data sets (CEA application) and applications based on geo-referenced data (BRGM application).

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

7.7. RNTL project OCETRE, 04-05

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

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

7.8. RNTL project IGGI, 04-05

The RNTL project IGGI associates the APACHE project, the companies BRGM and Mandrake. The goal is to develop middleware for Intranet clustering as an extension to the CLIC RNTL project. The funding is stil under discussion.

7.9. CIFRE with IFP, 03-06

A collaboration with the company IFP (Institut Français du Pétrole) and APACHE 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 ATHAPASCAN.

7.10. CIFRE with ST Microelectronics, 03-06

A PhD thesis involving APACHE 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 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 goal of this thesis is to perform a deep work on the improvement of exact methods as well as to propose near-optimal approximations of the problem when exact methods cannot be used anymore.

7.11. INRIA-Pixelis, 03-03

A collaboration with the company Pixelis and INRIA/APACHE is providing funds (8K euros) in order to parallelize with ATHAPASCAN an application in the area of bayesian calculus.

8. Other Grants and Activities

8.1. Regional initiatives

- Grappe200 project: MENRT -UJF-INPG-(800KF), Region Rhône-Alpes (1.2MF), INRIA (2.5MF), ENS-Lyon(300KF) have funded a 4.8 MF cluster composed by 110 bi-processors Itanium2 connected with a Myrinet (donation of MyriCom) high performance network. This project is lead by APACHE, ReMaP and SARDES. It is part of the project CIMENT which aims at building high performance distributed grid between several research labs.
- RagTime project, 02-03: This project gathers numerous academic partners in the Rhône-Alpes region, medical centers and hospitals on the subject of medical grids an medical image analysis. Apache is involved of security aspects.

8.2. National initiatives

- Sure Path, 03-04, ACI SECURITY: Partners (INRIA-Apache, IRISA-Armor, PRISM-Epri). In the area of distributed systems and networking, the objective of the project is to apply an expertise in mathematical tools, techniques, algorithms and software packages for performance, reliability or dependability studies.
- *AGRIBES*, 02-03, ACI GRID: The goal of this project is to create collaborations between research teams working in scientific computing, databases and parallelism on the important topic of large data management over peer to peer networks. Others partners include LSR (Grenoble), CEA (Grenoble), LIP6 (Paris), PRiSM (Versailles), IN2P3 (Lyon), CESR (Toulouse), PARIS (Rennes), LIP (Paris), and the HP labs (Grenoble).
- *GRID2*, 02-03, ACI GRID: animation project 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).
- *DOC-G*, 02-03, ACI GRID: The project aims at exploiting a grid architecture to solve challenging problems in combinatorial optimization. Partners are PRISM (Versailles), LIFL (Lille).
- *CGP2P* 02-03, ACI GRID: The project is related to the deployment of a global computing system and a peer to peer system. Partners are: ASCI (Orsay), LAL (Orsay), LaRIA (Amiens), LIFL (Lille) et LRI (Orsay).

- *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 : the projets MOVI, APACHE and ARTIS (INRIA Rhône-Alpes) and the LIRIS laborator (Lyon).
- In 2003, the APACHE project is participating in the following AS-CNRS:
  - Random models and performance evaluation of distributed systems. Leader: Laurent Truffet.
  - Programming model for grid computing. Leader: Raymond Namyst.
  - Study the infrastructure for a national research grid (Grid 5000). Leader: Franck Cappello.

8.3. International initiatives

8.3.1. Foreign office action (MAE and MENESR):

- **Europe:**

  CoreGrid: The project APACHE participates in the proposal of a Network Of Excellence CoreGrid.

  Polonium : with the university of Gdansk, Poland, about the analysis of ATHAPASCAN (performances and semantics).

  Polonium : with the university of Poznan, Poland, about the parallelisation of the algorithm for multiple alignment of DNA sequences.

  Procope : with the university of Kiel, Germany, about proven approximation algorithms for scheduling problems.

- **Africa:**

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

  Tunisia : with the engineering department of science of Tunis about algorithmics and parallel programming (Project CMCU).

8.3.2. North America

- Mexico : Collaboration with "Univeritad Autonoma Metropolitana Mexico" (UAM) with Pr. E. Perez-Cortes on tracing and monitoring component based distributed applications. Participation to the LAFMI and joint project with LANIA (Pr. V. German-Sanchez) on infrastructures of middleware. Collaboration with Pr. A. Tchernykh at CICESE (Ensenada) on performance evaluation of clusters.
- NSF Project with W. Stewart (NC State University), G. Ciardo (College William and Mary), S. Donatelli (U. de Turin), 2002-2006. The purpose of the project is to study structured methods for Markov chains in order to evaluate the performances of distributed systems.

8.3.3. South America

- CNPq-INRIA PAGE II project with the universities of Rio Grande do Sul, Brazil (UFRGS, UFSM, PUC, UNISINOS), around PC cluster and performance evaluation tools. Tow years funding (juin 2001- juin 2003).
- USP-COFECUB project with the universities of Sao Paulo and Fortaleza, Brazil, focused on the impact of communications on parallel task scheduling. One year funding.

8.4. Visiting scientists

- Philippe Navaux, UFRGS, Porto Alegre, Brésil, 2003, 10 days, january.
- William Stewart, université, North Carolina University, Raleigh, USA, 2003, 3 weeks, june.
- Andreï Tchernyk, CICESE, Ensenada, Mexique, 2003, 1 month, march.
- Paulo Fernandez, PUC Porto-Alegre, Brazil, 1 month, 2003, january.
- Alfredo Goldman, SãoPaulo, Brazil, 10 days, 2003, april.
- Tudrug, Varosovy, Poland, 10 days, 2003, september.

8.5. Cluster computing center

8.5.1. The ICluster1 and ICluster2 Platforms

The APACHE project manages a cluster computing center on the Grenoble campus. The center manages different architectures: a 225 processors PC cluster (ICluster-1), a 48 bi-processors PC (ID-POT), and, since september, the center is involved with a new cluster based on 110 bi-processors Itanium2 (ICluster-2) located at INRIA.

More than 60 research projects in France have used the architectures, especially the 225 processors Icluster-1. Half of them have run typical numerical applications on this machine, the remainder has worked on middleware and new technology for cluster and grid computing.

8.5.2. The GrImage Platform.

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

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

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

The first part of GrImage (75 Keuros) was funded in 2003 by the INRIA and the Ministère de la Recherche (via l'INPG). We expect a second funding in 2004 (165 Keuros) from the INRIA and the Ministère de la Recherche. GrImage will eventually include 16 projectors, 20 numerical cameras and 28 PCs.

9. Dissemination

9.1. Leadership within scientific community

- Program commitees :
  VECPAR 2004, Performance and Tools 2003, SBAC-PAD 2003, RENPAR 2003, NSMC 2003, IPDPS 2003, OPODIS 2003.
- 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.

9.2. Startup creation: ICATIS

P. Augerat has prepared in 2003 the launching of a start-up ICATIS in january 2004. ICATIS has obtained funds from Rhône-Alpes Region and CNRS. ICATIS will sell a software to install and manage large dynamic clusters. Y. Denneulin and B. Raffin will do expertise work within this company.

10. Bibliography

Major publications by the team in recent years

[1] K. ATIF, B. PLATEAU. *Stochastic Automata Network for modeling parallel systems.* in « IEEE Transactions on Software Engineering », number 10, volume 17, October, 1991.

[2] E. BAMPIS, J.-C. KONIG, D. TRYSTRAM. *Minimizing the Schedule Length for a parallel 3D-precedence Graph.* in « European Journal of Operational Research », number 95, 1996, pages 427-438.

[3] R. D. BLUMOFE, C. E. LEISERSON. *Space-efficient scheduling of multithreaded computations.* in « SIAM Journal on Computing », number 1, volume 27, 1998, pages 202-229.

[4] *Réseaux à haut débit de stations pour le support d'applications parallèles et réparties.* B. FOLLIOT, B. TOURANCHEAU, editors, series Calculateurs Parallèles Réseaux et Systèmes répartis, number 1, volume 10, HERMES, February, 1998.

[5] M. FRIGO, C. E. LEISERSON, K. H. RANDALL. *The Implementation of the Cilk-5 Multithreaded Language.* in « Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDÍ98) », June, 1998.

[6] T. GAUTIER, J.-L. ROCH, G. VILLARD. *Regular versus irregular problems and algorithms.* in « Proc. of IRREGULAR'95 », series LNCS, volume 980, A. FERREIRA, J. ROLIM, editors, Lyon, France, 1995.

[7] E. KRAEMER, J. T. STASKO. *The Visualization of Parallel Systems: An Overview.* in « Journal of Parallel and Distributed Computing », number 2, volume 18, June, 1993, pages 105-117.

[8] T. LEBLANC, J. MELLOR-CRUMMEY. *Debugging Parallel Programs with Instant Replay.* in « IEEE Transactions on Computers », number 4, volume C-36, April, 1987, pages 471-481.

[9] M. C. RINARD. *The design, implementation and evaluation of Jade : a portable, implicitly parallel programming language.* in « ACM Transactions on Programming Languages and Systems », number 3, volume 20, May, 1998, pages 483-545.

[10] J.-M. VINCENT. *Some Ergodic Results on Stochastic Iterative Discrete Event Systems.* in « Discrete Event Dynamic Systems », number 2, volume 7, 1997, pages 209-232.

[11] T. YANG, C. FU. *Space/Time-Efficient Scheduling and Execution of Parallel Irregular Computations.* in « ACM Transactions on Programming Languages and Systems », number 4, volume 21, 1999.

Doctoral dissertations and "Habilitation" theses

[12] A. BENOIT. *Méthodes et algorithmes pour l'évaluation des performances des systémes informatiques à grand espace d'états.*. Thèse de doctorat, Institut National Polytechnique de Grenoble, June, 2003.

[13] P. LOMBARD. *NFSP: Une solution de stockage distribue pour architectures grande echelle.* Thèse de doctorat, Institut National Polytechnique de Grenoble, December, 2003.

[14] C. MARTIN. *Déploiement et contrôle d'applications parallèles sur grappes de grande taille.* Thèse de doctorat, Institut National Polytechnique de Grenoble, December, 2003.

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

[16] B. RICHARD. *I-Cluster : Agrégation des ressources inexploitées d'un intranet et exploitation pour l'instanciation de services de calcul intensif.* Thèse de doctorat, Institut National Polytechnique de Grenoble, December, 2003.

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

[18] 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, December, 2003.

Articles in referred journals and book chapters

[19] A. DARTE, G. HUARD. *New Complexity Results on Array Contraction and Related Problems.* in « Journal on VLSI Signal Processing », 2003.

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

[21] T. GAUTIER, H. HONG, J.-L. ROCH, W. SCHREINER. V. W. J. GRABMEIER, editor, *Handbook of Computer Algebra – Foundations, Applications, Systems.* Springer Verlag, Heidelberg, 2002, chapter Parallel implementation.

[22] A. GOLDMAN, G. MOUNIE, D. TRYSTRAM. *1-Optimality of static BSP computations: scheduling independent chains as a case study.* in « Theoretical Computer Science », number 290, 2003, pages 1331-1359.

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

[24] A. GUPTA, G. PARMENTIER, D. TRYSTRAM. *Scheduling precedence task graphs with disturbances.* in « RAIRO Operational Research », 2003, to appear.

[25] R. LEPERE, G. MOUNIE, D. TRYSTRAM. *An Approximation Algorithm for Scheduling Trees of Malleable Tasks.* in « European Journal of Operational Research », number 142, 2002, pages 242-249.

[26] R. LEPERE, D. TRYSTRAM, G. WOEGINGER. *Approximation Scheduling For Malleable Tasks under Precedence constraints.* in « International Journal of Foundation in Computer Science », number 4, volume 13, 2002, pages 613-627.

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

Publications in Conferences and Workshops

[28] 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.

[29] 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.

[30] 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.

[31] A. BENOIT, L. BRENNER, P. FERNANDES, B. PLATEAU. *Agregation of Stochastic Automata with replicas.* in « International Conference on the Numerical Solution of Markov Chains », Urbana, Illinois, USA, September, 2003.

[32] A. BENOIT, L. BRENNER, P. FERNANDES, B. PLATEAU, W. STEWART. *The PEPS Software tool.* in « 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation », Springer, Urbana, Illinois, USA, September, 2003.

[33] A. BENOIT, B. PLATEAU, W. STEWART. *Memory-efficient Kronecker algorithms with applications to the modelling of parallel systems.* in « International Parallel and Distributed Processing Symposium, Workshop on Performance Modeling, Evaluation, and Optimization of Parallel and Distributed Systems », Nice, Avril, 2003.

[34] J. BLAZEWICZ, M. KOVALYOV, M. MACHOWIAK, D. TRYSTRAM, J. WEGLARZ. *Exact algorithms for scheduling Malleable Tasks.* in « EURO - INFORMS », Istanbul, Turkey, july, 2003.

[35] N. CAPIT, G. D. COSTA, G. HUARD, C. MARTIN, G. MOUNIÉ, P. NEYRON, O. RICHARD. *Expériences autour d'une nouvelle approche de conception d'un gestionnaire de travaux pour grappe.* in « Actes de CFSE 2003 », 2003.

[36] J. CHASSIN DE KERGOMMEAUX, C. GUILLOUD, B. DE OLIVEIRA STEIN. *Flexible performance debugging of parallel and distributed applications.* in « Proc. of Euro-Par 2003 », series LNCS, volume 2790, Springer Verlag, H. KOSCH, L. BSZRMNYI, H. HELLWAGNER, editors, pages 38-46, August, 2003.

[37] 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, August, 2002.

[38] 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, July, 2002.

[39] Y. DURAND, S. PERRET, J.-M. VINCENT, C. MARCHAND, F.-G. OTTOGALLI, V. OLIVE, S. MARTIN, B. DUMANT, S. CHAMBON. *SIDRAH: A software infrastructure for a resilient community of wireless devices.* in « Smart Objects Conference », pages 134-137, 2003.

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

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

[42] P.-F. DUTOT, G. MOUNIE, D. TRYSTRAM. *Scheduling Parallel Tasks: Approximation algorithms.* in « Handbook on Scheduling algorithms: Algorithms, Models and Performance Analysis », C. P. JOSEPH LEUNG ED., editor, to appear april 2004.

[43] 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, pages 308-315, 2003.

[44] 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) », pages 127–134, La Colle sur Loup, France, 2003.

[45] P. LOMBARD, Y. DENNEULIN, O. VALENTIN, A. LEBRE. *Improving the Performances of a Distributed NFS Implementation.* in « o appear in the Proceedings of the Fifth International Conference on Parallel Processing and Applied Mathematics (PPAM 2003), year = 2003, month = sep, publisher = Springer-Verlag, series = Lecture Notes in Computer Science ».

[46] A. MAHJOUB, C. RAPINE, D. TRYSTRAM. *Influence of starting solutions on the stabilization of scheduling algorithms.* in « EURO - INFORMS », Istanbul, Turkey, july, 2003.

[47] C. MARCHAND, G. DA COSTA. *Traces et profils utilisateurs dans les systèmes Pair à Pair, application à l'ADSL.* in « Atelier d'Evaluation de Performances 2003 », Reims, France, 2003.

[48] C. MARCHAND, G. DA COSTA. *Eléments de caractérisation des environnements des systèmes Pair à Pair.* in « Proceedings des Rencontres Francophones du Parallélisme (RenPar'15) », INRIA, editor, pages 161-168, La Colle sur Loup, France, October, 2003.

[49] C. MARCHAND, J.-M. VINCENT. *Détecteurs de défaillances et qualité de service dans un réseau ad-hoc hétérogène.* in « CFSE'3 », INRIA, editor, pages 525-536, La Colle sur Loup, France, October, 2003.

[50] B. PLATEAU. *The Grid : Challenges and Research issues.* in « Proceedings of the 13th International Conference on Domain Decomposition Methods », LNCS 2550, Hanoi, Vietnam, december, 2002, http://link.springer.de/link/service/series/0558/tocs/t2550.htm.

[51] R. REVIRE, F. ZARA, T. GAUTIER. *Efficient and Easy Parallel Implementation of Large Numerical Simulation.* in « Proceedings of ParSim03 of EuroPVM/MPI03 », SPRINGER, editor, pages 663–666, Venice, Italy, 2003.

[52] B. RICHARD, D. CHALON, D. M. NIOCLAIS. *Clique: A transparent, peer-to-peer replicated file system.* in « Proceedings of the 4th International Conference on Mobile Data Management, Melbourne, Australia, January, 2003 », 2003.

[53] 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.

[54] 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, editors, pages 169–176, La-Colle-Sur-Loup, France, 15–17 Octobre, 2003.

[55] J.-M. VINCENT, C. MARCHAND. *On the exact simulation of functionals of stationary Markov chains.* in « Fourth International Conference on the Numerical Solution of Markov Chains (NSMC'03) », pages 77-97, Urbana, Illinois, USA, September, 2003.

[56] 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, editors, pages 105–112, Blaubeuren, Germany, September, 2002.

[57] 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.

Internal Reports

[58] T. GAUTIER, R. REVIRE, J.-L. ROCH. *Athapascan: API for Asynchronous Parallel Programming.* Technical report, number RT-0276, APACHE, INRIA Rhône-Alpes, February, 2003, http://www.inria.fr/rrrt/rt-0276.html.