INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team PARIS

Programming Parallel and Distributed Systems for Large Scale Numerical Simulation Applications

Rennes - Bretagne Atlantique

THEME NUM

Table of contents

1. Team

The PARIS *Project-Team was created at* IRISA *in December 1999. In November 2001, it has been established as a joint project-team (projet commun) between* IRISA *and the Brittany Campus of* ENS CACHAN. *Since, the project activity is jointly supervised by a ad-hoc Committee on an annual basis.*

Head of project-team

Thierry Priol [Research Director (DR) INRIA, HdR]

Administrative assistants

Maryse Auffray [Secretary (TR) INRIA]
Sandrine L'Hermitte [XTREEMOS Scientific Coordinator Assistant INRIA]
Päivi Palosaari [CoreGRID Scientific Coordinator Assistant INRIA, till May 2007]
Olivia Vasselin [CoreGRID Scientific Coordinator Assistant INRIA, from June 2007]

Staff members Inria

Gabriel Antoniu [Research Associate (CR)]
Yvon Jégou [Research Associate (CR)]
David Margery [Research Engineer (IR, part-time 50%)]
Christine Morin [Research Director (DR), HdR]
Christian Pérez [Research Associate (CR), HdR]

Staff members University Rennes 1

Françoise André [Professor, HdR]
Jean-Pierre Banâtre [Professor, HdR]
Pascal Morillon [Engineer (IE)]
Yann Radenac [Temporary Teaching and Research Associate (ATER)[1], till September 2007]

Staff members Insa de Rennes

Jean-Louis Pazat [Professor, HdR]
Jérémy Buisson [Temporary Teaching and Research Associate (ATER)[2], till September 2007]

Staff member Ens Cachan

Luc Bougé [Professor, ENS CACHAN Brittany Campus, HdR]

Project technical staff (temporary positions)

Landry Breuil [INRIA Associate Engineer (IA), ANR MD Project LEGO]
Matthieu Fertré [INRIA Associate Engineer (IA), XTREEMOS IP Project, since March 2007]
Mathieu Kermarrec [INRIA Associate Engineer (IA), ANR CI Project NUMASIS]
Pascal Le Métayer [INRIA Associate Engineer (IA), XTREEMOS IP Project, since July 2007]
Raúl López Lozano [INRIA Associate Engineer (IA), ANR CI Project DISC]
Jean Parpaillon [INRIA Associate Engineer (IA), till September 2007]
Oscar Sanchez [INRIA, XTREEMOS IP Project]

PhD students

Julien Bigot [MENRT Grant, since October 2007]
Hinde Lilia Bouziane [INRIA Grant]
Loïc Cudennec [INRIA and Brittany Regional Council Grant]
Boris Daix [CIFRE EDF Industrial Grant]
Nagib Abi Fadel [SARIMA Grant]
Jérôme Gallard [INRIA Grant, since October 2007]
Emmanuel Jeanvoine [CIFRE EDF industrial Grant[3], till September 2007]
Sylvain Jeuland [INRIA Grant, since October 2007]
Bogdan Nicolae [MENRT Grant, since October 2007]

[1] PhD defended on April 18, 2007.
[2] PhD defended on September 25, 2006.
[3] PhD defended on November 27, 2007.

Alexandru Popovici [INRIA Grant, since November 2007]
Thomas Ropars [MENRT Grant]
Mohamed Zouari [INRIA and Brittany Regional Council Grant, since October 2007]

Post-doctoral fellows

Adrien Lèbre [INRIA Post-Doc, XTREEMOS IP Project]
Xuanhua Shi [INRIA Post-Doc, till September 2007]
Manuel Caeiro Rodríguez [COREGRID Post-Doc, from December 2007]

2. Overall Objectives

2.1. General objectives

The PARIS Project-Team aims at contributing to the programming of parallel and distributed systems for large-scale numerical simulation applications. Its goal is to design operating systems and middleware to ease the use of such computing infrastructure for the targeted applications. Such applications enable the speed-up of the design of complex manufactured products, such as cars or aircrafts, thanks to numerical simulation techniques.

As computer performance rapidly increases, it is possible to foresee in the near future comprehensive simulations of these designs that encompass multi-disciplinary aspects (structural mechanics, computational fluid dynamics, electromagnetism, noise analysis, etc.). Numerical simulations of these different aspects will not be carried out by a single computer due to the lack of computing and memory resources. Instead, several clusters of inexpensive PCs, and probably federations of clusters (aka. *Grids*), will have to be simultaneously used to keep simulation times within reasonable bounds. Moreover, simulation will have to be performed by different research teams, each of them contributing its own simulation code. These teams may all belong to a single company, or to different companies possessing appropriate skills and computing resources, thus adding geographical constraints. By their very nature, such applications will require the use of a computing infrastructure that is *both* parallel and distributed.

The PARIS Project-Team is engaged in research along five topics: *Operating System and Runtime for Clusters and Grids*, *Middleware Systems for Computational Grids*, *Large-Scale Data Management for Grids*, *Advanced Programming Models for the Grid* and *Experimental Grid Infrastructures*.

Topic *P2P System Foundations*, that was described in the previous activity report, has been spinned-off to a new project-team, called *ASAP*, headed by Anne-Marie Kermarrec, a former member of the PARIS Project-Team.

The research activities of the PARIS Project-Team encompass both basic research, seeking conceptual advances, and applied research, to validate the proposed concepts against *real* applications. The project-team is also heavily involved in managing a national grid computing infrastructure (GRID'5000) enabling large-scale experiments.

2.1.1. Parallel processing to go faster

Given the significant increase of the performance of microprocessors, computer architectures and networks, clusters of standard personal computers now provide the level of performance to make numerical simulation a handy tool. This tool should not be used by researchers only, but also by a large number of engineers, designing complex physical systems. Simulation of mechanical structures, fluid dynamics or wave propagation can nowadays be carried out in a couple of hours. This is made possible by exploiting multi-level parallelism, simultaneously at a fine grain within a microprocessor, at a medium grain within a single multi-processor PC, and/or at a coarse grain within a cluster of such PCs. This unprecedented level of performance definitely makes numerical simulation available for a larger number of users such as SMEs. It also generates new needs and demands for more accurate numerical simulation. Traditional parallel processing alone cannot meet this demand.

2.1.2. Distributed processing to go larger

These new needs and demands are motivated by the constraints imposed by a worldwide economy: making things faster, better and cheaper.

2.1.2.1. Large-scale numerical simulation.

Large scale numerical simulation will without a doubt become one of the key technologies to meet such constraints. In traditional numerical simulation, only one simulation code is executed. In contrast, it is now required to *couple* several such codes together in a single simulation.

A large-scale numerical simulation application is typically composed of several codes, not only to simulate one physics, but to perform multi-physics simulation. One can imagine that the simulation times will be in the order of weeks and sometimes months depending on the number of physics involved in the simulation, and depending on the available computing resources.

Parallel processing extends the number of computing resources locally: it cannot significantly reduce simulation times, since the simulation codes will not be localized in a single geographical location. This is particularly true with the global economy, where complex products (such as cars, aircrafts, etc.) are not designed by a single company, but by several of them, through the use of subcontractors. Each of these companies brings its own expertise and tools such as numerical simulation codes, and even its private computing resources. Moreover, they are reluctant to give access to their tools as they may at the same time compete for some other projects. It is thus clear that distributed processing cannot be avoided to manage large-scale numerical applications

2.1.2.2. Resource aggregation.

More generally, the development of large scale distributed systems and applications now rely on resource sharing and aggregation. Distributed resources, whether related to computing, storage or bandwidth, are aggregated and made available to the whole system. Not only this aggregation greatly improves the performance as the system size increases, but many applications would simply not have been possible without such a model (peer-to-peer file sharing, ad-hoc networks, application-level multicast, publish-subscribe applications, etc.).

2.1.3. Scientific challenges of the Paris Project-Team

The design of large-scale simulation applications raises technical and scientific challenges, both in applied mathematics and computer science. The PARIS Project-Team mainly focuses its effort on Computer Science. It investigates new approaches to build software mechanisms that hide the complexity of programming computing infrastructures that are *both* parallel and distributed. Our contribution to the field can thus be summarized as follows:

combining parallel and distributed processing whilst preserving performance and transparency.

This contribution is developed along five directions.

Operating system and runtime for clusters and grids. The challenge is to design and build an operating system for clusters hiding to the programmers and the users, the fact that resources (processors, memories, disks) are distributed. A PC cluster with such an operating system looks like a traditional multi-processor running a Single System Image (SSI).

Middleware systems for computational grids. The challenge is to design a middleware implementing a component-based approach for grids. Large-scale numerical applications will be designed by combining together a set of components encapsulating simulation codes. The challenge is to seamlessly mix both parallel and distributed processing.

Large-scale data management for grids. One of the key challenges in programming grid computing infrastructures for real, is data management. It has to be carried out at an unprecedented scale, and to cope with the native dynamicity and heterogeneity of the underlying grids.

Advanced programming models for the Grid. This topic aims at contributing to study unconventional approaches for the programming of grids based on the *chemical metaphors*. The challenge is to exploit such metaphors to make the use, including the programming, of grids more intuitive and simpler.

Experimental Grid Infrastructures. The challenge here is to be able to design and to build an *instrument* (in the sense of a large scientific instrument, like a telescope) for computer scientists involved in grid research. Such an instrument has to be highly reconfigurable and scalable to several thousand of resources.

2.2. Operating system and runtime for clusters and grids

Clusters, made up of homogeneous computers interconnected via high-performance networks, are now widely used as general-purpose, high-performance computing platforms for scientific computing. While such an architecture is attractive with respect to its price/performance ratio, there still exists a large potential for efficiency improvement at the software level. System software can be improved to better exploit cluster hardware resources. Programming environments need to be developed with both the cluster and human programmer efficiency in mind.

We believe that cluster programming remains difficult. This is due to the fact that clusters suffer from a lack of dedicated operating system providing a single system image (SSI). A single system image provides the illusion of a single, powerful and highly-available computer to cluster users and programmers, as opposed to a set of independent computers, whose resources have to be managed locally.

Several attempts to build an SSI have been made at the middleware level as Beowulf [82], PVM [67] or MPI [77]. However, these environments only provide a *partial* SSI. Our approach in the PARIS Project-Team is to design and implement a *full* SSI in the operating system. Our objective is to combine ease of use, high performance and high availability. *All* physical resources (processor, memory, disk, etc.) and kernel resources (process, memory pages, data streams, files, etc.) need to be visible and accessible from *all* cluster nodes. Cluster reconfigurations due to a node addition, eviction or failure, need to be automatically dealt with by the system, transparently to the applications. Our SSI operating system (SSI OS) is designed to perform global, dynamic and integrated resource management.

As the execution time of scientific applications may be larger than the cluster mean time between failures, checkpoint/restart facilities need to be provided, not only for sequential applications but also for parallel applications. This is independent of the underlying communication paradigm. Even though backward error recovery (BER) has been extensively studied from the theoretical point of view, an efficient implementation of BER protocols, transparent to the applications, is still a research challenge. There are very few implementations of recovery schemes for parallel applications. Our approach is to identify and implement as part of the SSI OS, a set of building blocks that can be combined to implement various checkpointing strategies and their optimization for parallel applications, whatever inter-process communication (IPC) layer they use.

In addition to our research activity on operating system, we also study the design of runtimes for supporting parallel languages on clusters. A runtime is a software offering services dedicated to the execution of a particular language. Its objective is to tailor the general system mechanisms (memory management, communication, task scheduling, etc.) to achieve the best performance given the target machine and its operating system. The main originality of our approach is to use the concept of *distributed shared memory* (DSM) as the basic communication mechanism within the runtime. We are essentially interested in Fortran and its OpenMP extensions [58]. The Fortran language is traditionally used in the simulation applications we focus on. Our work is based on the operating system mechanisms studied in the PARIS Project-Team. In particular, the execution of OpenMP programs on a cluster requires a global address space shared by threads deployed on different cluster nodes. We rely on the two distributed shared memory systems we have designed: one at user level, implementing weak memory consistency models, and the other one at operating-system level, implementing the sequential consistency model.

2.3. Middleware systems for computational grids

Computational grids are very powerful machines as they aggregate huge computational resources. A lot of work has been carried out with respect to grid resource management. Existing grid middleware systems mainly focus on resource management like discovery, registration, security, scheduling, etc. However, they provide very little support for grid-oriented programming models.

A suitable grid programming model should be able to take into account the dual nature of a computational grid which is a distributed set of (mainly) parallel resources.

Our general objective is to propose such a programming model and to provide adequate middleware systems. Distributed object or component models seems to be a promising solution. However, they need to be tailored for scientific applications. In particular, the parallel applications have to be encapsulated into objects or components. New paradigms of communication between *parallel* objects or components have to be designed, together with the required runtime support, deployment facilities, and capacity for dynamic adaptability.

The first issue is the relationship between object or component models, which should handle the distributed nature of grid, and the parallelism of computational codes, which should take into account the parallelism of resources. It is thus required to efficiently integrate both worlds into a coherent, single vision.

The second issue concerns the simplicity and the scalability of communication between parallel codes. As the available bandwidth is larger than what a single resource could consume, parallel communication flows should allow a more efficient utilization of network resources. Advanced flow control should be used to avoid congesting networks. A crucial aspect of this issue is the support for data redistribution involved in the communication between parallel codes.

The third issue refers to the dynamic behavior of applications. While software component models are demonstrating their usefulness in capturing the static architecture of applications, there are still few results on how to deal with the dynamic aspects. The composition operator should be revised so as not to hide such dynamic aspects into the component implementation code.

Promoting a programming model that simultaneously supports distributed as well as parallel middleware systems, independently of the actual resources, raises three new issues. First, middleware systems should be decoupled from the actual networks so as to be deployed on any kind of network. Second, several middleware systems should be able to be *simultaneously* active within a same process. Third, the solutions to the two previous issues should meet the user requirements for high performance.

The deployment of applications is another issue. Not only is it important to specify the deployment in term of the computational resources (GFlop/s, amount of memory, etc.), but it is also crucial to specify the requirements related to communication resources, such as the amount of bandwidth, or the latency between computational resources. Moreover, we have to deal with applications integrating several distributed middleware systems, like MPI, CORBA, JXTA, etc.

The last issue deals with the dynamic nature of computational grids. As targeted applications may run for very long time, the grid environment is expected to change. Not only middleware systems should support adaptability, but they should also be able to detect variations and to self-adapt. For example, it should be possible to partially redeploy an application on the fly, to benefit from new resources.

2.4. Large-scale data management for grids

A major contribution of the grid computing environments developed so far is to have decoupled *computation* from *deployment*. Deployment is typically considered as an *external service* provided by the underlying infrastructure, in charge of locating and interacting with the physical resources. In contrast, as of today, no such sophisticated service exists regarding *data management* on the grid: the user is still left to explicitly store and transfer the data needed by the computation between these sites. Like deployment, we claim that an adequate approach to this problem consists in decoupling *data management* from *computation*, through an *external service* tailored to the requirements of scientific applications. We focus on the case of a grid consisting of a federation of distributed clusters. Such a *data sharing service* should meet two main properties: *persistence* and *transparency*.

First, the data sets used by the grid computing applications may be very large. Their transfer from one site to another may be costly (in terms of both bandwidth and latency), so that such data movements should be carefully optimized. Therefore, the data management service should allow data to be *persistently* stored on the grid infrastructure independently of the applications, in order to allow their reuse in an efficient way.

Second, a data management service should provide *transparent* access to data. It should handle data localization and transfer without any help from the programmer. Yet, it should make good use of additional information and hints provided by the programmer, if any. The service should also transparently use adequate replication strategies and consistency protocols to ensure data availability and consistency in a large-scale, dynamic architecture.

Given that our target architecture is a federation of clusters, several additional constraints need to be addressed. The clusters which make up the grid are not guaranteed to remain available constantly. Nodes may leave due to technical problems or because some resources become temporarily unavailable. This should obviously not result in disabling the data management service. Also, new nodes may dynamically join the physical infrastructure: the service should be able to dynamically take into account the additional resources they provide. Therefore, adequate strategies need to be set up in order for the service to efficiently interact with the resource management system of the grid.

On the other hand, it should be noted that the algorithms proposed for parallel computing have often been studied on small-scale configurations. Our target architecture is typically made of thousands of computing nodes, say tens of hundred-node clusters. It is well-known that designing low-level, explicit MPI programs is most difficult at such a scale. In contrast, peer-to-peer approaches have proved to remain effective at a large scale, and can serve as fruitful inspiration sources.

Finally, data is generally shared in grid applications, and can be modified by multiple partners. Traditional replication and consistency protocols designed for DSM systems have often made the assumption of a small-scale, static, homogeneous architecture. These hypotheses need to be revisited and this should lead to new consistency models and protocols adapted to a dynamic, large-scale, heterogeneous architecture.

2.5. Advanced programming models for the Grid

Till now, research activities related to the grid have focused on the design and implementation of middleware and tools to experiment grid infrastructure with applications. Little attention has been paid to programming models suitable for such widely computing infrastructures. Programming such infrastructures is still done at a very low level. This situation may somehow be compared to using assembly language to program complex processors. Our objective is to study approaches for grid programming that do not expose the architectural details of the computing infrastructure to the programmers. More specifically, we are considering unconventional approach based on the *chemical reaction* paradigm, and more precisely the GAMMA Model [64].

GAMMA is based on multiset rewriting. The unique data structure in GAMMA is the multiset (a set than can contain several occurrences of the same element), which can be seen as a *chemical solution*. A simple program is a set of rules $Reaction\ \ condition \rightarrow Action$. Execution proceeds, without any explicit order, by replacing elements in the multiset satisfying the reaction condition by the products of the action (*chemical reaction*). The result is obtained when a stable state is reached, that is, when no more reactions applies. Our objective is to express the coordination of Grid components or services through a set of rules, while the multiset represents the services that have to be coordinated.

2.6. Experimental Grid Infrastructures

The PARIS Project-Team is engaged in research along five research topics: *Operating System and Runtime for Clusters and Grids*, *Middleware Systems for Computational Grids*, *Large-scale Data Management for Grids*, *Advanced Programming Models for the Grid* and *Experimental Grid Infrastructures*. The concepts proposed by each of these topics must be validated against real applications on realistic hardware. The project-team manages a computation platform dedicated to operating system and middleware experimentations. This platform is integrated within GRID'5000, a national computing infrastructure dedicated to large-scale Grid and peer-to-peer experiments. The GRID'5000 infrastructure federates experimental platforms (currently 9 platforms) across France. These platforms are connected through Renater using dedicated 10 Gigabit/s Ethernet links.

Our experimental platform is maintained up to date through periodic replacement of groups of nodes on a 1-2 year basis. It used to be heterogeneous (PowerPC and PC families of processors, 32-bit and 64-bit architectures, Linux and MacOS X operating systems) in the past with a major block of 64-bit Linux/PC boxes. It is now more homogeneous in processor/operating system types: only 64-bit PCs running Linux. However, the advent of multicore nodes introduces another form of heterogeneity in the nodes: the number of cores currently varies from 1 to 4. All nodes are locally connected through a 1 Gb/s Ethernet switch. They are connected with the other sites through a dedicated 10 Gb/s optical uplink managed by Renater. A group of 96 nodes are moreover connected with an extra Myrinet 10 Gb/s local network, and another group of 64 nodes with an InfiniBand network.

Our experimental platform is dedicated to operating system and middleware experimentation. It is possible to repeat experiments in a fully controlled environment (same machines, same network, etc.). The allocation of the resources to the experiments is handled through *OAR*, a job manager developed by our partners from the Grenoble site.

3. Scientific Foundations

3.1. Introduction

Research activity within the PARIS Project-Team encompasses several areas: operating systems, middleware and programming models. We have chosen to provide a brief presentation of some of the scientific foundations associated with them.

3.2. Data consistency

A shared virtual memory system provides a global address space for a system where each processor has only physical access to its local memory. Implementing of such a concept relies on the use of complex cache coherence protocols to enforce data consistency. To allow the correct execution of a parallel program, it is required that a read access performed by one processor returns the value of the last write operation previously performed by any other processor. Within a distributed or parallel a system, the notion of the *last* memory access is sometimes partially defined only, since there is no global clock to provide a total order of the memory operation.

It has always been a challenge to design a shared virtual memory system for parallel or distributed computers with distributed physical memories, capable of providing comparable performance with other communication models such as message-passing. *Sequential Consistency* [74] is an example of a memory model for which all memory operations are consistent with a total order. Sequential Consistency requires that a parallel system having a global address space appears to be a multiprogramming uniprocessor system to any program running on it. Such a strict definition impacts on the performance of shared virtual memory systems due to the large number of messages that are required (page access, invalidation, control, etc.). Moreover Sequential Consistency is not necessarily required to correctly run parallel programs, in which memory operations to the global address space are guarded by synchronization primitives.

Several other memory models have thus been proposed to relax the requirements imposed by sequential consistency. Among them, *Release Consistency* [68] has been thoroughly studied since it is well adapted to programming parallel scientific applications. The principle behind Release Consistency is that memory accesses are (should?) always be guarded by synchronization operations (locks, barriers, etc.), so that the shared memory system only needs to ensure consistency at synchronization points. Release Consistency requires the use of two new operations: *acquire* and *release*. The aim of these two operations is to specify when to propagate the modifications made to the shared memory systems. Several implementations of Release Consistency have been proposed [72]: an *eager* one, for which modifications are propagated at the time of a release operation; and a *lazy* one, for which modifications are propagated at the time of an acquire operation. These alternative implementations differ in the number of messages that needs to be sent/received, and in the complexity of their implementation [73].

Implementations of Release Consistency rely on the use of a logical clock such as a vector clock [76]. One of the drawback of such a logical clock is its lack of scalability when the number of processors increases, since the vector carries one entry per processor. In the context of computing systems that are both parallel and distributed, such as a grid infrastructure, the use of a vector clock is impossible in practice. It is thus necessary to find new approaches based on logical clocks that do not depend on the number of processors accessing the shared memory system. Moreover, these infrastructures are natively *hierarchical*, so that the consistency model should better take advantage of it.

3.3. High availability

> "A distributed system is one that stops you getting any work done when a machine you've never even heard about crashes." (Leslie Lamport)

The *availability* [69] of a system measures the ratio of service accomplishment conforming to its specifications, with respect to elapsed time. A system *fails* when it does not behave in a manner consistent with its specifications. An error is the consequence of a *fault* when the faulty part of the system is activated. It may lead to the system *failure*. In order to provide highly-available systems, *fault tolerance techniques* [75] based on redundancy can be implemented. Abstractions like *group membership*, *atomic multicast*, *consensus*, etc. have been defined for fault-tolerant distributed systems.

Error detection is the first step in any fault tolerance strategy. *Error treatment* aims at avoiding that the error leads to the system failure.

Fault treatment consists in avoiding that the fault be activated again. Two classes of techniques can be used for fault treatment: *reparation* which consists in eliminating or replacing the faulty module; and *reconfiguration* which consists in transferring the load of the faulty element to valid components.

Error treatment can be of two forms: *error masking* or *error recovery*. Error masking is based on hardware or software redundancy in order to allow the system to deliver its service despite the error. Error recovery consists in restoring a correct system state from an erroneous state. In *forward error recovery* techniques, the erroneous state is transformed into a safe state. *Backward error recovery* consists in periodically saving the system state, called a *checkpoint*, and rolling back to the last saved state if an error is detected.

A *stable storage* guarantees three properties in presence of failures: (1) *integrity*, data stored in stable storage is not altered by failures; (2) *accessibility*, data stored in stable storage remains accessible despite failures; (3) *atomicity*, updating data stored in stable storage is an all or nothing operation. In the event of a failure during the update of a group of data stored in stable storage, either all data remain in their initial state or they all take their new value.

3.4. Distributed data management

Past research on distributed data management led to three main approaches. Currently, the most widely-used approach to data management for distributed grid computation relies on *explicit data transfers* between clients and computing servers. As an example, the *Globus* [56] platform provides data access mechanisms (like data catalogs) based on the *GridFTP* protocol. Other explicit approaches (e.g., *IBP*) provide a large-scale data storage system, consisting of a set of buffers distributed over Internet. The user can "rent" these storage areas for efficient data transfers.

In contrast, *Distributed Shared Memory* (DSM) systems provide *transparent* data sharing, via a virtual, unique address space accessible to physically distributed machines. It is the responsibility of the DSM system to localize, transfer, replicate data, and guarantee their consistency according to some semantics. Within this context, a variety of consistency models and protocols have been defined. Nevertheless, existing DSM systems have generally shown satisfactory efficiency only on small-scale configurations, up to a few tens of nodes.

Recently, *peer-to-peer* (P2P) has proven to be an efficient approach for large-scale resource (data or computing resources) sharing [78]. The peer-to-peer communication model relies on a symmetric relationship between peers which may act both as clients and servers. Such systems have proven able to manage very large and dynamic configurations (millions of peers). However, several challenges remain. More specifically, as far as data sharing is concerned, most P2P systems focus on sharing *read-only* data, that do not require data consistency management. Some approaches, like *OceanStore* and *Ivy*, deal with *mutable* data in a P2P with restricted use. Today, one major challenge in the context of large-scale, distributed data management is to define appropriate models and protocols allowing to guarantee both *consistency* of replicated data and *fault tolerance*, in *large-scale, dynamic environments*.

3.5. Component model

Software component technology [83] has been emerging for some years, even though its underlying intuition is not very recent. Building an application based on components emphasizes programming by *assembly*, that is, *manufacturing*, rather than by *development*. The goals are to focus expertise on domain fields, to improve software quality, and to decrease the time-to-market thanks to reuse of existing codes.

The CORBA Component Model (CCM), which is part of the latest CORBA [80] specifications (Version 3), appears to be the most complete specification for components. It allows the deployment of a set of components into a distributed environment. Moreover, it supports heterogeneity of programming languages, operating systems, processors, and it also guarantees interoperability between different implementations. However, CCM does not provide any support for parallel components.

The CORBA Component Architecture (CCA) Forum [60] aims at developing a standard which specifically addresses the needs of the HPC community. Its objective is to define a minimal set of standard interfaces that any high-performance component framework should provide to components, and may expect from them, in order to allow disparate components to be composed together into a running application. CCA aims at supporting *both* parallel and distributed applications.

3.6. Adaptability

Due to the dynamic nature of large-scale distributed systems in general, and the Grid in particular, it is very hard to design an application that fits well in any configuration. Moreover, constraints such as the number of available processors, their respective load, the available memory and network bandwidth are not static. For these reasons, it is highly desirable that an application could take into account this dynamic context in order to get as much performance as possible from the computing environment.

Dynamic adaptation of a program is the modification of its behavior according to changes of the environment. This adaptivity can be achieved in many different ways, ranging from a simple modification of some parameters, to the total replacement of the running code. In order to achieve adaptivity, a program needs to be able to get information about the environment state, to make a decision according to some optimization rules, and to modify or replace some parts of its code.

Adaptivity has been implemented by designing ad hoc applications that take into account the specificities of the target environment. For example, this was done for the Web applications access protocol on mobile networks by defining the WAP protocol [59]. A more general way is to provide mechanisms enabling dynamic self-adaptivity by changing the program's behavior. In most cases, this has been achieved by embedding the adaptation mechanism within the application code. For example, the AdOC compression algorithm [71] includes such a mechanism to dynamically change the compression level according to the available resources.

However, it is desirable to separate the adaptation engine from the application code, in order to make the code easier to maintain, and to easily change or improve the adaptation policy. This was done for wireless and mobile environments by implementing a framework [66] that provides generic mechanisms for the adaptation process, and for the definition of the adaptation rules.

3.7. Chemical programming

The chemical reaction metaphor has been discussed in various occasions in the literature. This metaphor describes computation in terms of a chemical solution in which molecules (representing data) interact freely according to reaction rules. Chemical models use the multiset as their basic data structure. Computation proceeds by rewritings of the multiset which consume elements according to reaction conditions and produce new elements according to specific transformation rules.

To the best of our knowledge, the GAMMA formalism was the first "chemical model of computation" proposed as early as in 1986 [63] and extended later [64].

A GAMMA program is a collection of reaction rules acting on a multiset of basic elements. A reaction rule is made of a condition and an action. Execution proceeds by replacing elements satisfying the reaction condition by the elements specified by the action. The result of a GAMMA program is obtained when a stable state is reached that is to say when no more reactions can take place. Here is an example illustrating the GAMMA style of programming:

$$primes = \mathbf{replace} x, y \mathbf{by} y \mathbf{if} multiple(x, y)$$

The reaction $primes$ computes the prime numbers lower or equal to a given number N when applied to the multiset of all numbers between 2 and N ($multiple(x, y)$ is true if and only if x is a multiple of y). Let us emphasize the conciseness and elegance of these programs. Nothing had to be said about the order of evaluation of the reactions. If several disjoint pairs of elements satisfy the condition, the reactions can be performed in parallel.

GAMMA makes it possible to express programs without artificial sequentiality. By artificial, we mean sequentiality only imposed by the computation model and unrelated to the logic of the program. This allows the programmer to describe programs in a very abstract way. In some sense, one can say that GAMMA programs express the very idea of an algorithm without any unnecessary linguistic idiosyncrasies. The interested reader may find in [64] a long series of examples (string processing problems, graph problems, geometry problems, etc.) illustrating the GAMMA style of programming and in [62] a review of contributions related to the chemical reaction model. Later, the idea was developed further into the CHAM [65], the P-systems [81], etc. Although built on the same basic paradigm, these proposals have different properties and different expressive powers.

The γ-calculus [61] is an attempt to identify the basic principles behind chemical models. It exhibit a minimal chemical calculus, from which all other "chemical models" can be obtained by addition of well-chosen features. Essentially, this minimal calculus incorporates the γ-reduction which expresses the very essence of the chemical reaction, and the associativity and commutativity rules which express the basic properties of chemical solutions.

4. Application Domains

4.1. Application Domains

Keywords: *Scientific computing*, *co-operative applications*, *large-scale computing*.

The project-team research activities address scientific computing and specifically numerical applications that require the execution of several codes simultaneously. This kind of applications requires both the use of parallel and distributed systems. Parallel processing is required to address performance issues. Distributed processing is needed to fulfill the constraints imposed by the localization and the availability of resources, or for confidentiality reasons. Such applications are being experimented within contracts with the industry or through our participation to application-oriented research grants.

5. Software

5.1. Kerrighed

Keywords: *Cluster operating system*, *checkpointing*, *distributed shared memory (DSM)*, *distributed file system*, *global scheduling*, *high availability*, *process migration*, *single system image (SSI)*.

Participants: Matthieu Fertré, Jérôme Gallard, Adrien Lèbre, Christine Morin, Jean Parpaillon.

Contact: Christine Morin, `Christine.Morin@irisa.fr`

URL: http://www.kerrighed.org/ and http://ssi-oscar.gforge.inria.fr/

Status: Registered at APP, under Reference `IDDN.FR.001.480003.006.S.A.2000.000.10600`.

License: GNU General Public License (GPL) version 2. KERRIGHED is a registered trademark.

Presentation: KERRIGHED is a *Single System Image* (SSI) operating system for high-performance computing on clusters. It provides the user with the illusion that a cluster is a virtual SMP machine.

In KERRIGHED, all resources (processes, memory segments, files, data streams) are globally and dynamically managed to achieve the SSI properties. Global resource management makes distribution of resources transparent throughout the cluster nodes, and allows to take advantage of the whole cluster hardware resources for demanding applications. Dynamic resource management enables transparent cluster reconfigurations (node addition or eviction) for the applications, and high availability in the event of node failures. In addition, a checkpointing mechanism is provided by KERRIGHED to avoid restarting applications from the beginning when some node failure occurs.

KERRIGHED preserves the interface of a standard, single-node operating system, which is familiar to programmers. Legacy sequential or parallel applications running on this standard operating system can be executed without modification on top of KERRIGHED, and further optimized if needed.

KERRIGHED is not an entirely new operating system developed from scratch. Just in the opposite, it has been designed and implemented as an extension to an existing standard operating system. KERRIGHED only addresses the distributed nature of the cluster, while the native operating system running on each node remains responsible for the management of local physical resources. Our current prototype is based on *Linux*, which is extended using the standard module mechanism. The Linux kernel itself has only been slightly modified.

A public mailing list (`kerrighed.users@irisa.fr`) and a technical forum are available to provide a support to KERRIGHED users.

Current status: KERRIGHED (version V2.2.0) includes 40,000 lines of code (mostly in C). It involved more than 250 persons-months.

This version of KERRIGHED provides the illusion of a virtual multiprocessor. Based on Linux 2.6.20 kernel, it relies on the TIPC communication system and supports SMP nodes.

In 2007, KERRIGHED has been ported to Linux 2.6.20. The code has significantly been improved during this port, resulting in a more compact software. Moreover, KERRIGHED is also distributed as an *official* spin-off OSCAR package with the SSI-OSCAR package. The SSI-OSCAR packages based on the development version of KERRIGHED and OSCAR 5.0, are available for Linux distributions supported by OSCAR (e.g., Fedora Core 5, RedHat Enterprise Linux 4, etc.) and for the Debian Linux distribution.

Demonstrations of KERRIGHED have been presented in 2007 at *Linux Expo* (Paris, February 2007, J. Parpaillon), and *Supercomputing 2007 Conference* (Reno, Nevada, November 2007, A. Lèbre, Ch. Morin).

5.2. PaCO++

Keywords: *CORBA*, *Grid*, *data parallelism*, *middleware system*.

Participants: Raúl López Lozano, Christian Pérez, Thierry Priol.

Contact: Christian Pérez, `Christian.Perez@inria.fr`

URL: http://paco.gforge.inria.fr/

Status: Registered at APP, under Reference `IDDN.FR.001.450014.000.S.P.2004.000.10400`.

License: GNU General Public License (GPL) version 2 and GNU Lesser General Public License (LGPL) version 2.1.

Presentation: The PACO++ objective is to allow a simple and efficient embedding of a SPMD code into a parallel CORBA object, and to allow parallel communication flows and data redistribution during an operation invocation on such a parallel CORBA object.

PACO++ provides an implementation of the concept of parallel object applied to CORBA. A parallel object is an object whose execution model is parallel. It is externally accessible through an object reference, whose interpretation is identical to a standard CORBA object.

PACO++ extends CORBA, but does not modify the underlying model. It is meant to be a *portable* extension to CORBA, so that it can be added to any CORBA implementation. The parallelism of an object is in fact considered to be an implementation feature of this object, and the OMG IDL is not dependent on it.

PACO++ is made of two components: a compiler and a runtime library.

- The compiler generates parallel CORBA stub and skeleton from an IDL file which describes the CORBA interface, and from an XML file which describes the parallelism of the interface. The compilation is done in two steps. The first step involves a Java IDL-to-IDL compiler based on *SableCC*, a compiler of compiler, and *Xerces* for the XML parser. The second part, written in Python, generates the stubs files from templates configured with inputs generated during the first step.
- The runtime, currently written in C++, deals with the parallelism of the parallel CORBA object. It is very portable thanks to the utilization of abstract APIs for communications, threads and redistribution libraries.

Current status: The development of PACO++ started at the end of 2002. It involved 60 persons-months. The first public version, referenced as PACO++ 0.1 has been released in November 2004. The second version (0.2) has been released in March 2005. It has been successfully tested on top of three CORBA implementations: *Mico*, *omniORB3* and *omniORB4*. Moreover, it supports PADICOTM, an open integration framework for communication middleware and runtime systems developed in the PARIS Project-Team, which enables several middleware systems (such as CORBA, MPI, SOAP, etc.) to be used at the same time.

The version 0.2 of PACO++ includes 63,000 lines of Java (around 1.5 MB), 7,800 lines of Python (around 436 kB), 16,000 lines of C++ (around 390 kB) and 2,000 lines of `shell`, `make` and `configure` scripts (60 kB).

PACO++ has been supported by the RMI Project of the French ACI GRID program. It has been used, or it is used, by several other French projects: ACI GRID HydroGrid, ACI GRID EPSN, RNTL VTHD ++ and INRIA ARC RedGrid. It is currently used within two French ANR CI projects: DISC and NUMASIS.

PACO++ is co-developed with the EDF R&D company.

It has been downloaded 124 times, from 48 unique IPs.

5.3. Adage

Keywords: *Grid*, *deployment*, *middleware system*.

Participants: Landry Breuil, Loïc Cudennec, Christian Pérez, Thierry Priol.

Contact: Christian Pérez, `Christian.Perez@inria.fr`
URL: http://www.irisa.fr/paris/ADAGE/
Status: Registered at APP, under Reference `IDDN.FR.001.270020.000.S.P.2007.000.10000`.
License: GNU General Public License (GPL) version 2.
Presentation: ADAGE (*Automatic Deployment of Applications in a Grid Environment*) is a research prototype that aims at studying the deployment issues related to multi-middleware applications. Its original contribution is to use a *generic* application description model (*GADe*) to transparently handle various middleware systems.

With respect to application submission, ADAGE requires an application description, which is specific to a programming model, a reference to a resource information service (MDS2, or an XML file), and a control parameter file. The application description is internally translated into a generic description, so as to support multi-middleware applications. The control parameter file allows a user to express constraints on the placement policy, which is specific to an execution. For example, a constraint may specify the latency and the bandwidth between a computational component and a visualization component.

The support of multi-middleware applications is based on a plug-in mechanism. The plug-in is involved in the conversion from the specific to the generic application description, but also during the execution phase so as to deal with specific middleware configuration actions.

ADAGE currently deploys static applications only. It supports standard programming models like MPI (*MPICH1*, *MPICH2* and *OpenMPI*), CCM, JXTA, and Gfarm.

Current status: The version 0.2 of ADAGE includes 22,000 lines of C++. It is a complete re-implementation of ADAGE 0.1 based on well defined specification. It has been registered at APP in June 2007 and the public release has been delivered in September 2007.

It has been download 35 times, from 14 unique IPs.

5.4. JuxMem

Keywords: *JXTA*, *Peer-to-peer*, *data grids*, *large-scale data management*.
Participants: Gabriel Antoniu, Luc Bougé, Landry Breuil, Loïc Cudennec, Mathieu Jan, Sébastien Monnet.

Contact: Gabriel Antoniu, `Gabriel.Antoniu@irisa.fr`
URL: http://juxmem.gforge.inria.fr/
License: GNU Lesser General Public License (LGPL) version 2.1.
Status: Registered at APP, under Reference `IDDN.FR.001.180015.000.S.P.2005.000.10000`.
Presentation: JUXMEM is a supportive platform for a data-sharing service for grid computing. This service addresses the problem of managing mutable data on dynamic, large-scale configurations. It can be seen as a hybrid system combining the benefits of *Distributed Shared Memory* (DSM) systems (transparent access to data, consistency protocols) and *Peer-to-Peer* (P2P) systems (high scalability, support for resource volatility). The target applications are numerical simulations, based on code coupling, with significant requirements in terms of data storage and sharing. JUXMEM's architecture decouples fault-tolerance management from consistency management. Multiple consistency protocols can be built using fault-tolerant building blocks such as *consensus*, *atomic multicast*, *group membership*. Currently, a hierarchical protocol implementing the entry consistency model is available. A more relaxed consistency protocol adapted to visualization is also available.

Current status: Two implementations are available, in Java and C. JuxMem is based on the *JXTA* generic platform for P2P services (Sun Microsystems, http://www.jxta.org/). At this time, it includes 16,700 lines of Java code and 16,000 lines of C code. Implementation started in February 2003. The first public version, referenced as JUXMEM 0.1 has been released in April 2005. The Java version is no longer supported. In 2007, progress has been made on: 1) adding a prototype database-oriented API; 2) adding persistence support on secondary storage using the Gfarm global file system (AIST, Tsukuba, Japan, http://datafarm.apgrid.org/).

JUXMEM has been the central framework for the GDS (*Grid Data Service*) project of the ACI MD Program, ended in 2006. JUXMEM is currently used for transparent data sharing within the following running projects: ANR CI LEGO project, and ANR MD RESPIRE project. An industrial collaboration with Sun Microsystems has been started in August 2005 for 3 years. JUXMEM is currently used within several international collaborations: AIST (Tsukuba, Japan), University of Pisa, University of Calabria. Other past users: University of Illinois at Urbana Champaign.

5.5. Dynaco

Keywords: *Grid*, *components*, *framework*, *objects*.

Participants: Françoise André, Jérémy Buisson, Jean-Louis Pazat.

Contact: Jérémy Buisson, `Jeremy.Buisson@irisa.fr`

URL: http://dynaco.gforge.inria.fr/

Status: Version 0.2 is available.

License: GNU Lesser General Public License (LGPL) version 2.1.

Presentation: DYNACO (*Dynamic Adaptation for Components*) is a framework that helps in designing and implementing dynamically adaptable components. This framework is developed by the PARIS Project-Team. The implementation of DYNACO is based on the *Fractal Component Model* and its formalism.

In DYNACO, the process of achieving dynamic adaptation is split over three phases:

- Upon the reception of an event that notifies of a change in certain conditions, the component has to make a decision: should it adapt itself to the new situation or not? To do so, it can rely on monitors in order to observe the system. This decision phase is captured by the *Decider Component*.
- Once it has been decided that the component should adapt itself, the component needs to investigate how the adaptation can be achieved. In particular, it has to design the list of the tasks that should be performed. This phase is captured by the *Planner Component*.
- Finally, this adaptation plan has to be executed. The *Executor Component* is the virtual machine that implements the semantics of the instructions used by the Planner Component. To do so, it can rely on the *Modification Controller Components*, which implement some primitive instructions by giving a direct access to the content of the components.

DYNACO mainly defines interfaces between those components. In addition, it includes a reference implementation for the *Julia* implementation of *Fractal*. With this implementation, only the Modification Controller Components are placed in the membrane of the adaptable component.

When the contents of the component encapsulates a parallel code, the Executor Component has to take care of the synchronization between the parallel processes executing the applicative code and the adaptation actions. Our solution for handling this problem relies on a separated framework, called AFPAC.

5.6. Mome

Keywords: *DSM*, *large scale data repository*.

Participant: Yvon Jégou.

Contact: Yvon Jégou, `Yvon.Jegou@irisa.fr`

URL: http://www.irisa.fr/paris/Mome/welcome.htm

License: APP registration in the future, license type not yet defined (LGPL?).

Presentation: The MOME DSM provides a shared segment space to parallel programs running on distributed memory computers or clusters. Individual processes can freely request mappings between their local address space and MOME segments. The initial implementation of MOME has been completely revised in order to address the major limitations of the basic version: limited size of shared address space, static management of meta-data, restricted number of nodes, static management of nodes on the grid, etc.

MOME now provides a hierarchical management of the local objects (page managers, synchronization objects): each object manager is still in charge of a limited number of clients, but it can now depend on a another manager inside a manager hierarchy. All internal components are created and connected "on-the-fly", and can be reclaimed when no longer in use. On-the-fly creation of object managers limits the effective memory used by object meta-data on some computation node to the objects active on this node. Dynamic management of object managers also greatly reduces the startup time requested by meta-data initialization. The dynamic management of computation nodes allows nodes to be integrated to, or removed from an existing computation, and reduces the startup time (the computation can start even when all nodes are not connected yet).

Current status: The initial implementation of MOME (50,000 lines of C code) is no longer supported. The full specification of MOME 1 has been finalised and the software implementation is on-going.

5.7. Vigne

Contact: Christine Morin, `Christine.Morin@irisa.fr`

URL: http://www.irisa.fr/paris/web/GridOS.html

Status: Version 1.0 soon available

License: GNU General Public License (GPL).

Presentation: VIGNE is a prototype of a grid-aware operating system for grids, whose goal is to ease the use of computing resources in a grid for executing distributed applications. VIGNE is made up of a set of operating system services based on a peer-to-peer infrastructure. This infrastructure currently implements a structured overlay network inspired from *Pastry* and an unstructured overlay network inspired from *Scamp* for join operations. On top of the structured overlay network, a transparent data-sharing service based on the sequential consistency model has been implemented. It is able to handle an arbitrary number of simultaneous reconfigurations. An application execution management service has also been implemented including resource discovery, resource allocation, and application monitoring services. The VIGNE prototype is coupled with a discrete event simulator.

In 2007, the VIGNE prototype has been extended in two ways. First of all, the application management service has been extended in order to handle several patterns of distributed applications like code coupling or workflow applications. Second, the discrete event simulator of the VIGNE prototype has been extended to model the workload of tasks. It allows to rigorously compare several resource discovery protocols implemented in VIGNE, using the simulation mode where the experimental conditions are reproducible. Moreover, VIGNE has been experimented in the framework of the *SALOME* integration platform for numerical simulation (http://www.salome-platform.org/), allowing running experiments with a real workflow application at the EDF R&D company.

Current status: The VIGNE prototype has been developed in C and includes 30,000 lines of code. This prototype has been coupled with a discrete-event simulator. The use of this simulator enabled to evaluate the VIGNE system in systems composed of a large number of nodes.

6. New Results

6.1. Introduction

Research results are presented according to the scientific challenges of the PARIS Project-Team, in connection with the *CoreGRID Network of Excellence*, in which PARIS is actively involved.

6.2. Operating system and runtime for clusters and grids

6.2.1. Cluster operating systems

Participants: Matthieu Fertré, Jérôme Gallard, Adrien Lèbre, Christine Morin, Jean Parpaillon.

A Single System Image (SSI) OS provides the illusion that a distributed cluster is a virtual multiprocessor machine. Therefore, it considerably eases the cluster use, programming and management. In particular, legacy applications can be executed without modification on top of a SSI. Since 2006, the Linux-based KERRIGHED SSI OS is developed within a open source community (http://www.kerrighed.org/) and industrialized by KERLABS, a spin-off from the PARIS Project-Team created in October 2006. In 2007, we continued to contribute to the design and implementation of KERRIGHED in the framework of the XTREEMOS European IP project.

In 2007, we designed a distributed implementation of the standard Posix IPC interface in KERRIGHED (message queues, semaphores, etc.). We also revisited a previous implementation of checkpoint/restart mechanisms for individual processes in KERRIGHED, taking into account the KERRIGHED refactoring done in 2006, and improving their robustness. These contributions are integrated in KERRIGHED version 2.2.0 official release.

We also worked on the design and implementation of *kDFS* (kernel/KERRIGHED Distributed File System), a distributed file system exploiting the disks attached to the computing nodes of a cluster. One of the main ideas consists in developing a distributed file system that is pluggable into the *Linux Virtual File System* (VFS) and is only based on the KERRIGHED *KDDM* communication service. This service provides mechanisms to share kernel level data cluster-wide. KDDM is used in kDFS to build a cooperative cache for both data and meta-data.

A first prototype has been released in November 2007 (5,000 lines of C code in kernel space). It allows basic cluster-wide file management. We have started to implement data striping mechanisms and policies to improve kDFS efficiency in executing parallel applications. We have also started to design I/O probes that will be used by the global scheduler to implement load-balancing scheduling policies, taking into account file data localization.

A SSI OS such as KERRIGHED is implemented by a set of distributed services. The configuration of a cluster may evolve when for instance an administrator adds or stops one or several cluster nodes while the SSI is up and running applications. In collaboration with Pascal Gallard from KERLABS, we worked on the design and implementation of KERRIGHED reconfiguration mechanisms. The hot-node addition feature has been implemented and is now operational. The hot-node eviction feature is under implementation. Future work in this area is to extend the reconfiguration service in order to be able to automatically reconfigure KERRIGHED services in the event of node failures.

In the context of Jérôme Gallard's Master internship, we designed a global scheduler for a Single System Image SSI OS to self-regulate the cluster load. The proposed scheduler is able to select an execution node when a process is started, on a process migration from one node to another during its execution, on suspending, stopping or restarting the execution of a process based on resource usage (processor, disk, network, memory, etc.). It provides a generic framework to define load management policies. A prototype has been implemented in KERRIGHED and evaluations demonstrated the benefits of this approach in terms of performance [53].

Even if SSI solutions are usually more complete in terms of functionality, batch schedulers are usually preferred because of their simplicity in terms of both configuration and usage. Moreover, since a few years, combining virtual machines and batch systems provides more advanced resource management capabilities, using features such as virtual machine live migration. Because of the latest contributions in the domain, some may argue that SSI technologies are now deprecated.

We analyzed whether virtualization technologies would overcome the SSI approach, and the extent at which these two models are complementary. In fact, after evaluating different configurations, we showed that combining these approaches allows to improve several aspects of application management, such as flexibility of administration, simplicity of use, security and portability [39]. We plan to experiment various configurations combining virtualization technologies with KERRIGHED SSI OS to evaluate the potential gain in performance. Another direction of work is to extend the scope of this study to investigate how virtual machines can be used in the framework of a Grid operating system such as XTREEMOS in order to provide strong application isolation, to manage in flexible way heterogeneous application execution contexts, and to adapt to the dynamicity of Grid environments.

In 2007, we contributed to several new releases of KERRIGHED. The first version of KERRIGHED for Linux 2.6 has been released in April (Kerrighed 2.0 on Linux 2.6.11). The 2.1, 2.1.1 and 2.2.0 versions have also been released this year, making KERRIGHED available for Linux 2.6.20. We contributed to the packaging of KERRIGHED for RPM-based Linux distributions in collaboration with NEC and Mandriva, and also for the Debian distribution. All KERRIGHED versions released in 2007 are available as Debian packages on the KERRIGHED website. The KERRIGHED website has been extensively redesigned. It is now based on a wiki and a forum with a common look-and-feel. Moreover, installation and user manuals have been regularly updated. KERRIGHED *manual pages* have also been made available on the KERRIGHED website.

Since three years, Kerrighed has been distributed through the OSCAR software suite for high-performance computing on Linux clusters.

INRIA have officially joined the *Open Cluster Group* http://oscar.openclustergroup.org/ in 2007, rewarding years of contributions, mostly through the SSI-OSCAR project. J. Parpaillon has been appointed Release Manager for version 6.0 of OSCAR. This version will include major architecture changes in the package management. We designed a new package format and implemented a package compiler which produces packages in the format of supported distributions (RPM or Debian) from the OSCAR package description.

OSCAR websites, formerly made of two wikis and a website based on *Drupal*, have been merged into a single one (http://oscar.openclustergroup.org/), based on the *trac* environment, in order to ease communication and contributions.

In 2007, in the framework of Nicolas Aupetit's internship, we improved the platform used to perform non-regression tests on KERRIGHED software. It is now possible to automatically deploy the KERRIGHED development version on a GRID'5000 cluster directly from the source code available in the development repository. This allows us to perform not only compilation tests, but also execution tests running the standard *LTP* test suite to check conformance to Posix, and the *KTP* test suite dedicated to the test of functionalities specific to KERRIGHED.

6.2.2. Grid operating systems

Participants: Matthieu Fertré, Emmanuel Jeanvoine, Yvon Jégou, Sylvain Jeuland, Adrien Lèbre, Pascal Le Métayer, Sandrine L'Hermitte, David Margery, Christine Morin, Thomas Ropars, Oscar Sanchez.

6.2.2.1. Vigne, a system for large-scale, dynamic Grids

Our research aims at easing the execution of distributed computing applications on computational grids. These grids are composed of a large number of geographically-distributed computing resources. This large-scale distribution makes the system dynamic: failures of single resources are frequent (both network and machine failures), and any participating entity may decide at any time to add or remove nodes from the grid.

To ease the use of such dynamic, distributed systems, we propose to build a distributed operating system which provides a Single System Image (SSI), which is self-healing, and which can be tailored to the needs of the users. Such an operating system is composed of a set of distributed services, each of them providing a Single System Image for a specific type of resource, in a fault-tolerant way. We are implementing this system on a research prototype called VIGNE. Experimental evaluations are made on the GRID'5000 research grid. The work of Year 2007 is twofold.

First, we have proposed a generic way to describe the most common patterns of distributed applications. The application management service embeds an engine designed to execute the tasks of an application according to three relationships between the tasks (precedence, synchronization, spatial) [30]. Furthermore, no modification is required in the application codes. This work has been evaluated with the *Saturne* application (EDF R&D) that is a workflow composed of a code coupling.

Second, we have carried out an extended evaluation of a resource discovery protocol (RW-OGS) previously implemented in VIGNE. This protocol is an optimization of the random walk concept designed to perform an efficient and lightweight resource discovery in the context of grid resource allocation; it uses caches and a specific dissemination mechanism. This evaluation has been performed through simulation: various parameters of the protocol were varied, and we compared it with two other protocols described in the literature.

6.2.2.2. XtreemOS Grid operating system

The scientific coordination of the XTREEMOS European project is done by Ch. Morin, assisted by O. Sanchez, Technical Manager, and S. L'Hermitte, Project Office Assistant. The objective of XTREEMOS project is to design, implement and promote a Linux-based Grid operating system providing a native virtual organization support.

In 2007, the research activities of the PARIS Project-Team were focused on the design and implementation of a fault-tolerance service offering transparent checkpointing to Grid applications, on the design of virtual organization and security services, and on the design and implementation of *LinuxSSI*, leveraging KERRIGHED SSI operating system for the cluster flavour of XTREEMOS system. Our work on *LinuxSSI* is described in Section 6.2.1.

6.2.2.2.1. Virtual organization and security services.

A key feature of XTREEMOS is its support for *Virtual Organizations* (VO). We participated in the design of the XTREEMOS approach for VO management in close collaboration with ICT, STFC and TID. The proposed approach addresses four key challenges:

- interoperability with other frameworks,
- customizable isolation,
- access control and auditing,
- scalable dynamic management of VOs

In XTREEMOS, support for VOs follows a number of design principles: single sing-on, independence of user and resource management, dynamic mapping between VO entities and Unix entities, minimized changes to Linux kernel. VO management is divided in two layers: one at VO level, one at node level. We mainly focused on the design of node-level VO management. A first prototype has been implemented by ICT and TID. We contributed to the testing and debugging of this prototype. Our future work directions include the design of advanced features such as providing strong isolation between applications executed in the framework of VOs through the use of virtualization technologies, designing mechanisms for efficient data accesses in VOs.

6.2.2.2.2. Fault tolerance for Grid application.

We designed the architecture of XTREEMOS service dealing with reliable application execution. In XTREEMOS, an application is defined as a set of application units executing on several grid nodes. An application unit is defined a set of processes running on a given node. Application checkpointing in XTREEMOS is hierarchically divided into three levels: a kernel checkpointer, a system-level checkpointer and a grid-aware check-pointer. The two former checkpointers are implemented in the XTREEMOS-F foundation

layer, while the latter is a service in XTREEMOS-G Grid services layer. The grid checkpointer is a service of the application execution manager responsible for supervision of checkpoints for an application: it applies the check- pointing strategy to all running application units. The system checkpointer is an application execution manager service that manages checkpointing for an application unit. It registers checkpointing strategies and implements them. The kernel checkpointer offers a very basic process checkpointing mechanism to save and restore the state of a process.

We implemented a first prototype of the system checkpointer. We also implemented a kernel checkpointer for Linux processes, extending the existing *BLCR mechanism* developed by Berkeley National Laboratory. BLCR was not designed to be used in the context of a Grid. The extensions we proposed make it suitable to such an environment. For example, the executable code and the libraries are included in the checkpoint in order to be able to restart a process on a Grid node that does not offer the same configuration as the initial execution node.

To provide fault tolerance for message passing applications, techniques based on rollback/recovery mechanisms are mainly used. Message logging has the advantage over coordinated checkpointing that it does not require every process of an application to rollback in the event of a single failure. We have proposed an extremely optimistic message logging protocol called *O2P*. It has been proved to tolerate multiple concurrent failures. The extremely optimistic assumption used to log message makes it more scalable than existing (moderately) optimistic message logging protocols.

To optimize execution performance in a Grid consisting of a cluster federation, message passing applications must be adapted to the hierarchical structure. We have proposed to combine the advantages of optimistic and pessimistic message logging protocols in a fault tolerance protocol for message passing applications executed in a cluster federation. Optimistic message logging optimizes performance within a cluster, whereas pessimistic message logging provides independence between clusters.

6.3. Middleware systems for computational grids

6.3.1. Parallel CORBA objects and components

Keywords: *CORBA*, *Grid*, *distributed component*, *distributed object*, *parallelism*.

Participants: Mathieu Kermarrec, Raúl López Lozano, Christian Pérez, Thierry Priol.

Distributed parallel object/component appears to be a key technology for programming distributed numerical simulation systems. It extends the well-known object/component-oriented model with a parallel execution model. Previous works such as PACO and GRIDCCM focused on communications between two parallel objects and components.

With respect to PACO++, the work carried out in 2007 was related to the improvement of PACO++, mainly bug fix issues, as well as the development of an irregular data distribution library to support a seismological code coming from the NUMASIS ANR project. Moreover, experiments have be done on NUMA machines to evaluate the behavior of PACO++ on such machines.

We have also started working on a hierarchical parallel object/component model. It is a particular but important case of the previous parallel object/component proposal. It is based on the assumption that the resource topology is hierarchical, as it turns out to be in real systems. Experiments done with a preliminary prototype on GRID'5000 show that it is possible to keep a very simple API for application developers, while being able to take advantage of the hierarchy at runtime to deliver improved performance.

Future work will mainly concern the development of such a hierarchical parallel component model and its validation with numerical applications on GRID'5000. The support of PACO++ will be continued, as it is an valuable building block.

6.3.2. Spatio-temporal software component models

Keywords: *CORBA Component Model (CCM)*, *Grid*, *software component*, *spatial description*, *temporal description*.

Participants: Julien Bigot, Hinde Lilia Bouziane, Christian Pérez, Thierry Priol.

Software component models have succeeded in handling another level of the software complexity by dealing with system architecture. However, a current limitation is that only *spatial* architectures can be handled with. Temporal description is currently handled by workflow-like models. As applications are expected to exhibit both a spatial and temporal dimension, it appears of particular interest to combine both descriptions into a coherent one.

We have started to explore how to combine both models by deriving a component model from the *GriCol* language. *GriCol* is a workflow-like language which is dual-layered: each element of the workflow may be described with respect to a data-flow model which reads and/or writes data to a global database. Next, we have defined a model of a spatio-temporal component model which is based on the concept of *task-component*. A *task-component* is a component with spatial and temporal ports. Hence, an assembly made of such components capture the two dimensions. As the model is hierarchical (a composite can be made of an assembly of component), it is possible to use both spatial and temporal composition at any level of the hierarchy.

The next steps are to define an operational semantic for such a model as the rules to determining when a component may or must be created/destroyed are not obvious. Moreover, we plan to implement such a model.

6.3.3. Application deployment on computational Grids

Participants: Landry Breuil, Boris Daix, Christine Morin, Christian Pérez, Thierry Priol.

The deployment of parallel, component-based applications is a critical issue in using computational Grids. It consists in selecting a number of nodes and in launching the application on them. We proposed a generic deployment model that aims to automatically deploy complex, static applications on Grids. The core of the model is a *Generic Application Description model* (GADe) which enables to decouple the deployment tool from a specific application description.

In 2007, we revisited the generic deployment model to be able to support dynamic resources as well as applications. We proposed a new model based on a clear separation between the description of the applications and the resources, and a model of actions on these entities. We also decided to redevelop ADAGE, a tool which implements the generic deployment model we propose. It is based on well-defined specification and thus provides a clean interface to the plug-in auxiliary sub-system in charge of the specific application description management. Currently, ADAGE is based on the static generic deployment model.

Future works are twofold. First, we will complete the dynamic generic deployment model and we will evaluate its benefits through a prototype. Second, we will continue with the development of ADAGE by adding a basic mechanism for handling dynamicity.

6.3.4. Adaptive framework for component software

Keywords: *Grid*, *components*, *fractal*, *framework*, *objects*.

Participants: Françoise André, Jérémy Buisson, Jean-Louis Pazat.

Since Grid architectures are also known to be highly dynamic, software must be able to dynamically react to the changes of the underlying execution environment. In order to help developers to create reactive software for the Grid, we are investigating a model for the adaptation of parallel components. Based on this model, we have built a generic framework for dynamic adaptation of components, called DYNACO, and a specific implementation for synchronizing parallel SPMD codes for adaptations.

Our group has worked with Ch. Pérez and H. Bouziane to integrate dynamic adaptation in the master-worker paradigm. The master-worker model defined by them (see Section 6.3.2) could benefit from the DYNACO framework to dynamically adapt to changing environments [21].

We have studied the impact of dynamic adaptation on the design of resource allocators and batch schedulers, in the context of scheduling malleable applications in multi-cluster systems [29].

6.3.5. Adaptation for data management

Participants: Françoise André, Mohamed Zouari.

The usage of context-aware data management in mobile environments has been investigated by Françoise André in collaboration with Mayté Segarra and Jean-Marie Gilliot from ENST Bretagne (Brest). A context-aware data replication and consistency system that adapts dynamically to changes in the environment has been proposed, based on the use of the DYNACO framework. This work has been supported by a contract (*ReCoDEM*) between ENST Bretagne and Orange Labs (previously known as France-Télécom R&D)

In the *ReCoDEM* project, the distributed aspects of the adaptation system has not been thoroughly investigated. Therefore, a new subject is launched since October 2007 (with M. Zouari as PhD student) to propose a generic distributed adaptation framework. This work will use data management in Grid and mobile environments as an illustrative application. Mayté Segarra from ENST Bretagne is co-adviser for the PhD thesis of M. Zouari.

6.3.6. Adaptation for fault tolerance

Participants: Jean-Louis Pazat, Xuanhua Shi.

The use of adaptive framework as been studied to build dependable applications for Grids in the context of the *SafeScale* Project. Standard cases of attacks have been simulated and taken into account using the DYNACO framework and the *MPICH-V communication library* developed at LRI. The use of such a framework for a platform for ubiquitous computing has been studied in [37].

In the future, we will connect the DYNACO framework to the *Kaapi* environment developed at IMAG/LIG in order to be able to adapt the execution of task graphs to faulty environments.

6.3.7. Dynamic load balancing

Keywords: *Load balancing*.

Participants: Jean-Louis Pazat, Nagib Abi Fadel.

Dynamic load-balancing algorithms have proven to be better than static load-balancing algorithms. However, in many cases, a single algorithm cannot be the best one with respect to the whole life of the application, especially in multi-phase applications. We are studying the dynamic adaptation of load-balancing algorithms.

We have implemented a *centralized controller* for dynamically changing a load-balancing algorithm during the execution of a program. We used the *AMPI* software and the *Charm++* library, which includes some load-balancing algorithms. The algorithm have been evaluated on the GRID'5000 platform.

In the next future, we will first study a distributed version of this algorithm.

6.4. Large-scale data management for grids

6.4.1. Using the JuxMem data-sharing service for databases

Keywords: *DSM*, *Databases*, *Grid data-sharing*, *JXTA*, *peer-to-peer*.

Participants: Gabriel Antoniu, Luc Bougé, Landry Breuil, Loïc Cudennec, Bogdan Nicolae.

Since 2003, we have been working on the concept of *data-sharing service* for Grid computing, that we defined as a compromise between two rather different kinds of data-sharing systems:

- *DSM systems*, which propose consistency models and protocols for efficient transparent management of *mutable data, on static, small-scaled configurations (tens of nodes)*;
- *P2P systems*, which have proven adequate for the management of *immutable data* on *highly dynamic, large-scale configurations (millions of nodes)*.

We illustrated this concept through the JUXMEM software platform, mainly developed by our group within the framework of Mathieu Jan's PhD thesis [70] and Sébastien Monnet's PhD thesis [79]. JUXMEM relies on the JXTA [57] generic peer-to-peer framework, which provides basic building blocks for user-defined, peer-to-peer services. L. Cudennec's PhD thesis is specifically devoted to improving the deployment of JXTA-based programs in the context of large-scale grid platforms such as GRID'5000.

In 2007, we have explored the possibility of building a distributed database management system (DBMS) on top of JUXMEM, as a natural extension of previous approaches based on the distributed shared memory paradigm. The approach we propose consists in providing the DBMS with a transparent, persistent and fault-tolerant access to the stored data, within a unstable, volatile and dynamic environment. The DBMS is thus alleviated from any concern regarding the dynamic behavior of the underlying nodes. During Abdullah Almaksour's Master internship, we performed a feasibility study, whose results were published in [20].This work has been done within the framework of the *RESPIRE* ANR project.

This work is continued within the framework of the PhD thesis of B. Nicolae, started in September 2007, with a focus on efficient storage and access to large data chunks.

6.4.2. Hierarchical Grid Storage based on the JuxMem Grid Data-Sharing Service and on the Gfarm Global File System

Keywords: *DSM*, *Databases*, *JXTA*, *grid data sharing*, *peer-to-peer*.

Participants: Gabriel Antoniu, Loïc Cudennec, Landry Breuil.

While Grid file systems provide an elegant solution for *persistent* storage of *large volumes of data* on physically distributed files, the concept of a Grid data-sharing service offers *efficient* access to globally shared data by relying on main memory storage. We claim that all these properties (large storage capacity, data persistence *and* access efficiency) are equally important and we propose a hierarchical Grid storage system that simultaneously addresses these issues.

We have defined a hybrid architecture which relies on both the JUXMEM grid-data sharing service and the *Gfarm* Grid file system (http://datafarm.apgrid.org/), and combines their specific benefits. The main idea is to allow applications to use JUXMEM's efficient memory-oriented API, while letting JUXMEM persistently store data on disk files by transparently making calls to *Gfarm* in the background. Our proposal has been validated through a prototype that couples the *Gfarm* file system with the JUXMEM data-sharing service. During Majd Ghareeb's Master internship, we performed a feasibility study. The advantages of our approach are confirmed by several experiments on the GRID'5000 testbed. A paper on this work has been submitted to an international conference.

This work has been conducted within the framework of our current bilateral collaboration with AIST/Tsukuba University, Japan. Further work will concern performance improvements through parallel communications between JUXMEM and *Gfarm*.

6.4.3. Large-scale evaluation of JXTA protocols on Grids

Keywords: *JXTA*, *Peer-to-peer*, *performance evaluation*.

Participants: Gabriel Antoniu, Mathieu Jan, Loïc Cudennec.

Features of the P2P model, such as scalability and volatility tolerance, have motivated its use in distributed systems. Several generic P2P libraries have been proposed for building distributed applications. However, very few experimental evaluations of these frameworks have been conducted, especially at large scales. In collaboration with Sun Microsystems, we have evaluated the scalability of two main protocols proposed by the *JXTA P2P platform*: the *rendezvous protocol*, whose role is to set up and maintain the JXTA P2P overlay, and the *discovery protocol*, used to find resources inside a JXTA network. We performed a detailed, large-scale, multi-site experimental evaluation of these protocols, using up to 580 nodes spread over the nine clusters of the French GRID'5000 testbed.

This work is part of our current collaboration with Sun Microsystems. It was presented at IPDPS 2007 [24].

6.4.4. Grid meta-data management based on peer-to-peer and web services

Keywords: *JXTA*, *Knowledge grid*, *meta-data management*, *peer-to-peer*.

Participant: Gabriel Antoniu.

To allow grid applications to efficiently and reliably access various heterogeneous, distributed resources, meta-data information describing the available resources plays an important role. It is therefore crucial to provide efficient meta-data management architectures and frameworks.

In collaboration with the University of Calabria (Italy), within the framework of Sébastien Monnet's post-doctoral work, we have designed of a Grid meta-data management service [19]. We focused on a particular use case: the Knowledge-Grid architecture, which provides high-level Grid services for distributed knowledge discovery applications. As meta-data is actually stored as pieces of data (e.g., XML files), they may be treated as such. We take advantage of the properties exhibited by the JUXMEM Grid data-sharing service, to transparently and reliably store and retrieve meta-data. We then build a distributed and replicated hierarchical index of available meta-data. The proposed solution lies at the border between peer-to-peer systems and Web services.

6.4.5. The Mome data-repository

Keywords: *Grids*, *scalability*.

Participant: Yvon Jégou.

Providing the data to the applications is a major issue in Grid computing. The execution of an application on some site is possible only when the data of the application are present on the "data-space" of this site. It is necessary to move the data from the sites where they are produced or located, to the execution sites. Using a *Distributed Shared Memory* (DSM) for sharing data objects has been shown to facilitate the execution of applications in distributed environments. However, traditional DSM systems have been developed for clusters of computers and target simple applications. Grid environments introduce an additional major level of complexity in data management: applications are more complex (workflows, coupled applications), more dynamic (a new application can be started dynamically and interact with an existing computation); shared data spaces are larger (several terabytes); computation resources are more heterogeneous (memory, processors), they can be grouped into clusters (cluster of clusters), they are much more dynamic (nodes can be dynamically added or removed, or can even fail), they are more numerous (thousands of nodes).

The recent developments on the MOME DSM allow to dynamically manage the DSM nodes (add and remove), to take into account the Grid interconnection structure through the hierarchical management of the nodes, and to dynamically manage the shared space of the applications using a new specific memory allocator.

6.5. Advanced programming models for the Grid

Keywords: *Chemical programming*, *autonomic systems*, *coordination*, *desktop grids*.

Participants: Jean-Pierre Banâtre, Nicolas Le Scouarnec, Thierry Priol, Yann Radenac.

In our past work, we developed the γ-calculus and *HOCL*, a Higher-Order Chemical Language based on the γ-calculus. HOCL has been used to express workflow enactment and autonomic systems. This was the subject of Yann Radenac's PhD thesis, defended and published in April 2007 [15].

This year, we have investigated how to use HOCL as a coordination language to program Desktop Grids. The aim was to express a simple Ray Tracing program and its execution within a Desktop Grid, without any central control. A distributed architecture has been designed. The implementation of a simulation was used to validate the approach. The resulting paper has been accepted for the e-Science conference in December 2007 [25].

Moreover, we have study the coordination mechanisms of HOCL and their application to Kahn's networks. This work, done in collaboration with Pascal Fradet from INRIA GRENOBLE – RHÔNE-ALPES, will be published in a special volume in memory of Gilles Kahn.

The article about programming self-organizing systems with HOCL has been published [17]. Finally, a sequential implementation of HOCL has been developed (but not released yet), and a multi-thread implementation has been started.

In the coming years, a new PhD will start working on programming web services with HOCL. Besides, a new project has been funded by the so-called *White Program* of the French ANR. This project, named *AutoChem*, aims at investigating and exploring chemical computing to program complex computing infrastructures such as Grids and real-time, deeply-embedded systems.

6.6. Experimental Grid infrastructures

Participants: Yvon Jégou, David Margery, Pascal Morillon.

The deployment of the GRID'5000 site of Rennes was initiated in November 2003. The major steps for the platforms were

Date	*# Nodes*	*# Procs*	*# Cores*	*Processor type*	*Node type*
Dec. 2003	66	132	132	Intel Xeon IA32	Dell PowerEdge 1750
Oct. 2004	33	66	66	IBM PowerPC	Apple Xserve G5
Nov. 2004	66	132	132	AMD Opteron 248	Sun V20z
Dec. 2005	102	204	204	AMD Opteron 246	HP DL145 G2
Nov. 2006	66	132	264	Intel Xeon 5148LV	Dell PowerEdge 1950
Sep. 2007	33	66	132	Intel Xeon 5148LV	Dell PowerEdge 1950

As of the end of 2007, 267 nodes corresponding to 534 processors and 732 cores are active on our platform. The following interconnection equipments have been acquired since 2003:

Date	*# Ports*	*Throughput*	*Uplink*	*Type*	*Model*
Dec. 2003	2x48	100Mb/s	1 Gb/s	Ethernet	Foundry EdgeIron
Dec. 2004	8x24	1 Gb/s	1 Gb/s	Ethernet	Cisco 3750
Dec. 2005	66	10 Gb/s		Infiniband	Mellanox/Voltaire
Feb 2006	320	1 Gb/s	2x10 Gb/s	Ethernet	Cisco 6509
Apr. 2006	33	10 Gb/s		Myrinet	Myricom
Sep. 2007	64	10 Gb/s	2x10 Gb/s	Myrinet	Myricom

As of the end of 2007, the production network interconnects all nodes at 1 Gb/s using Ethernet technology, and provides connectivity to GRID'5000 sites through a 10 Gb/s optical link. A private Ethernet network, the management network interconnecting all nodes, is used for node management: monitoring, reboot, etc. It is exploited by the management software of the platform (*OAR*, *kadeploy*). Two local high-performance networks are available: an Infiniband network interconnecting 66 nodes at 10 Gb/s and a Myrinet 10G network interconnecting 97 nodes at 10 Gb/s.

The statistics show an average platform usage higher than 70%. The results provided by local users, mainly from the PARIS Project-Team, show that experimentations on our platform are cited in 6 PHD thesis, 6 book chapters or journal articles, 30 communications to international conferences and in 11 communications to national conferences.

7. Contracts and Grants with Industry

7.1. EDF Contract 1

Participants: Christine Morin, Emmanuel Jeanvoine.

Program: The collaboration with EDF R&D aims at designing, implementing and evaluating a resource discovery and allocation service for a cluster federation.

Starting time: October 1, 2004

Ending time: September 30, 2007

Partners: EDF R&D, INRIA

Support: EDF R&D funding, CIFRE PhD Grant (E. Jeanvoine)

Project contribution: The work carried out by the PARIS Project-Team relates to the design and implementation of the VIGNE Grid-aware system for grids. As part of this contract, we design a distributed information system and a distributed application life-cycle management service, based on an underlying peer-to-peer overlay network. It enables to cope with the decentralized and dynamic nature of a large-scale grid.

In 2007, we evaluated by simulation a resource discovery protocol relying on an unstructured overlay network and based on optimized random work algorithms. We also proposed an approach for improving the discovery of rare resources. Finally, we integrated specific functionalities into VIGNE to support workflow applications relying on an external workflow engine. VIGNE has been experimentally validated in the framework of the *SALOME* platform for numerical simulation http://www.salome-platform.org/ with an unmodified legacy workflow application provided by EDF R&D.

7.2. EDF Contract 2

Participants: Boris Daix, Christine Morin, Christian Pérez.

Program: The collaboration with EDF R&D aims at improving the dynamic deployment of scientific code-coupling applications on cluster federations, taking into account their execution constraints.

Starting time: January 1, 2006

Ending time: December 31, 2008

Partners: EDF R&D, INRIA

Support: EDF R&D funding, CIFRE PhD Grant (B. Daix)

Project contribution: The work carried out by the PARIS Project-Team relates to the dynamic deployment of coupled, parallel scientific applications on federations of clusters, taking into account their execution constraints. In 2007, we worked on the design of a deployment model for applications and ressources that both have properties of parallelism/distribution, heterogeneity, and dynamicity.

7.3. Sun Microsystems

Participants: Gabriel Antoniu, Luc Bougé, Loïc Cudennec, Thierry Priol.

Starting time: October, 2005

Ending time: September, 2008

Partners: Sun Microsystems,INRIA

Support: Sun funding, PhD grant (*Loïc Cudennec*)

Project contribution: The work addresses techniques to optimize the use of the JXTA P2P library on Grid infrastructures. In January 2007, Gabriel Antoniu and Loïc Cudennec visited the JXTA team in Santa Clara. Main achievements in 2007: paper presented at the IPDPS 2007 conference; release of the new ADAGE generic deployment tool and its dedicated plug-in to deploy and monitor the execution of JXTA-C-based applications; proposal of a load-balancing algorithm for the management of the future version of JXTA's overlay.

8. Other Grants and Activities

8.1. Regional grants

8.1.1. Brittany Council

8.1.1.1. PhD grants

Participants: Loïc Cudennec, Mohamed Zouari.

The Brittany Regional Council provides half of the financial support for the PhD theses of Loïc Cudennec (starting on October 1, 2005, for 3 years) and Mohamed Zouari (starting on October 1, 2007, for 3 years). This support amounts to a total of 28,000 Euros/year.

8.1.1.2. 5000NET Project

Participants: Yvon Jégou, David Margery, Pascal Morillon.

The 5000NET Project is funded by the Brittany Regional Council until July 2007. Its aim was to provide financial support for the integration of high-speed interconnection networking equipments in our GRID'5000 platform.

8.1.1.3. Support to XtreemOS Project Management

Participants: Sandrine L'Hermitte, Christine Morin.

The Brittany Regional Council provides a financial support for the management of the XTREEMOS IP project. This supports amounts to a total of 30,000 Euros. It contributes to funding S. L'Hermitte, who assists the scientific coordinator and ensures the clerical management of the XTREEMOS project office and of all XTREEMOS management bodies.

8.2. National grants

8.2.1. ANR WP: ANR White Program

8.2.1.1. ANR WP AutoChem Project

Participants: Jean-Pierre Banâtre, Alexandru Popovici, Thierry Priol.

The *AutoChem* Project of the ANR WP gathers 4 partners: the *PARIS* Project-Team from INRIA RENNES – BRETAGNE ATLANTIQUE, the *POP-ART* Project-Team from INRIA GRENOBLE – RHÔNE-ALPES, the University of Évry and the Atomic Energy Agency (CEA). This project aims at investigating and exploring an unconventional approach, based on chemical computing, to program complex computing infrastructures, such as Grids and real-time deeply-embedded systems. It is a 3-year project which started in December 2007.

8.2.2. ANR CI: ANR Program on High-Performance Computing and Simulation

8.2.2.1. ANR CI DISC Project

Participants: Raúl López Lozano, Christian Pérez, Thierry Priol.

The *DISC* Project of the ANR CI gathers 7 partners: 6 academic research teams – the *CAIMAN*, *SMASH* and *OASIS* Project-Teams from INRIA SOPHIA-ANTIPOLIS – MÉDITERRANÉE, the PARIS Project-Team from INRIA RENNES – BRETAGNE ATLANTIQUE, the *MOAIS* Project-Team from INRIA GRENOBLE – RHÔNE-ALPES and Laboratory ID-IMAG, and the *Distributed Systems and Objects* Team from LaBRI, and one industrial partner – EADS CCR.

It aims at studying and promoting a new paradigm for programming non-embarrassingly parallel scientific computing applications on distributed, heterogeneous, computing platforms. The *DISC* project concentrates its activities on numerical kernels and related issues that are of interest to a large variety of application contexts. The emphasis is put on designing parallel numerical algorithms and programming simulation software that efficiently exploit a computational grid and more particularly, the GRID'5000 testbed.

It is a 3-year project which started in January 2006. Project site: http://www-sop.inria.fr/caiman/personnel/Stephane.Lanteri/discogrid/cigc_disc.html.

8.2.2.2. ANR CI LEGO Project

Participants: Gabriel Antoniu, Landry Breuil, Hinde Lilia Bouziane, Loïc Cudennec, Christian Pérez.

The *LEGO* Project of the ANR CI gathers 6 partners: LIP – INRIA Project-Team *GRAAL*; IRISA– INRIA Project-Team PARIS; LaBRI – INRIA Project-Team *Runtime*; the IRIT Laboratory in Toulouse; and the *CRAL*, Center of Astronomical Research of Lyon.

The aim of this project is to provide algorithmic and software solutions for large-scale architectures, focusing on performance issues. The software component approach provides a flexible programming model where resource management issues and performance optimizations are handled by the implementation. On the other hand, the current component technology does not provide adequate data-management facilities, needed for large data in widely distributed platforms, and it does not efficiently deal with dynamic behaviors. The project addresses topics in programming models, communication models, and scheduling. The results are validated on three applications: an ocean-atmosphere numerical simulation, a cosmology simulation, and a sparse-matrix solver.

It is a 3-year project which started in January 2006. Project site: http://graal.ens-lyon.fr/LEGO/.

8.2.2.3. ANR CI NUMASIS Project

Participants: Christian Pérez, Mathieu Kermarrec.

The NUMASIS Project of the ANR CI gathers 8 partners: two industrial companies – BULL (Echirolles) and Total (Pau), two EPIC institutions – BRGM (Orléans) and CEA (Bruyères-le-Châtel), and 4 academic laboratories – ID-IMAG (INRIA Projects-Teams *Mescal* and *Moais*), LaBRI (INRIA projects-Teams *Runtime* and *Scalapplix*), LMA (INRIA Project-Team *Magique 3D*) and IRISA (INRIA Project-Team PARIS).

It deals with recent NUMA multiprocessor machines with a deep hierarchy. In order to efficiently exploit it, the project aims at evaluating the features of current systems, at proposing and implementing new mechanisms for process, data and communication management. The target applications come from the seismology field that appear representative of current needs in scientific computing.

It is a 3-year project which started in January 2006. Project site: http://numasis.gforge.inria.fr/.

8.2.3. ANR MD: ANR Program on Data Masses and Ambient Knowledge

8.2.3.1. ANR MD RESPIRE Project

Participants: Gabriel Antoniu, Luc Bougé, Landry Breuil, Loïc Cudennec.

The RESPIRE Project of the ANR MD program aims at providing a peer-to-peer (P2P) environment for advanced data management applications. It started in January 2006 and gathers research teams from the "databases" area and from the "distributed systems" area, in order to take advantage from their respective background, to have a more global view of the problem and to raise synergy. The RESPIRE Project is based on the JXTA infrastructure which provides a complete abstraction from the underlying P2P network organization (DHT, flooding, super-peer). RESPIRE services are divided into basic services (peer management, communication management, group subscribing, notification, data storage and key-based retrieval) and advanced services, which rely upon basic services for data access (querying), logical clustering, collaborative work and distributed query evaluation. Part of the basic services will be provided by the JXTA infrastructure. The main actions that will be developed in the project are resource access and sharing, managing logical cluster, handling replication and automated deployment of the environment. During Abdullah Almaksour's Master internship, we performed a feasibility study, whose results were published in [20].

The project started in January 2006 for 3 years. Gabriel Antoniu is the local correspondent of RESPIRE for the PARIS Project-Team. Project site: http://respire.lip6.fr/.

8.2.4. ANR SI: ANR Program on Security and Informatics

8.2.4.1. ANR SI SafeScale

Participants: Jean-Louis Pazat, Françoise André, Jérémy Buisson, Nagib Abi Fadel, Xuanhua Shi.

The *SafeScale* Project is concerned with security and safety in global ambient computing systems, e.g., computational grids. Partners of this project are LIPN (Coordinator, Paris), ID-IMAG (Grenoble), ENSTB (Brest) and LMC-IMAG (Grenoble).

We have used our adaptive techniques (e.g., DYNACO) to implement application reactions to use-case attacks on an experiment on GRID'5000. Next year we will connect DYNACO to the *Kaapi* task execution environment to study adaptation with work-stealing.

8.2.5. ACI GRID: Incentive Co-Ordinated Program for Fundamental Research on Grids

8.2.5.1. ACI GRID Grid'5000 Project

Participants: Yvon Jégou, David Margery, Pascal Morillon.

The ACI GRID GRID'5000 Project, terminated in July 2007, provided financial support for the integration of high-performance networking on the GRID'5000 platform in Rennes.

8.2.6. ANR TLOG: ANR Program on Software Technologies

8.2.6.1. ANR TLOG NeuroLog Project

Participant: Yvon Jégou.

The *NeuroLog* consortium (*Software technologies for integration of process, data and knowledge in medical imaging*) is targeting software technologies in medical domains for large scale management of data, knowledge and computation: management and access of partly structured data, heterogeneous and distributed in an open environment; access control and protection of private medical data; control of workflows implied in complex computing process on grid infrastructures; extraction and quantification of relevant parameters for different pathologies.

8.3. European grants

8.3.1. CoreGRID NoE Project

Participants: Françoise André, Gabriel Antoniu, Hinde Lilia Bouziane, Jérémy Buisson, Päivi Palosaari, Christian Pérez, Thierry Priol, Olivia Vasselin.

Thierry Priol is the Scientific Coordinator of a *Network of Excellence* proposal, called COREGRID, in the area of Grid and Peer-to-Peer (P2P). The COREGRID network started on September 1, 2004. As many as 41 partners, mostly from 18 European countries are involved. The COREGRID Network of Excellence aims at building a European-wide research laboratory that will achieve scientific and technological excellence in the domain of large-scale distributed, Grid, and Peer-to-Peer computing. The primary objective of the COREGRID Network of Excellence is to build solid foundations for Grid and Peer-to-Peer computing both on a methodological basis and a technological basis. This will be achieved by structuring research in the area, leading to integrated research among experts from the relevant fields, more specifically distributed systems and middleware, programming models, knowledge discovery, intelligent tools, and environments.

The research programme is structured around six complementary research areas, i.e., work packages that have been selected on the basis of their strategic importance, their research challenges, and the European expertise in these areas to develop next generation Grids: *Knowledge and Data Management*, *Programming Models*, *Architectural Issues: Scalability, Dependability, Adaptability*, *Grid information, Resource and Workflow Monitoring Services*, *Resource Management and Scheduling*, *Grid Systems, Tools and Environments*

INRIA is managing the network in collaboration with the ERCIM office. ERCIM is in charge of administrative and financial management. Th. Priol is the *Scientific Coordinator* (SCO), leading the network with respect to the scientific aspects, and looking after its overall management. He is assisted by Olivia Vasselin who took over Päivi Palosaari in June 2007. The main tasks of the SCO during this third year were coordinating and monitoring the activities related to the scientific and technical workpackages, coordinating the COREGRID *Scientific Advisory Board*, performing the first ranking of partners activity, coordinating the preparation of the second *Joint Program of Activities* and providing the first internal assessment of the network. In addition, the SCO participated in dissemination tasks by giving presentations, contributing to the COREGRID Newsletters, etc.

Christian Pérez is responsible for the COREGRID contract within INRIA. He is responsible for managing the four INRIA Project-Teams (PARIS, *Grand-Large*, *OASIS* and *SARDES*) with respect to periodic reporting, etc. His main tasks were to represent INRIA in the COREGRID Members General Assembly meetings and votes.

8.3.2. EchoGRID Specific Support Action

Participants: Jean-Pierre Banâtre, Thierry Priol, Yann Radennac.

Th. Priol is the *Scientific Coordinator* (SCO) of the ECHOGRID Specific Support Action that is funded under the FP6 IST Work Programme. This action aims to foster collaboration in Grid research and technologies by defining short-, mid-, and long-term vision in the field. It is a 2-year project which started in February 2007. It involves 10 partners from 4 European countries plus China. Th. Priol participated to one workshop and one conference organized in Beijing, respectively in February and November 2007. In the context of this action, Yann Radennac has been awarded a 12-month post-doc grant starting october 2007 to work at ICT from the Chinese Academia of Science. His research are devoted to advanced programming models for Grids.

8.3.3. XtreemOS IP Project

Participants: Matthieu Fertré, Jérôme Gallard, Yvon Jégou, Sylvain Jeuland, Adrien Lèbre, Pascal Le Métayer, Sandrine L'Hermitte, David Margery, Christine Morin, Thierry Priol, Thomas Ropars, Oscar Sanchez.

Ch. Morin is the *Scientific Coordinator* (SCO) of the XTREEMOS Integrated Project (IP) that addresses Strategic Objective 2.5.4 *Advanced Grid Technologies, Systems and Services*, Focus 3 on *Network-centric Grid Operating Systems* as described in the IST 2006 Work Programme.

The XTREEMOS projects aims at the design, implementation, evaluation and distribution of an open source Grid operating system with native support for virtual organizations and capable of running on a wide range of underlying platforms, from clusters to mobiles. The approach we propose in this project is to investigate the construction of a new Grid OS, XTREEMOS, based on the existing general-purpose OS Linux [5].

It is a 4-year project which started in June 2006. It involves 19 partners from 7 European countries plus China. The XTREEMOS consortium composition is a balance between academic and industrial partners interested in designing and implementing the XTREEMOS components (Linux extensions to support VOs and Grid OS services), packaging and distributing the XTREEMOS system on various hardware platforms, promoting and providing user support for the XTREEMOS system, and experimenting with Grid applications using the XTREEMOS system. Various end-users are involved in XTREEMOS project, providing a large variety of test cases in scientific and business computing domains.

INRIA is managing the project in collaboration with the *Caisse des Dépôts et Consignations* (CDC). CDC is in charge of administrative and financial management, while Ch. Morin as a scientific coordinator is leading the project with respect to the scientific and technical aspects. The XTREEMOS Project Office was established at the beginning of the project, involving an Administrative Assistant, S. L'Hermitte, and a Technical Manager, O. Sanchez. The main tasks of the Project Office in 2007 are coordinating and monitoring the project activities, providing the clerical support for XTREEMOS management bodies: Governing Board, Executive Committee, Scientific Advisory Committee, IPUDC, organizing meetings of the management bodies, general technical meetings and the project review. In addition, the Project Office participated in dissemination and communication tasks by delivering presentations, creating and maintaining the XTREEMOS internal and external web-sites (http://www.xtreemos.eu/), making posters and flyers, and editing the XTREEMOS electronic newsletter.

J.-P. Banâtre is the INRIA representative at the *Governing Board*. Th. Priol is a member of the *Scientific Advisory Committee*.

Y. Jégou leads the WP4.3 Work-Package, aiming at setting up XTREEMOS testbeds. The GRID'5000 experimental grid platform will be used as a testbed by XTREEMOS partners. Ch. Morin leads WP1.1, Project management, WP2.1, Virtual Organization support in Linux, WP2.2 Federation management and WP5.3, Collaboration with other IST Grid-related projects.

9. Dissemination

9.1. Community animation

9.1.1. Leaderships, Steering Committees and community service

European COREGRID IST-FP6 Network of Excellence. Th. Priol is the *Scientific Coordinator* of the COREGRID Network of Excellence (http://www.coregrid.net/). This network started on September 2004, for a duration of 4 years. Ch. Pérez is the INRIA Scientific Correspondent of the COREGRID NoE.

European ECHOGRID IST-FP6 Supported Action. Th. Priol is the *Scientific Coordinator* of the ECHOGRID project (http://echogrid.ercim.org/). This project started on February 2007, for a duration of two years.

European XTREEMOS IST-FP6 Integrated Project. Ch. Morin is the *Scientific Coordinator* of the XTREEMOS Integrated Project (http://www.xtreemos.eu/). This integrated project started on June 2006, for a duration of four years. Y. Jégou is a member of XTREEMOS Executive Committee. Th. Priol is a member of XTREEMOS Scientific Advisory Committee. J.-P. Banâtre is the INRIA representative in the XTREEMOS Governing Board.

ACI GRID Program, Ministry of Research. Th. Priol was the Director of the ACI GRID Program, funded by the French National Ministry of Research till July 2007. The ACI GRID was the national French initiative in the area of Grid computing.

Euro-Par Conference Series. L. Bougé serves as the Vice-Chair of the *Steering Committee* of the *Euro-Par* annual conference series on parallel computing (250–300 attendees, http://www.europar.org/). In 2007, the Euro-Par conference was organized in Rennes. Th. Priol and L. Bougé have served as Vice-Chairs of the Program Committee. G. Antoniu has served as Publicity Chair.

ANR CI NUMASIS Project. Ch. Pérez is the local correspondent of the NUMASIS Project (*Adaptation et optimisation des performances applicatives sur architectures NUMA. Étude et mise en oeuvre sur des applications en SISmologie*). This 3-year project started in January 2006 (http://numasis.gforge.inria.fr/).

ANR CI DISC Project. Ch. Pérez is the local correspondent of the DISC Project (*Distributed objects and components for high performance scientific computing on the* GRID'5000 *test-bed*). This 3-year project started in January 2006 (http://www-sop.inria.fr/caiman/personnel/Stephane.Lanteri/discogrid/).

ANR MD RESPIRE Project. G. Antoniu is the local correspondent of the RESPIRE Project (*Peer-to-peer resources and services, querying and replication*). This 3-year project started in January 2006 (http://respire.lip6.fr/).

ANR CI LEGO Project. G. Antoniu is the local correspondent of the LEGO Project (*League for Efficient Grid Operation*). This 3-year project started in January 2006 (http://graal.ens-lyon.fr/LEGO/).

ANR SI SAFESCALE Project. J.-L. Pazat is the local correspondent of the SAFESCALE Project (*Security And Fault-tolerance to Exploit Safety ambient Computing in lArge scaLe Environments*). This 3-year project started in January 2006 (https://www-lipn.univ-paris13.fr/safescale/).

CNRS, GDR ASR. J.-L. Pazat is co-director of the GSP working group on Grids, Systems and Parallelism of the CNRS Research Co-operative Federation (*Groupement de recherche*, GDR) ASR (*Architectures, Systems and Networks*). F. André serves as the coordinator of the ADAPT action (*Dynamic Adaptation*) of the GSP working group.

Agrégation of Mathematics. L. Bougé serves as one of the Vice-Chairs of the National Selection Committee for High-School Mathematics Teachers (*Agrégation de mathématiques*). He is in charge of the newly-founded *Fundamental Computer Science* track of the selection process.

9.1.2. Editorial boards, direction of program committees

L. Bougé is a member of the *Editorial Advisory Board* of the *Scientific Programming* Journal, IOS Press.

A. Lèbre organized a workshop (attended by 25 participants) on *HPC File Systems: From Cluster to Grids* in the framework of the French SIGOPS Chapter *Journées thèmes émergents*, Rennes, France, October 2007.

Ch. Morin served as the Local Chair of Topic 1 on *Support Tools and Environments* at the *Euro-Par 2007* Conference, Rennes, France, August 2007.

J.-L. Pazat serves as the Chair of the Organizing Committee of the RenPar, CFSE and Sympa federated conference series. He is the chairman of the Steering Committee of RenPar (*Rencontres francophone du parallélisme*, http://www.renpar.org/). The next edition of RenPar will be held in Fribourg in February 2008.

Th. Priol is a member of the Editorial Board of the *Parallel Computing* Journal.

He is a member of the Editorial Board of the *International Journal of Web Services Research*.

He was Co-Chair of the Program Committee of the *2007 CoreGRID Symposium*, Rennes, France, August 2007.

He is the Chair of the Program Committee of the *2008 CCGRID conference*, Lyon, France, May 2008.

9.1.3. Program Committees

G. Antoniu served in the Program Committees for the following conferences:

- PDP 2007: *16th Euromicro Conference on Parallel Distributed and network-based Processing*, Naples, Italy, February 2007.
- CCGrid 2007: *IEEE/ACM International Symposium on Cluster Computing and the Grid*, Rio de Janeiro, May 2007.
- MSOP2P 2008 *2nd International Workshop on Modeling, Simulation, and Optimization of Peer-to-peer Environments*. In conjunction with PDP 2008, Toulouse, February 2008.
- CCGrid 2008: *IEEE/ACM International Symposium on Cluster Computing and the Grid*, Lyon, May 2008.
- HIPS 2008: *International Workshop on High-Level Parallel Programming Models and Supportive Environments*, Miami, Florida, USA, April 2008.
- HPDGrid 2008: *International Workshop on High-Performance Data Management*, Toulouse, France, June 2008, in conjunction with VecPar 2008.
- DaMap 2008: *International Workshop on Data Management in P2P systems*. In conjunction with EDBT 2008 (*International Conference on Extending Database Technology*), March 2008, Nantes, France.

L. Bougé served in the Program Committees for the following conferences:

- NPC 2007: *IFIP International Conference on Network and Parallel Computing*, Dalian, China, September 2008.

NPC 2008: *IFIP International Conference on Network and Parallel Computing*, Shangai, China, September 2008.

Ch. Morin served in the Program Committees of the following conferences:

TopModel 2007: Workshop on Tools, Operating Systems and Programming Models to Develop Reliable Systems (TOPMoDelS), in conjunction with IPDPS 2007, (Long Beach), California, USA, March 2007.

HPCVirt 2007: *1st Workshop on System-level Virtualization for High Performance Computing*. In conjunction with EuroSys 2007. Lisbon, Portugal, March 2007.

SDMAS 2007: International Workshop on Scalable Data Management Applications and Systems (SDMAS). In conjunction with the 2007 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), Las Vegas (Nevada), USA, June 2007.

ICDCS 2007: *International Conference on Distributed Computing Systems*, Toronto, Canada, June 2007.

Euro-Par 2007: *The 13th International Euro-Par Conference European Conference on Parallel and Distributed Computing*, Rennes, France, August 2007.

Cluster 2007: *IEEE Cluster 2007*, Austin, Texas, September 2007.

E-Science 2007: *Third IEEE International Conference on e-Science and Grid Computing*, Bangalore, India, December 2007.

RenPar 18: *18e Rencontres francophones du parallélisme*, Fribourg, Switzerland, February 2008.

ICA3PP 2008 *The 8th International Conference on Algorithms and Architectures for Parallel Processing*, Cyprus, June 2008.

J.-L. Pazat served in the Program Committees of the following conferences

GPC 2007: *International Conference on Grid and Pervasive Computing*, Paris, France, May 2007.

RenPar 18: *18e Rencontres francophones du parallélisme*, Fribourg, Switzerland, February 2008.

Ch. Pérez served in the Program Committees of the following conferences:

AINA 2007: *The IEEE 21st IEEE International Conference on Advanced Information Networking and Applications*, Niagara Falls, Canada, May 2007.

PMGC 2007: *Workshop on Programming Models for Grid Computing*, Rio de Janeiro, Brazil, May 2007.

ICNS 2007: *The Third International Conference on Networking and Services*, Athens, Greece, June 2007.

ATC 2007: *4th International Conference on Autonomic and Trusted Computing*, Hong Kong, July 2007.

Euro-Par 2007: *The 13th International Euro-Par Conference European Conference on Parallel and Distributed Computing*, Rennes, France, August 2007.

CoreGRID Symposium 2007: *CoreGRID Symposium*, Rennes, France, August 2007.

Cluster 2007: *IEEE Cluster 2007*, Austin, Texas, September 2007.

EuroPVM/MPI 2007: *The 14th European PVM/MPI Users' Group Meeting*, Paris, France, October 2007.

HPC-GECO/CompFrame 2007: *Joint Workshop on: HPC Grid Programming Environments and Components; and Component and Framework Technology in High-Performance and Scientific Computing*. In conjunction with ooPSLA 2007, Montreal, Canada, October 2007.

Middleware 2007: *The ACM/IFIP/USENIX 8th International Middleware Conference*, Newport Beach, California, USA, November 2008.

RenPar 18: 18e Rencontres francophones du parallélisme, Fribourg, Switzerland, February 2008.

ICNS 2008: *The Fourth International Conference on Networking and Services*, Gosier, Guadeloupe, March 2008.

AINA 2008: *The IEEE 22nd International Conference on Advanced Information Networking and Applications*, GinoWan, Okinawa, Japan, March 2008.

CCGrid 2008: *8th IEEE International Symposium on Cluster Computing and the Grid*, Lyon, France, May 2008.

Th. Priol served in the Program Committees of the following conferences:

CCGRID 2007: *IEEE International Symposium on Cluster Computing and the Grid*, Rio de Janeiro, May 2007.

GADA 2007: *International Conference on Grid computing, high-PerformAnce and Distributed Applications*. Vilamoura, Algarve, Portugal, November 2007.

German e-Science 2007: *German e-Science Conference*, Baden Baden, Germany, May 2007.

HPC-GECO/CompFrame 2007: *Joint Workshop on: HPC Grid Programming Environments and Components; and Component and Framework Technology in High-Performance and Scientific Computing*. In conjunction with ooPSLA 2007, Montreal, Canada, October 2007.

HPCVirt 2007: *1st Workshop on System-level Virtualization for High Performance Computing*. In conjunction with EuroSys 2007, Lisbon, Portugal, March 2007.

HPDC 2007: *16th IEEE International Symposium on High-Performance Distributed Computing*, Monterey, USA, June 2007.

ICWS 2007: *IEEE International Conference on Web Services*, Salt Lake City, USA, July 2007.

ICCS 2007: *International Conference on Computational Science*, Beijing, China, May 2007.

WI 2007: *2007 IEEE/WIC/ACM International Conference on Web Intelligence*, Silicon Valley, USA, November 2007.

9.1.4. Evaluation committees, consulting

L. Bougé has been registered as an academic expert for the AERES, the *French National Agency for Evaluation for Research and Academic institutions*.

He served as a member of the Selection Committee for the ASTI PhD Award 2007 (http://www.asti.asso.fr/, Federation of French associations for Information Sciences and Techniques).

He was solicited as a referee for the Île-de-France Region *Digiteo* Research Program on Software and Complex Systems.

He was solicited as a member of the INRIA FUTURS/Saclay Selection Committee for Junior Researchers (CR2).

He was solicited as a member of the Annual Selection Committee for the *Research and Doctoral Supervision Award* of the French Ministry of Research (PEDR, *Prime d'encadrement doctoral et de recherche*).

He served as an external referee for the Habilitation Thesis of Yves Denneulin, INRIA GRENOBLE – RHÔNE-ALPES.

Ch. Morin acted as a referee for the Foreign PhD Committee of Martin Kacer from CTU, Prague, Czech Republic.

She acted as a referee for the Foreign PhD Committee of Gladys Utrera Iglesias, UPC, Barcelona, Spain.

She acted as a referee for the Foreign PhD Committee of Andrew Maloney, Deakin University (Australie).

Th. Priol was a member of the Scientific Committee of the ANR CI Program on *High Performance Computing and Simulation* of the French National Research Agency.

He was a reviewer for the "Starting Grants" of the EU European Research council. He acted as an evaluator of the EU e-Infrastructure unit.

He is member of an International Committee appointed by the *Fundação para a Ciência e a Tecnologia*, Portugal, to evaluate the research units in Electrical Engineering and Computer Science (EECS) in Portugal in 2007-2008.

We was a reviewer for the Austrian Science Fund (Austria) and the Faculté Polytechnique de Mons (Belgium)

9.2. Academic teaching

Only the teaching contributions of project-team members on non-teaching positions are mentioned below.

G. Antoniu is teaching part of the *Operating Systems* Module at *IUP 2 MIAGE*, IFSIC. He has given lectures on peer-to-peer systems within the *High Performance Computing on Clusters and Grids* Module and within the *Peer-to-Peer Systems* Module of the Master Program, UNIVERSITY RENNES 1, and within the *Distributed Systems* Module taught for the final year engineering students of INSA Rennes.

B. Daix gave lectures on *GNU/Linux specialized for visually-impaired students in scientific domain* at INSA, Lyon, July 2007.

A. Lèbre gave lectures on high-performance I/O in clusters within the *Distributed Systems: from networks to Grids* Module of the Master Program, UNIVERSITY RENNES 1.

Ch. Morin is responsible for a graduate teaching Module *Distributed Systems: from networks to Grids* of the Master Program in Computer Science, UNIVERSITY RENNES 1. Within this module, she gave lectures on cluster and Grid computing.

She gave a lecture on cluster single system image operating systems within the *Parallelism* Module of the 3rd-year students of INT of Évry.

Ch. Pérez gave lectures to 5th-year students of INSA of Rennes on CORBA and CCM within the course *Objects and components for distributed programming*.

He also gave lectures to 5th-year students of Polytech Nantes on CORBA and CCM within the course *Objects and components for distributed programming*.

Th. Priol gave lectures on Distributed Shared Memory and Grid Programming within the *Distributed Systems: from Network to Grids* Module of the Master Program, UNIVERSITY RENNES 1.

9.3. Conferences, seminars, and invitations

Only the events not listed elsewhere are listed below.

B. Daix gave a talk entitled *Déploiement automatique d'applications sur les plates-formes d'exécution dans le contexte HPC* at *Journées des doctorants SINETICS* (JDS), EDF R&D, Clamart, October 2007.

E. Jeanvoine gave a talk entitled *Vigne : un système d'exploitation pour simplifier l'usage des grilles*, Cosinus seminar, EDF R&D Clamart, France, March 2007.

A. Lèbre gave a talk entitled *kDFS Overview: Current State and Main Objectives* at the French SIGOPS chapter's *Journées thèmes émergents* on *HPC File Systems: From Cluster to Grids*, Rennes, France, October 2007.

Ch. Morin gave a talk entitled *Needs and Plans Concerning Kerrighed in the XtreemOS Project* at the First Kerrighed Summit, Paris, February 2007.

She was invited to give a talk entitled *XtreemOS: a Linux-based Grid Operating System Providing Native Virtual Organization Support* at the First International Workshop on Global Computing, Sibiu, Romania, April 2007.

She gave a talk entitled *XtreemOS: an Operating System for Next Generation Grids* at 10th anniversary of IrisaTech club, Rennes, France, June 2007.

She gave an invited talk entitled *XtreemOS: a Grid Operating System providing a native Support to Virtual Organizations* at the NorduGrid Conference, Copenhaguen, Danemark,September 2007.

She presented a talk on *XtreemOS: a Grid OS based on Linux* at the Grid Operating Systems Community BOF session at SC'07, Reno, USA, November 2007.

She was invited to participate to a panel at the *Sciences et techniques : un avenir pour filles et garçons* colloquium organized by the *Femmes et Sciences* association, Paris, France, November 2007.

J. Parpaillon gave a talk entitled *Oscar Experience Feedback and actions proposal*, OSCAR meeting, ORNL, Oak Ridge, USA, January 2007.

He gave a talk entitled *Using GIT for* KERRIGHED at the first KERRIGHED Summit, Paris, February 2007.

He gave a talk entitled *OSCAR Roadmap and New Package Architecture* at the OSCAR BOF held in conjunction with HPCS, Saskatoon, Canada, May 2007.

He was invited to give a talk entitled *Administrer une grappe de calcul* at the *Journées systèmes : gestion des serveurs de calcul*, Lyon, France, September 2007.

Ch. Pérez gave a talk on *Defining, Implementing, Executing and Deploying a High Performance Component Model* at the France Télécom research seminar on *Grid Computing: research challenges for a Telco operator*, Issy-les-Moulineaux, February 6th 2007.

He gave a talk on *Extending Software Component Port Model to Simplify Application Development* at the Ames Laboratory Seminar, Ames, October 29th 2007.

He gave a talk on *Extending Software Component Port Model to Simplify Application Development* at the Ames Laboratory Seminar, Ames, October 29th 2007.

Th. Priol gave a keynote presentation on *the CoreGRID Component Model* at the International Conference on Grid and Pervasive Computing, Paris, May, 2007.

9.4. Administrative responsibilities

F. André is the vice-chair of the Administrative Committee of IFSIC, the Computer Science Teaching Department of UNIVERSITY RENNES 1.

L. Bougé chairs the Computer Science and Telecommunication Department (*Département Informatique et Télécommunications, DIT*) of the Brittany Extension of ENS CACHAN on the Ker Lann Campus in Bruz, in the close suburb of Rennes.

He leads the Master Program in Computer Science at the Brittany Extension of ENS CACHAN (*Magistère Informatique et Télécommunications*, for short, the famous MIT Rennes :-)). This program is co-supported with UNIVERSITY RENNES 1. It was launched in September 2002. Olivier Ridoux, LANDE Project-Team, IRISA, co-supervises the program for UNIVERSITY RENNES 1.

He serves as the Vice-Chairman of the Selection Committee (*Commission de spécialistes d'Établissement*, CSE) for Computer Science at ENS CACHAN, and as an external deputy-member of the Computer Science CSE at UNIVERSITY RENNES 1.

J.-L. Pazat leads the Master Program of the 5th year of Computer Science at INSA of Rennes.

He is responsible for a teaching module on Parallel Processing for engineers at INSA of Rennes. Within this module, he gave lectures on parallel and distributed programming. He is responsible for a graduate teaching module Objects and components for distributed programming for 5th-year students of INSA of Rennes. He gave lectures on parallelism in operating systems in a module for graduate students

9.5. Miscellaneous

F. André is a member of the Selection Committee (Commission de spécialistes, CSE) of IFSIC (Computer Science department of University of Rennes1), of the Computer Science department of INSA of Rennes and of the Computer Science group of University of Rennes 2.

L. Bougé is a member of the Project-Team Committee of IRISA (*Comité des projets*), standing for the ENS CACHAN partner.

Ch. Morin has served since May 2007 as an external deputy member in the Selection Committee (*Commission de spécialistes*, CSE) for the Computer Science department of INSA Rennes.

Ch. Morin was a member of the 2007 Selection Committee for the Junior Researcher permanent positions (CR2, CR1) at INRIA RENNES – BRETAGNE ATLANTIQUE.

J.-L. Pazat is a member of the Computer Science Department committee. He is the local coordinator for the international exchange of students at the computer science department of INSA. He serves as the Chairman of the Selection Committee (*Commission de spécialistes d'Établissement*, CSE) for Computer Science at INSA Rennes

Ch. Pérez is a member of the IRISA Laboratory Committee (*Conseil de laboratoire*).

10. Bibliography

Major publications by the team in recent years

[1] M. ALDINUCCI, F. ANDRÉ, J. BUISSON, S. CAMPA, M. COPPOLA, M. DANELUTTO, C. ZOCCOLO. *Parallel program/component adaptivity management*, in "ParCo 2005, Málaga, Spain", 13-16 September 2005, http://www.irisa.fr/paris/Biblio/Papers/Buisson/AldAndBuiCamCopDanZoc05PARCO.pdf.

[2] F. ANDRÉ, M. LE FUR, Y. MAHÉO, J.-L. PAZAT. *The Pandore Data Parallel Compiler and its Portable Runtime*, in "High-Performance Computing and Networking (HPCN Europe 1995), Milan, Italy", Lecture Notes in Computer Science, vol. 919, Springer Verlag, May 1995, p. 176–183.

[3] G. ANTONIU, L. BOUGÉ. *DSM-PM2: A portable implementation platform for multithreaded DSM consistency protocols*, in "Proc. 6th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS '01), San Francisco", Lect. Notes in Comp. Science, Available as INRIA Research Report RR-4108, vol. 2026, Springer-Verlag, Held in conjunction with IPDPS 2001. IEEE TCPP, April 2001, p. 55–70, http://hal.inria.fr/inria-00072523.

[4] J.-P. BANÂTRE, D. LE MÉTAYER. *Programming by Multiset Transformation*, in "Communications of the ACM", vol. 36, n° 1, January 1993, p. 98–111.

[5] CONSORTIUM, XTREEMOS. *Annex 1 - Description of Work*, XtreemOS Integrated Project, IST-033576, April 2006, Contract funded by the European Commission.

[6] A. DENIS, C. PÉREZ, T. PRIOL. *PadicoTM: An Open Integration Framework for Communication Middleware and Runtimes*, in "IEEE Intl. Symposium on Cluster Computing and the Grid (CCGrid2002), Berlin, Germany", Available as INRIA Reserach Report RR-4554, IEEE Computer Society, May 2002, p. 144–151, http://hal.inria.fr/inria-00072034.

[7] A.-M. KERMARREC, C. MORIN, M. BANÂTRE. *Design, Implementation and Evaluation of ICARE*, in "Software Practice and Experience", n° 9, 1998, p. 981–1010.

[8] T. KIELMANN, P. HATCHER, L. BOUGÉ, H. BAL. *Enabling Java for High-Performance Computing: Exploiting Distributed Shared Memory and Remote Method Invocation*, in "Communications of the ACM", Special issue on Java for High Performance Computing, vol. 44, n° 10, October 2001, p. 110–117.

[9] Z. LAHJOMRI, T. PRIOL. *KOAN: A Shared Virtual Memory for iPSC/2 Hypercube*, in "Proc. of the 2nd Joint Int'l Conf. on Vector and Parallel Processing (CONPAR'92)", Lecture Notes in Computer Science, vol. 634, Springer Verlag, September 1992, p. 441–452, http://hal.inria.fr/inria-00074927.

[10] T. PRIOL. *Efficient support of MPI-based parallel codes within a CORBA-based software infrastructure*, in "Response to the Aggregated Computing RFI from the OMG, Document orbos/99-07-10", July 1999.

Year Publications

Books and Monographs

[11] S. GORLATCH, M. BUBAK, T. PRIOL (editors). *Achievements in European Research on Grid Systems*, CoreGRID Books, Springer, November 2007.

[12] A.-M. KERMARREC, L. BOUGÉ, T. PRIOL (editors). *Proceedings of the Euro-Par 2007 Parallel Processing*, Lecture Notes in Computer Science, vol. 4641, Springer, INRIA, August 2007.

[13] T. PRIOL, M. VANNESCHI (editors). *Towards Next Generation Grids*, CoreGRID books, Springer, INRIA, August 2007.

Doctoral dissertations and Habilitation theses

[14] E. JEANVOINE. *Intergiciel pour l'exécution efficace et fiable d'applications distribuées dans des grilles dynamiques de très grande taille*, In French, Thèse de doctorat, Université de Rennes 1, IRISA, Rennes, France, November 2007.

[15] Y. RADENAC. *Programmation "chimique" d'ordre supérieur*, Thèse de doctorat, Université de Rennes 1, April 2007.

Articles in refereed journals and book chapters

[16] G. ANTONIU, H. L. BOUZIANE, M. JAN, C. PÉREZ, T. PRIOL. *Combining data sharing with the master-worker paradigm in the common component architecture*, in "Cluster Computing", vol. 10, n° 3, 2007, p. 265 – 276.

[17] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Programming Self-Organizing Systems with the Higher-Order Chemical Language*, in "International Journal of Unconventional Computing", vol. 3, n^{o} 3, 2007, p. 161-177.

[18] E. JEANVOINE, L. RILLING, C. MORIN, D. LEPRINCE. *Using Overlay Networks to Build Operating System Services for Large Scale Grids*, in "Scalable Computing: Practice and Experience", Special issue on Practical Aspects of Large-Scale Distributed Computing. Selected paper and extended version of ISPDC '06., vol. 8, n^{o} 3, 2007, p. 229–239, http://www.scpe.org/vols/vol08/no3/SCPE_8_3_01.pdf.

Publications in Conferences and Workshops

[19] M. ALDINUCCI, F. ANDRÉ, J. BUISSON, S. CAMPA, M. COPPOLA, M. DANELUTTO, C. ZOCCOLO. *An Abstract Schema Modelling Adaptivity Management*, in "Integrated Research in GRID Computing", S. GORLATCH, M. DANELUTTO (editors), proceedings of the CoreGRID Integration Workshop 2005, Springer, 2007, http://www.irisa.fr/paris/Biblio/Papers/Buisson/AldAndBuiCamCopDanZoc06CGIW.pdf.

[20] A. ALMOUSA ALMAKSOUR, G. ANTONIU, L. BOUGÉ, L. CUDENNEC, S. GANÇARSKI. *Building a DBMS on top of the JuxMem Grid Data-Sharing Service*, in "Proc. HiPerGRID Workshop, Brasov, Romania", Held in conjunction with Parallel Architectures and Compilation Techniques 2007 (PACT2007)., September 2007, http://hal.inria.fr/inria-00178655/en/.

[21] F. ANDRÉ, H. L. BOUZIANE, J. BUISSON, J.-L. PAZAT, C. PÉREZ. *Towards dynamic adaptability support for the master-worker paradigm in component based applications*, in "CoreGRID Symposium in conjunction with Euro-Par 2007 conference, Rennes, France", 27-28 August 2007.

[22] G. ANTONIU, E. CARON, F. DESPREZ, A. FÈVRE, M. JAN. *Towards a Transparent Data Access Model for the GridRPC Paradigm*, in "Proc. of the 13th Internationl Conference on High Performance Computing (HiPC 2007), Goa, India", Lect. Notes in Comp. Science, To appear., Springer-Verlag, December 2007.

[23] G. ANTONIU, A. CONGIUSTA, S. MONNET, D. TALIA, P. TRUNFIO. *Peer-to-Peer Metadata Management for Knowledge Discovery Applications in Grids*, in "CoreGRID Workshop on Grid Middleware, Dresden, Germany", To appear, available as CoreGRID technical report TR0083., Held in conjunction with the Internationnal Supercomputing Conference (ISC 2007), June 2007, http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0083.pdf.

[24] G. ANTONIU, L. CUDENNEC, M. DUIGOU, M. JAN. *Performance scalability of the JXTA P2P framework*, in "Proc. 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007), Long Beach, CA, USA", March 2007, 108, http://hal.inria.fr/inria-00119916/en.

[25] J.-P. BANÂTRE, N. LE SCOUARNEC, T. PRIOL, Y. RADENAC. *Towards "Chemical" Desktop Grids*, in "Proceedings of the 3rd IEEE International Conference on e-Science and Grid Computing, Bangalore, India", December 2007.

[26] J. BIGOT, C. PÉREZ. *Enabling Collective Communications between Components*, in "CompFrame '07: Proceedings of the 2007 symposium on Component and framework technology in high-performance and scientific computing, New York, NY, USA", ACM Press, 21-22 October 2007, p. 121–130, https://www.irisa.fr/paris/bibadmin/./uploads/pdf/Enabling_collective_communications_between_components.pdf.

[27] H. L. BOUZIANE, C. PÉREZ, N. CURRLE-LINDE, M. RESCH. *Analysis of Component Model Extensions to Support the GriCoL Language*, in "CoreGRID Workshop on Grid Programming Model, Grid and P2P Systems Architecture, Grid Systems, Tools and Environments, Heraklion, Crete, Greece", June 2007, p. 36-45.

[28] J. BUISSON, F. ANDRÉ, J.-L. PAZAT. *Supporting Adaptable Applications in Grid Resource Management Systems*, in "8th IEEE/ACM International Conference on Grid Computing", 19-21 September 2007.

[29] J. BUISSON, O. SONMEZ, H. MOHAMED, W. LAMMERS, D. EPEMA. *Scheduling Malleable Applications in Multicluster Systems*, in "IEEE International Cluster Conference, Austin, USA", 17-20 September 2007.

[30] E. JEANVOINE, C. MORIN, D. LEPRINCE. *Vigne: Executing Easily and Efficiently a Wide Range of Distributed Applications in Grids*, in "Proceedings of Euro-Par 2007, Rennes, France", 2007, p. 394–403.

[31] R. L. LOZANO, C. PÉREZ. *Improving MPI Support for Applications on Hierarchically Distributed Resources*, in "Recent Advances in Parallel Virtual Machine and Message Passing Interface, 14th European PVM/MPI User's Group Meeting, Paris, France", Lecture Notes in Computer Science, Springer Berlin, September 2007, p. 187-194.

[32] C. MORIN. *XtreemOS: A Grid Operating System Making your Computer Ready for Participating in Virtual Organizations*, in "ISORC '07: Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, Washington, DC, USA", IEEE Computer Society, 2007, p. 393–402, http://dx.doi.org/10.1109/ISORC.2007.62.

[33] J. PARPAILLON. *OSCAR KernelPicker: Handling Clients Kernels*, in "21st International Symposium on High Performance Computing Systems and Applications (HPCS '07)", J. PARPAILLON (editor), 2007.

[34] L. RILLING, S. SIVASUBRAMANIAN, G. PIERRE. *High Availability and Scalability Support for Web Applications*, in "Proceedings of the IEEE International Symposium on Applications and the Internet", January 2007.

[35] T. ROPARS, E. JEANVOINE, C. MORIN. *GAMoSe: An Accurate Monitoring Service for Grid Applications*, in "6th International Symposium on Parallel and Distributed Computing (ISPDC 2007), Hagenberg, Austria", July 2007, p. 295–302.

[36] T. ROPARS. *Combining Optimism and Pessimism in a Grid Message Logging Protocol*, in "Student Forum of International Conference on Dependable Systems and Networks (DSN 2007) (Supplemental Volume), Edinburgh, UK", June 2007.

[37] X. SHI, J.-L. PAZAT. *A Novel Adaptive and Safe Framework for Ubicomp*, in "Proceedings of the 2007 International Workshop on Service, Security and its Data management for Ubiquitous Computing (SSDU-07)", May 2007.

[38] E. Y. YANG, B. MATTHEWS, A. LAKHANI, Y. JÉGOU, C. MORIN, O. D. SÁNCHEZ, C. FRANKE, P. ROBINSON, A. HOHL, B. SCHEUERMANN, D. VLADUSIC, H. YU, A. QIN, R. LEE, E. FOCHT, M. COPPOLA. *Virtual Organization Management in XtreemOS: an Overview*, in "CoreGRID Symposium", Springer Verlag, August 2007.

Internal Reports

[39] J. GALLARD, A. LÈBRE, C. MORIN, P. GALLARD, G. VALLÉE. *Is Virtualization Killing Single System Image Research?*, Technical report, n° RR-6389, INRIA, November 2007, http://hal.inria.fr/inria-00196717.

[40] S. LANTÉRI, R. L. LOZANO, C. PÉREZ. *Une extension MPI pour les ressources distribuées hiérarchiques*, Technical report, n° RT-0336, INRIA, 2007, http://hal.inria.fr/inria-00150543.

[41] A. LÈBRE, R. LOTTIAUX, C. MORIN. *Reducing Kernel Development Complexity in Distributed Environments*, To appear, Technical report, n° 6405, INRIA, 2008, http://hal.inria.fr/inria-00201911.

[42] T. ROPARS, C. MORIN. *O2P: An Extremely Optimistic Message Logging Protocol*, Research Report, n° 6357, IRISA/Paris Research group, Université de Rennes 1, INRIA, 2007, https://hal.inria.fr/inria-00187682.

Miscellaneous

[43] N. AUPETIT. *Amélioration de la plate-forme de test Kerrighed*, Technical report, ENST - Bretagne, September 2007.

[44] J. BIGOT. *Composants logiciels parallèles et communications collectives*, Technical report, Université de Rennes 1, 2007.

[45] CONSORTIUM, XTREEMOS. *Design and implementation of basic application unit checkpoint/restart mechanisms in Linux*, 2007, XtreemOS deliverable.

[46] CONSORTIUM, XTREEMOS. *Design and implementation of basic reconfiguration mechanisms*, November 2007, Deliverable D2.2.4.

[47] CONSORTIUM, XTREEMOS. *Design and implementation of basic reconfiguration mechanisms in LinuxSSI*, 2007, XtreemOS deliverable.

[48] CONSORTIUM, XTREEMOS. *Design and implementation of high performance disk input-output operations in a federation*, 2007, XtreemOS deliverable.

[49] CONSORTIUM, XTREEMOS. *Design of a basic Linux version for mobile devices*, 2007, XtreemOS deliverable.

[50] CONSORTIUM, XTREEMOS. *Prototype of the basic version of Linux-XOS*, 2007, XtreemOS deliverable.

[51] CONSORTIUM, XTREEMOS. *Prototype of the basic version of LinuxSSI*, 2007, XtreemOS deliverable.

[52] CONSORTIUM, XTREEMOS. *Study of XtreemOS testbed extension*, 2007, XtreemOS deliverable.

[53] J. GALLARD. *Ordonnancement de tâches dans des grappes de calculateurs.*, Technical report, IFSIC - Université de Rennes 1, June 2007, http://www.irisa.fr/paris/bibadmin/uploads/pdf/Gal07Master.pdf.

[54] Y. JÉGOU. *ACI GRID - Programme GRID'5000, Projet Grid'5000-Rennes, Rapport de fin de contrat*, 2007, Rapport de fin de contrat.

[55] Y. JÉGOU. *Grid'5000 - pôle Bretagne Rapport d'activité 2003-2006*, 2007, Rapport de fin de contrat.

References in notes

[56] I. FOSTER, C. KESSELMAN (editors). *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, 1998.

[57] *Project JXTA: Java programmers guide*, Sun Microsystems, Inc., 2001, http://www.jxta.org/white_papers.html.

[58] *OpenMP Fortran Application Program Interface*, Version 2.0, November 2000.

[59] *Wireless Application Protocol 2.0: technical white paper*, January 2002, http://www.wapforum.org/what/WAPWhite_Paper1.pdf.

[60] R. ARMSTRONG, D. GANNON, A. GEIST, K. KEAHEY, S. KOHN, L. MCINNES, S. PARKER, B. SMOLINSKI. *Toward a Common Component Architecture for High-Performance Scientific Computing*, in "Proceeding of the 8th IEEE International Symposium on High Performance Distributed Computation", August 1999.

[61] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Principles of Chemical Programming*, in "Proceedings of the 5th International Workshop on Rule-Based Programming (RULE 2004)", S. ABDENNADHER, C. RINGEISSEN (editors), ENTCS, vol. 124, n^{o} 1, Elsevier, June 2005, p. 133–147.

[62] J.-P. BANÂTRE, P. FRADET, D. LE MÉTAYER. *Gamma and the Chemical Reaction Model: Fifteen Years After*, in "Multiset Processing", LNCS, vol. 2235, Springer-Verlag, 2001, p. 17–44.

[63] J.-P. BANÂTRE, D. LE MÉTAYER. *A new computational model and its discipline of programming*, Technical report, n^{o} RR0566, INRIA, September 1986, http://hal.inria.fr/inria-00075988.

[64] J.-P. BANÂTRE, D. LE MÉTAYER. *Programming by Multiset Transformation*, in "Communications of the ACM", vol. 36, n^{o} 1, January 1993, p. 98–111.

[65] G. BERRY, G. BOUDOL. *The Chemical Abstract Machine*, in "Theoretical Computer Science", vol. 96, 1992, p. 217–248.

[66] D. CHEFROUR, F. ANDRÉ. *Auto-adaptation de composants ACEEL coopérants*, in "3e Conférence française sur les systèmes d'exploitation (CFSE 3)", 2003.

[67] A. GEIST, A. BEGUELIN, J. DONGARRA, W. JIANG, R. MANCHEK, V. SUNDERAM. *PVM 3 Users Guide and Reference manual*, Oak Ridge National Laboratory, Oak Ridge, TN, USA, May 1994.

[68] K. GHARACHORLOO, D. LENOSKI, J. LAUDON, P. GIBBONS, A. GUPTA, J. HENESSY. *Memory Consistency and event ordering in scalable shared memory multiprocessors*, in "17th Annual Intl. Symposium on Computer Architectures (ISCA)", ACM, May 1990, p. 15–26.

[69] J. GRAY, D. SIEWIOREK. *High Availability Computer Systems*, in "IEEE Computer", September 1991.

[70] M. JAN. *JuxMem : un service de partage transparent de données pour grilles de calculs fondé sur une approche pair-à-pair*, Thèse de doctorat, Université de Rennes 1, IRISA, Rennes, France, November 2006.

[71] E. JEANNOT, B. KNUTSSON, M. BJORKMANN. *Adaptive Online Data Compression*, in "IEEE High Performance Distributed Computing (HPDC 11)", 2002.

[72] P. KELEHER, A. COX, W. ZWAENEPOEL. *Lazy Release Consistency for Software Distributed Shared Memory*, in "19th Intl. Symposium on Computer Architecture", May 1992, p. 13–21.

[73] P. KELEHER, D. DWARKADAS, A. COX, W. ZWAENEPOEL. *TreadMarks: Distributed Shared Memory on standard workstations and operating systems*, in "Proc. 1994 Winter Usenix Conference", January 1994, p. 115–131.

[74] L. LAMPORT. *How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs*, in "IEEE Transactions on Computers", vol. 28, n° 9, September 1979, p. 690–691.

[75] P. LEE, T. ANDERSON. *Fault Tolerance: Principles and Practice*, vol. 3 of Dependable Computing and Fault-Tolerant Systems, Springer Verlag, second revised edition, 1990.

[76] F. MATTERN. *Virtual Time and Global States in Distributed Systems*, in "Proc. Int. Workshop on Parallel and Distributed Algorithms, Gers, France", North-Holland, 1989, p. 215–226.

[77] MESSAGE PASSING INTERFACE FORUM. *MPI: A Message Passing Interface Standard*, Technical report, University of Tennessee, Knoxville, TN, USA, 1994.

[78] D. S. MILOJICIC, V. KALOGERAKI, R. LUKOSE, K. NAGARAJA, J. PRUYNE, B. RICHARD, S. ROLLINS, Z. XU. *Peer-to-Peer Computing*, Submitted to Computing Surveys, Research Report, n° HPL-2002-57, HP Labs, March 2002, http://www.hpl.hp.com/techreports/2002/HPL-2002-57R1.pdf.

[79] S. MONNET. *Gestion des données dans les grilles de calcul : support pour la tolérance aux fautes et la cohérence des données*, Thèse de doctorat, Université de Rennes 1, IRISA, Rennes, France, November 2006.

[80] OMG. *CORBA Component Model V3.0*, June 2002, OMG Document formal/2002-06-65.

[81] G. PĂUN. *Computing with Membranes*, in "Journal of Computer and System Sciences", vol. 61, n° 1, 2000, p. 108-143.

[82] D. RIDGE, D. BECKER, P. MERKEY, T. STERLING. *Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs*, in "IEEE Aerospace Conference", 1997.

[83] C. SZYPERSKI. *Component Software - Beyond Object-Oriented Programming*, Addison-Wesley / ACM Press, 1998.