INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team PARIS

Programming Parallel and Distributed Systems for Large Scale Numerical Simulation Applications

Rennes

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 Extension of* ENS CACHAN. *Since, the project activity is jointly supervised by a ad-hoc Committee on an annual basis. Regarding 2004, this committee met at* IRISA *on July 5, jointly with the similar committee for the DistribCom Project-Team, a newly established joint project-team between* IRISA *and the Brittany Extension of* ENS CACHAN.

Head of project-team
Thierry Priol [DR INRIA]

Administrative assistant
Maryse Auffray [TR INRIA]

Staff member Inria
Gabriel Antoniu [CR]
Yvon Jégou [CR]
Anne-Marie Kermarrec [DR (since February 2004)]
Christine Morin [DR]
Christian Pérez [CR]
David Margery [IR (since September 2004)]

Staff member University of Rennes 1
Jean-Pierre Banâtre [Professor]
Françoise André [Professor (since October 2004)]

Staff member Insa de Rennes
Jean-Louis Pazat [Professor]

Staff member Ens Cachan
Luc Bougé [Professor, ENS CACHAN Brittany Extension]

Project technical staff
Renaud Lottiaux [INRIA, DGA *COCA* Contract]
David Margery [INRIA, DGA *COCA* Contract (till September 2004)]
Guillaume Mornet [INRIA, RNTL CASPER Contract (till April 2004), PRIR Brittany Regional Council (since May 2004)]

PhD student
Lilia Hinde Bouziane [INRIA Grant (since October 2004)]
Alexandre Denis [MENRT Grant, ENS Lyon (till September 2004)]
Jérémy Buisson [MENRT Grant]
Pascal Gallard [INRIA Grant]
Mathieu Jan [INRIA-Regional Council Grant]
Emmanuel Jeanvoine [Cifre EDF industrial Grant (since October 2004)]
Sébastien Lacour [INRIA Grant]
Sébastien Monnet [MENRT Grant]
Yann Radenac [MENRT Grant)]
André Ribes [INRIA-Regional Council Grant]
Etienne Rivière [MENRT Grant (since October 2004)]
Louis Rilling [MENRT Grant, ENS CACHAN]
Gaël Utard [INRIA Grant]
Geoffroy Vallée [Cifre EDF industrial Grant (till February 2004)]

Post-doctoral fellow
Zsolt Nemeth [ERCIM Grant]
Geoffroy Vallée [INRIA PDI (from March 2004)]

Long-term visitor

Isaac Scherson [University of California (March-July 2004)]

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 enables the speed-up of the design of complex manufactured products, such as cars or aircrafts, thanks to numerical simulation techniques. As computer performance increases rapidly, 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 simulation 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 clusters of clusters (aka *Grids*), will have to be used simultaneously 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 six themes: *Operating System and Runtime for Clusters*, *Middleware for Computational Grids*, *P2P System Foundations*, *Large-scale Data Management for Grids*, *Advanced Models for the Grid* and *Experimental Grid Infrastructures*. These research activities encompass both basic research, seeking conceptual advances, and applied research, to validate the proposed concepts against real applications. The project-team is also involved in setting-up 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 only by researchers, 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, 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 SMEs. It also generates new needs and demands for more accurate numerical simulation. But 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. 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 needed 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 their 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

2.1.3. 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 six directions.

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

Middleware 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 mix both parallel and distributed processing seamlessly.

P2P System Foundations. The peer-to-peer communication paradigm has recently become a natural candidate to tackle the scalability requirements of recent distributed systems. More specifically, many conventional distributed protocols and algorithms need to be revisited according to this fully decentralized model.

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

Advanced models for the Grid. This theme 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 Infrastructure. The challenge here is to be able to design and to build an instrument (in the sense of a scientific instrument) for computer scientists involved in grid research. Such instrument has to be highly reconfigurable and scalable to several thousand of resources.

2.2. Operating system and runtime for clusters

Clusters, made up of homogeneous computers interconnected via high performance networks, are now the most widely used general, high-performance computing platforms for scientific computing. While such an architecture is attractive with respect to price/performance there still exists a great potential for efficiency improvements at the software level. System software requires improvements 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, each with resources locally managed.

Several attempts to build an SSI have been made at the middleware level as Beowulf [82], PVM [69] or MPI [78]. However, these environments provide only a partial SSI. Our approach in 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) and kernel resources (process, memory pages, data streams, files) 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 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 application. This is independent of the underlying communication paradigm. Even though backward error recovery (BER) has extensively been studied from the theoretical point of view, efficiently implement BER protocols transparently to the applications is yet to be solved. There are very few implementations of recovery 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 different 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 as the basic communication mechanism within the runtime. We are essentially interested in Fortran and its OpenMP extensions [63]. 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 few support for grid-oriented programming model.

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 relevant solution. However, they need to be tailored for scientific applications, in particular with respect of the encapsulation of parallel codes into objects or components, the communications between "parallel" objects or components, the required runtime support, the deployment and the 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 code, which should take into account the parallelism of resources. It is thus required to efficiently integrate both worlds into a coherent one.

The second issue concerns the simplicity and the scalability of communications 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.

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 *simultaneously* active within a same process. Third, solutions to the two previous issues should support high performance constraints to be accepted by users.

The deployment of applications is another issue. Not only is it important to constrain the deployment by specifying the requirements in term of the computational resource (GFlops/s, amount of memory, etc.), but it is also crucial to specify the constraints related to communication resources such as the amount of bandwidth or the latency between computational resources.

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

2.4. P2P System Foundations

The past decade has been dominated by a major shift in scalability requirements of distributed systems and applications mainly due to the exponential growth of the Internet. A standard distributed system today is related to thousand even millions of computing entities scattered all over the world and deals with a huge amount of data. Conventional distributed algorithms designed in the context of local area networks do not scale to such extreme configurations and have to be revisited to fit into this new challenging setting. The peer-to-peer communication paradigm is now the prevalent model to cope with the requirements of large scale distributed systems. Peer-to-peer systems rely on a symmetric communication model where peers are potentially both client and servers. They are fully decentralized, thus avoiding the bottleneck imposed by the presence of servers in traditional systems. They are highly resilient to peers arrivals and departures. Finally, individual peer behavior is based on a local knowledge of the system and yet the system converges toward global properties.

peer-to-peer systems pose many interesting research challenges. The first area is related to the way peers are logically connected on top of IP to form an overlay network. Two main classes of such overlays dominate, structured and unstructured. The differences relate to the choice of the neighbors in the overlay and to the presence of an underlying naming structure. Second, a large number of functionalities may be defined on such overlays related to the localization, search, routing, etc. Finally, peer-to-peer overlays networks are an attractive support *(i)* to solve the scalability issues of traditional distributed applications and *(ii)* to define new challenging cooperative applications. Our objective in that area is to focus on the foundations of such systems and to define new structures and algorithms that could be used in a large number of emerging distributed applications.

2.5. 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 computation. 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 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, a few constraints need to be addressed. The clusters which make up the grid are not guaranteed to remain constantly available. 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.

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 large scales and can serve as inspiration source.

Finally, in grid applications, data is generally shared 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.

2.6. Advanced models for the Grid

Till now, research activities related to the Grid have been focused on the design and implementation of middleware and tools to experiment grid infrastructure with applications. Very few attention has been paid to programming models suitable for such widely computing infrastructures. Programming of such infrastructures is still 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 [3].

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, 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.

2.7. Experimental Grid Infrastructures

The PARIS Project-Team is engaged in research along six themes: *Operating System and Runtime for Clusters*, *Middleware for Computational Grids*, *P2P System Foundations*, *Large-scale Data Management for Grids*, *Advanced Models for the Grid* and *Experimental Grid Infrastructures*. The concepts proposed by each of these themes must be validated against real applications on realistic hardware. The project-team manages a computation platform dedicated to operating system and middleware experimentations and is involved in setting-up a national grid computing infrastructure, GRID 5000, enabling such large-scale experiments.

Our experimental platform is dedicated to operating system and middleware experimentations: it is possible the repeat experiments in the same environment (same machines, same network, etc.). The allocation of our resources to the experiments is handled through *GridPrems*, a shared resource manager developed in our

group. Our experimental platform is heterogeneous: PowerPC and PC families of processors, 32-bit and 64-bit architectures, Linux and Mac OS X operating system. Heterogeneity allows realistic validation of interoperability of middleware and P2P systems. Our platform is composed of sufficiently large groups of homogeneous computation nodes: 66 dual Xeon, 66 dual Opterons, 33 Xserve G5. This enables to evaluate the scalability of operating systems, runtimes and applications on various architectures.

The integration of our platform in the GRID 5000 national grid computing infrastructure enables large-scale experiments. The GRID 5000 infrastructure federates experimental platforms (currently six platforms) across France. These platform are connected through Renater using dedicated Gigabit Ethernet links.

3. Scientific Foundations

3.1. Introduction

Research activities within the PARIS Project-Team encompass 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 physical access only to its local memory. Implementation 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 written operation performed by another processor previously. Within a distributed or parallel a system, the notion of the *last* memory access is sometimes undefined since there is no global clock that gives 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* [75] 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 appear 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 run parallel programs correctly, 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* [70] has been thoroughly studied since it is well adapted to programming parallel scientific applications. The principle behind Release Consistency that memory accesses are (should?) always be guarded by synchronization operations (locks, barriers, etc.), so that the shared memory system only needs to be consistent 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 have been proposed of Release Consistency [73]: 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 two alternative implementations differ in the number of messages that needs to be sent/received, and in the complexity of the implementation [74]. Implementations of Release Consistency rely on the use of a logical clock such as a vector clock [77]. 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 practically impossible. 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* [71] 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* [76] 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 is 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 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 a 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. Localization and routing

Localization and routing are core functionalities of large-scale distributed systems. Localization is related to the ability of finding items in a system and routing to the ability to reach any destination from any source. Recent research on emerging *peer-to-peer* (P2P) systems [81] has focused on designing adequate localization and routing strategies for large-scale, highly-decentralized environments. The proposed algorithms have the properties, that address the main requirements of such environments: high scalability, fault tolerance (with respect to node or link failures), no (or very little) dependence on centralized entities.

The first fully distributed approach to localization, illustrated by *Gnutella*, relies on flooding. A second generation of P2P systems (e.g., *KaZaA*) have introduced the notion of super-peer: localization is flooding-based between the super-peers, which serve as local directories for groups of regular peers. However, flooding strategies have one main weakness: since they generate a lot of traffic, a limit has to be set on the number of times queries are re-propagated. As a result, queries for data may fail, whereas the data are actually stored in the system.

In order to provide both high fault tolerance and the guarantee to always reach data available in the network, recent research has focused on localization schemes based on *Distributed Hash Tables* (DHT). This promising approach is illustrated by *Chord* (MIT), *Pastry* (Microsoft Research) and *Tapestry* (UC Berkeley) and has also been used for the latest major version (2.0) of the *JXTA* generic environment for P2P services started by Sun Microsystems.

3.5. High-performance communication

High-performance communication [67] is uttermost crucial for parallel computing. However, it is less important in distributed computing, whereas interoperability is more important. The quest for high-performance communication has led to the development of specific hardware technologies along the years: *SCI*, *Myrinet-1*, *VIA*, *Myrinet-2000*, *InfiniBand*, etc. A dedicated low-level communication library is often required to fully benefit from the hardware specific feature: *GM* or *BIP* for *Myrinet*, *SISCI* for *SCI*, etc. To face the diversity of low level communication libraries, research has focused on generic high-performance environments such as *Active Message* (Univ. of Berkeley), *Fast Message* (Univ. of Illinois), MADELEINE (LaBRI, Bordeaux), *Panda/Ibis* (Univ. of Amsterdam) and *Nexus* (Globus Toolkit). Such generic environments are usually *not* assumed to be directly used by a programmer. Higher-level communication environments are specifically designed: PVM, MPI or software DSM such as *TreadMarks* are such examples in the field of parallel computing. While high performance communication research has mainly focused on system-area networks, the emergence of grid computing enlarges its focus to wide-area networks and, more specifically, to *high-bandwidth, wide-area networks*. Research is needed to efficiently utilize such networks. Some examples are adaptive dynamic compression algorithms, and parallel stream communication.

Previous work [66] has shown that high-performance communication not only requires an adequate communication library, but also demands some cooperation with the *thread scheduler*. It is particular important as more and more middleware systems as well as applications are multithreaded. Another related issue, which deserves further research, is to minimize network reactivity without generating too much overhead.

3.6. Distributed resource 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* [61] 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 [79]. The peer-to-peer communication model relies on a symmetric relationship between peers which may act both as client and server. 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*. Another major issue is related to efficient search algorithms in large scale distributed systems. Such algorithms represent the core of many resource management systems. Whereas some P2P systems have proven efficient to deal with exact queries (DHT), efficient keyword-based or range queries P2P approaches are yet to be designed.

3.7. Component model

Software component technology [84] 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 [80], 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 CCA Forum [65] 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.

3.8. 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 these constraints 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 adaptation 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 an adaptation, 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.

Adaptation 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 [64]. A more general way is to provide mechanisms enabling dynamic self-adaptation 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 [72] 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 [68] that provides generic mechanisms for the adaptation process and for the definition of the adaptation rules is needed.

4. Application Domains

Keywords: *Scientific computing*, *cooperative applications*, *coupling of numerical codes*.

The project-team research activities address in priority 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 and 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.

If scientific computing is our primary target to apply the results gained by the project-team, we do not exclude other kind of applications such as multimedia or discrete-event distributed applications for which our research can be applied. More specifically, phone companies represent a target for our research on peer-to-peer computing. The basic idea is to replace global servers dealing with registration and connection of sources and destinations of a phone call, by end-users proprietary cooperatives entities.

5. Software

5.1. Kerrighed

Keywords: *Cluster operating system*, *checkpointing*, *cooperative caching*, *distributed file system*, *distributed shared memory*, *global scheduling*, *high availability*, *process migration*, *remote paging*, *single system image*.

Participants: Pascal Gallard, Renaud Lottiaux, David Margery, Christine Morin, Louis Rilling, Gaël Utard, Geoffroy Vallée.

Contact: Christine Morin

URL: http://www.kerrighed.org/

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

License: GNU General Public License version 2. Kerrighed is a registered trademark.

Presentation: KERRIGHED (formerly known as *Gobelins*) 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 all the SSI properties. Global resource management makes distribution of resources, throughout the cluster nodes, transparent 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 to have to restart applications from the beginning when node failure happens. 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 may be executed without modification on top of KERRIGHED and further optimized if needed.

KERRIGHED is not an entirely new operating system developed from scratch. 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 of 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|) is available to provide a support to KERRIGHED users.

Current status: KERRIGHED includes 90,000 lines of code (mostly in C). It represents 170 person-months of effort. The development of KERRIGHED started in late 1999. The stable release of KERRIGHED is Version V1.0 (December 2004) based on Linux 2.4.24. It provides a customizable cluster wide process scheduler, a cluster wide Unix process interface, high performance stream migration allowing migration of MPI processes, process checkpointing and an efficient distributed file system. It also offers a complete *Pthread* support, allowing to execute legacy OpenMP and multithreaded applications on a cluster without any recompilations.KERRIGHED SSI features are customizable.

Several demonstrations of KERRIGHED have been presented this year at *Linux Expo* (Paris, February 2004, Renaud Lottiaux and Christine Morin), *Les matinales de Rennes Atalante* (Rennes, May 2004, Renaud Lottiaux, David Margery and Christine Morin), *Supercomputing 2004 Conference* (Pittsburgh (USA), November 2004, Pascal Gallard, Renaud Lottiaux, Christine Morin and Geoffroy Vallée). KERRIGHED is currently experimented by *Cap Gemini*, *ONERA CERT*, DGA *CELAR* in the framework of COCA contract, as well as by EDF *R&D*, Ulm University (Germany) and ORNL (USA). More than 250 external downloads of KERRIGHED have been recorded in 2004.

5.2. PadicoTM

Keywords: *Grid*, *communication framework*, *middleware system*.

Participants: Alexandre Denis, Christian Pérez, Thierry Priol.

Contact: Christian Pérez

URL: http://www.irisa.fr/paris/Padicotm/

Status: Registered at APP, under Ref. IDDN.FR.001.260013.000.S.P.2002.000.10000.

License: GNU General Public License version 2.

Presentation: PADICOTM is an open integration framework for communication middleware and runtimes. It enables several middleware systems (such as CORBA, MPI, SOAP, etc.) to be used at the same time. It provides an efficient and transparent access to all available networks with the appropriate method.

PADICOTM is composed of a core, which provides a high-performance framework for networking and multi-threading, and services, plugged into the core. High-performance communications and threads are obtained thanks to MARCEL and MADELEINE, provided by PM 2 . The PADICOTM core aims at making the different services running at the same time run in a cooperative way rather than competitive.

An extended set of commands is provided with PADICOTM to ease the compilation of its modules (|padico-cc|, |padico-c++|, etc.). In particular, a very useful one aims at hiding the differences between different CORBA implementation. The first version was called *Ugo* (available in the 0.1.x Series). It has since been replaced by *myCORBA*.

PadicoControl is a JAVA application that helps to control the deployment of PADICOTM application. It allows a user to select the deployment node and to perform individual or collective operation like loading or running a PADICOTM module.

PadicoModule (still under development) is a JAVA application which assists the low-level administration of a PADICOTM installation. It allows to check module dependency, to modify module attributes, etc. It can work on local file system as well as through a network thanks to a SOAP daemon being part of the service.

A public mailing list (|padico-users@listes.irisa.fr|) is available to support users of PADICOTM.

Current status: The development of PADICOTM has started end of 2000. It represents 86 person-month effort.

The stable release of PADICOTM is Version 0.1.5 (November 2002). The unstable version (CVS version) is 0.3.0beta1.

The stable version (0.1.x series) includes the PADICOTM core, PadicoControl, Ugo and external software: a PADICOTM-enabled version of *omniORB* (3.0.2), a PADICOTM-enabled version of *MPICH* (1.1.2), a customized version of PM 2 , and a regular version of *Expat* (1.95.2)

PADICOTM 0.1.5 (without external software) includes 31,000 lines of C and C++ (ca. 900 kB), 2,300 lines of JAVA (ca. 70 kB) and 7,000 lines of|shell|, |make| and |configure| scripts (ca. 200 kB).

The CVS version (0.3.x series) includes an updated version of PADICOTM core (bug fixes as well as some internal rewriting), *PadicoControl*, *myCORBA* (replaces *Ugo*) and includes external software: a customized version of PM 2 and a regular version of *Expat* (1.95.2). One major feature of this version is that is does not require any special version of supported middleware systems. Current supported middleware systems are *omniORB3*, *omniORB4* and *Mico* 2.3.x for CORBA, *MPICH* 1.1.2 and *MPICH* 1.2.5 for MPI and *gSOAP* 2.6.x for SOAP.

Users: 147 external downloads with 116 unique IPs between July 2002 and October 2004.

PADICOTM has been funded by the French ACI GRID RMI. As we are aware of, it is currently used by several French projects: ACI GRID HydroGrid, ACI GRID EPSN, RNTL VTHD ++ and INRIA ARC RedGrid. It is also used in the European FET project POP.

5.3. PaCO++

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

Participants: André Ribes, Christian Pérez, Thierry Priol.

Contact: Christian Pérez

URL: http://www.irisa.fr/paris/Paco++/

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

License: GNU General Public License version 2 and GNU Lesser General Public License version 2.1.

Presentation: The PACO++ objectives are 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 accessible externally through an object reference whose interpretation is identical to a standard CORBA object.
PACO++ extends CORBA but not to modify the model because we aim at defining a *portable* extension to CORBA so that it can be added to any CORBA implementation. This choice stems also from the consideration that the parallelism of an object appears to be an implementation issue of the object. Thus, the OMG IDL is not required to be modified.
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++ has started end of 2002. It represents 40 person-month effort. The first public version, referenced as PACO++ 0.1 has been released in November 2004. It has been successfully tested on top of three CORBA implementations: *Mico*, *omniORB3* and *omniORB4*. Moreover, it supports PADICOTM.
The version 0.1 of PACO++ includes 7,000 lines of JAVA (ca. 250 kB), 5,000 lines of Python (ca. 390 kB), 14,000 lines of C++ (ca. 390 kB) and 2,000 lines of |shell|, |make| and |configure| scripts (60 kB).
PACO++ is supported by the French ACI GRID RMI. Non-public beta versions are used by several French projects: ACI GRID HydroGrid, ACI GRID EPSN, RNTL VTHD ++ and INRIA ARC RedGrid.

5.4. CASPer

Keywords: *Application Service Provider*, *Grid Services*.

Participants: Guillaume Mornet, Jean-Louis Pazat.

Contact: Jean-Louis Pazat, http://www.telecom.gouv.fr/rntl/AAP2001/Fiches_Resume/CASPER.htm

License: LGPL

Presentation: CASPER aims at providing an *Application Service Provider* (ASP) for Grid computing.
The server side is based on the *Globus Toolkit* (GTK 3): CASPER is made of services that communicate with well-defined protocols, mainly XML-RPC calls (for *Grid Services*), JDBC connections (for databases) and HTTP connections. The ASP manages authentication, user interface, persistent data storage, job scheduling. Batch queues provide computing power for jobs submitted by users through the ASP.
On the Client side, CASPER can work with any standard Web Browser, this ensures that CASPER will be usable from most platforms.
CASPER is partly built using *components off the shelf* (COTS) for the Web browser, Web server (*TOMCAT*), SQL database (*MySQL*). The job managers currently targeted are OpenPBS, LSF and LoadLeveler. A Distributed Job Manager (*XtremWeb*) will be also be integrated within CASPER as a special job manager.
Security in CASPER is managed at different layers: first, we secure the HTTP connection between the client and the ASP (SSL and certificates), then we secure the communications between the ASP and batch queues (services have certificates). This is needed because batch queues may be spread across Virtual Organizations).
CASPER provides a Job Scheduler as a service responsible for scheduling job requests from users to the appropriate batch queue. Criterion for scheduling include: the required architecture, the list of queues the user is authorized to run jobs on, the current state of the queue, the job type (e.g., parallel or distributed), etc.
User management is done by a module that takes care of generating certificates, and updating access control lists (ACLs).
A CASPER application is made of a GUI which main functions is selecting job submission parameters, and a Job Runner that requests a job submission to the job scheduler in order to submit the code that executes the simulation.
Computations results are transferred from the batch queue to the ASP using the RFT Grid Service (which relies on a secured FTP protocol). These files will be stored on the ASP, with owner information. The result files can be remotely viewed (if a suitable viewer applet is available), downloaded, or deleted. The CASPER security manager controls access to the files.

Current status: The CASPER ASP is under development. The current release is for internal testing only. This project started in October 2003 and is supported by a RNTL contract. The main industrial contractor is EADS-CCR.

5.5. Mome

Keywords: *DSM*, *data repository*.

Participant: Yvon Jégou.

Contact: Yvon Jégou

Status: Prototype under development

Contact: Yvon Jégou, 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 DSM handles the consistency of mapped memory regions at the page-level. Two consistency models are currently implemented and can be selected by the user programs at the page level: the classical sequential model and an explicit weak model. MOME initial target was the execution of programs from the high performance community which exploit loop-level parallelism using a SPMD computation model. the current release of MOME supports page aliasing, the coupling of heterogeneous applications through shared memory, in-memory checkpointing and the dynamic connection of processes.
The current developments around MOME involve the implementation of an OpenMP runtime system, the integration of the release consistency model, the hierarchical implementation of the consistency protocols for federations of clusters and the deployment of a persistent data repository for the grid.

Current status: MOME is implemented in C (50,000 lines) and represents a 24-person-month effort. The current release is MOME 0.8. The DSM has been used in ALCATEL collaboration (checkpointing), in VTHD contracts (code coupling using a DSM), in *e-Toile* project (DSM-based data-repository for grid computing) and in the POP project (OpenMP runtime).

5.6. JuxMem

Keywords: *JXTA*, *Peer-to-peer*, *data grids*, *large-scale data management*.

Participants: Gabriel Antoniu, Luc Bougé, Mathieu Jan, Sébastien Monnet.

Contact: Gabriel Antoniu, http://www.irisa.fr/paris/Juxmem/

License: Not yet defined.

Presentation: JUXMEM is a supportive platform for a data-sharing service for grid computing. The 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. Several studies on replication strategies for fault tolerance and consistency protocols for volatile environments are under way within the framework provided by JUXMEM. A more detailed description of the approach is given in 6.4.2.

Current status: JUXMEM is still in a development phase. It is implemented in JAVA, based on the *JXTA* generic platform for P2P services (Sun Microsystems, http://www.jxta.org/). 20,000 lines of Java code. Implementation started in February 2003.
JUXMEM is the central framework based on which a data-sharing service is currently being built in collaboration with the GRAAL (Lyon) and REGAL (Paris) research groups, within the framework of the GDS (Grid Data Service) project of the ACI MD (see Section 8.2.8). In this context, a hierarchical failure detector developed by REGAL has been integrated with JUXMEM (7,000 lines of Java code, not taken into account above).

5.7. GridPrems

Keywords: *collaborative resource management*.

Participants: Yvon Jégou, Guillaume Mornet.

Contact: Guillaume Mornet

License: Not yet defined.

Presentation: GRIDPREMS is a collaborative resource manager for the PARIS and GRID 5000 experimental platforms. Registered users can select and reserve computation nodes, or consult the current reservations, through the web interface of GridPrems. Reservations can be exclusive (no reservation overlap of the same resource) and periodical. The *calendar* page of GridPrems provides the user with a global view of all reservations using a calendar presentation. GridPrems is accessible through Internet at https://www.irisa.fr/gridprems and is password-protected.

5.8. Other software

Other software not yet distributed (Adage by Sebastien L., Gabriel (done), Anne-Marie, ...)

5.8.1. ADAGE:

Sébastien Lacour and Christian Pérez (Contact: mailto://Christian.Perez@irisa.fr, License: Not yet defined, Keywords: Grid, middleware system, CORBA, deployment. Status: Under development)

ADAGE (Automatic Deployment of Applications in a Grid Environment) is a prototype middleware designed to automatically launch distributed or parallel applications on the resources of a computational grid. The middleware requires two pieces of information: a packaged, self-described application and a description of the resources available in the grid. The application to be deployed can be a CORBA component assembly or an MPICH-G2 parallel code. Resource description includes compute nodes and their characteristics (operating system, architecture, storage space, memory size, CPU speed and number, etc.) as well as network information (topology, performance characteristics). Using those two pieces of information, ADAGE automatically selects resources which will run the application and maps the application processes onto the selected resources. Finally it automatically launches the application processes remotely using the Globus Toolkit (version 2) as a grid access middleware, and initiates the application execution.

5.8.2. P2PSim:

Anne-Marie Kermarrec (Contact: mailto://Anne-Marie.Kermarrec@irisa.fr, License: Not yet defined, Keywords: P2P, simulation. Status: Under development)

P2PSim is a discrete-event simulator developed in C# to evaluate various search algorithms in peer-to-peer systems. More specifically, we used it during the past year to create additional semantic links between related peers in a distributed system and observe the hit ratio obtained for search queries using such links.

5.8.3. JDF:

Gabriel Antoniu and Mathieu Jan (Contact: mailto://Gabriel.Antoniu@irisa.fr, License: Sun Project JXTA Software license, Keywords: P2P, JXTA, deployment. Status: Under development)

JDF is a deployment and benchmark tool whose goal is to facilitate automated testing of JXTA-based systems. It provides a generic framework allowing to easily define custom tests, deploy all the required resources on a distributed testbed and run the tests with various configurations of the JXTA platform. JDF was initiated by Sun Microsystems and enhanced by Mathieu Jan (project owner) and Gabriel Antoniu (contributer).

JDF is based on a regular Java Virtual Machine (JVM), a Bourne shell and |ssh| or |rsh|. File transfers and remote control are handled using either |ssh|/|scp| or |rsh|/|rcp|. JDF assumes that all the physical nodes are visible from the control node. JDF is run through a regular shell script which launches a distributed test. This script executes a series of elementary steps: install all the needed files; initialize the JXTA network; run the specified test; collect the generated log and result files; analyze the overall results; and remove the intermediate files. Additional actions are also available, such as killing all the remaining JXTA processes. This can be very useful if the test failed for some reason. Finally, JDF allows one to run a sequence of such distributed tests.

5.8.4. Vigne:

Louis Rilling and Christine Morin (Contact: mailto://Christine.Morin@irisa.fr, License: Not yet defined, Keywords: cluster federation, transparent data sharing service, high availability, P2P. Status: Under development)

Vigne is a prototype of a Grid-aware system for cluster federations. It currently implements a peer-to-peer overlay network inspired from *Pastry* [83] and, on top of it, a transparent data sharing service based on the sequential consistency model and able to handle an arbitrary number of simultaneous reconfigurations. Vigne prototype has been developed in C and includes 20,000 lines of code. This prototype has been coupled with a discrete event simulator. The use of this simulator enabled us to evaluate the Vigne system in systems composed of a large number of nodes.

6. New Results

6.1. Operating system and runtime for clusters

Keywords: *Cluster*, *MPI*, *OpenMP*, *Pthread*, *checkpointing*, *cluster federation*, *cooperative caching*, *data stream migration*, *distributed file system*, *distributed shared memory*, *distributed system*, *global scheduling*, *high performance communication*, *high availability*, *multithreading*, *operating system*, *peer-to-peer*, *process migration*, *remote paging*, *self-organizing system*, *single system image*, *synchronization*.

6.1.1. Kerrighed

Participants: Pascal Gallard, Emmanuel Jeanvoine, Renaud Lottiaux, David Margery, Christine Morin, Louis Rilling, Etienne Rivière, Isaac Scherson, Gaël Utard, Geoffroy Vallée.

The PARIS Project-Team is engaged in the design and development of KERRIGHED, a genuine *Single System Image* cluster operating system for general high-performance computing [17][37]. A genuine SSI offers users and programmers the illusion that a cluster is a single high-performance and highly available computer, instead of a set of independent machines interconnected by a network. An SSI should offer four properties: (1) *Resource distribution transparency*, i.e., offering processes transparent access to all resources, and resource sharing between processes whatever the resource and process location; (2) *High performance*; (3) *High availability*, i.e., tolerating node failures and allowing application checkpoint and restart [40]; and (4) *Scalability*, i.e., dynamic system reconfiguration, node addition and eviction, transparently to applications.

6.1.1.1. Current achievements with Kerrighed

In 2004, two major releases have been delivered. KERRIGHED V0.80, released in January 2004, is the first release based on Linux 2.4. kernel. KERRIGHED V1.0, released in November 2004 [57], is the first stable release implementing the resource distribution and high performance SSI properties and process checkpointing. The robustness of KERRIGHED has been significantly enhanced, and several new functionalities have been implemented. KERRIGHED V1.0 is suitable for the execution of wide range of applications, including legacy

OpenMP and MPI applications (HRM1D, Cathare, Gorf 3D, ...) provided by our industrial partners (EDF, DGA).

Since August 2004, Geoffroy Vallée, has integrated the OSCAR Team at the Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA in order to integrate Kerrighed in OSCAR. OSCAR (http://oscar.openclustergroup.org/) is a distribution for Linux clusters which provides a snapshot of the best known methods for building, programming and using clusters. The first SSI-OSCAR package based on Kerrighed has been built and released at Supercomputing 2004 [60].

6.1.1.2. Scheduling policies

KERRIGHED provides a *configurable* global scheduler. Thus, it is possible to adapt the global scheduling policy to the workload characteristics [13][53]. In 2004, we have experimented different dynamic load balancing policies. In particular, we studied, during the master internship of Etienne Rivière, global scheduling policies that take distributed shared memory into account [59]. We designed such a policy and implemented it using Kerrighed global scheduling framework.

A batch interface on top of KERRIGHED is a user requirement. We started the design of a batch system exploiting KERRIGHED features. One important goal of this study is to avoid redundant functionalities between KERRIGHED and the batch system by keeping the batch system as simple as possible (essentially limited to the job submission interface).

6.1.1.3. Distributed file system

A distributed file system has been designed and implemented to exploit the disks attached to cluster nodes. This new file system provides a unique naming space cluster wide and allows to store the files of a directory in different disks in the cluster. It has been implemented based on the container concept, originally proposed for global memory management. Containers have been extended to manage not only memory pages but generic objects. The meta-data structures of the file system are kept consistent cluster wide using object containers. The performance evaluation has shown that files read and write accesses are efficient compared to other SSI distributed file systems.

6.1.1.4. Data stream migration

In order to cope with the migration of communicating processes, we have designed the *Kernet* System [11] to support the global management of any data stream in KERRIGHED. In 2004, the *pipe* interface has been implemented on top of *Kernet*. Performance evaluations have been pursued with real (industrial) MPI applications based on the MPICH environment.

Kernet relies on the *Gimli/Gloïn* system. It is a portable, high-performance communication system, providing to KERRIGHED distributed services at kernel-level active messages and pack/unpack primitives in addition to the traditional send/receive interface. The implementation of the new *Gimli/Gloïn* architecture, which offers high-performance communication both at kernel level and at user level, has been finalized [11].

6.1.1.5. Unix process interface

Process identifiers, unique cluster wide, have been implemented. All traditional Unix process management commands such as top, ps, kill operate at the cluster scale, based on these unique process identifiers.

6.1.1.6. Capabilities

We have proposed and implemented capabilities in KERRIGHED as a tool to enable or disable cluster SSI mechanisms on a per process basis. A process inheritates its capabilities from its parent process. System commands are also provided to modify process capabilities. For instance, capabilities can be used to enable/disable remote process creation, checkpointing, process migration, etc. Capabilities have revealed to be very useful for the debugging of KERRIGHED, allowing to selectively deactivate some of the SSI features during the execution of an application. It is also a powerful tool to allow an application to make the best use of the SSI properties, as it has been demonstrated with the *Ligase* application provided by DGA.

6.1.1.7. Evaluation

In 2004, an important effort has been done in the evaluation of KERRIGHED performance. We have installed on the same cluster the three open source Linux based SSI operating systems: KERRIGHED, OpenMosix and

OpenSSI. Performance measurements have shown that KERRIGHED outperforms the two other SSI systems for all common mechanisms [54][47].

Moreover, KERRIGHED features have also been evaluated by executing several real applications provided by our industrial partners: multithreaded, OpenMP and MPI applications.In particular, we carried out a study of the programming techniques for a computing cluster through applications such as the 2D FFT and Shearsort. In both cases, we developed techniques to efficiently utilize the cluster network to minimize the communications overhead associated with the in-memory matrix transposition problem [37].

Finally, as part of the Procope project in cooperation with Ulm University, we have conducted a performance comparison between Kerrighed shared memory system, Mome distributed shared memory (DSM) and the Plurix system based on a low level DSM. This is the first step of a larger study consisting in comparing fault tolerance strategies implemented in these three DSM systems [21].

6.1.1.8. High performance cluster-wide I/0

High performance I/O are of primary importance for the applications executed on clusters. Efficient but costly SAN solutions are available. However, as of today, no cluster file system provides all the performance one can expect from the internal disks and from a single interconnection network.

Rather than putting forward another middleware, we explore a new approach to make the operating system capable of efficient distributed I/O. We propose to share kernel objects related to the file system across the whole cluster. Apart for the consistency management operations achieved through Distributed Shared Memory (DSM) techniques, the I/O code path remains strictly identical to the centralized one. As objects tend to migrate to the right nodes, and as the path of I/O request and data is as short as possible, our system shows very good performances. Moreover, it reuses existing file system code and layout. It is also compatible with striping and mirroring as in software RAID systems [52].

A prototype has been implemented based on a modified version of the Linux Kernel. Our approach has been validated with respect to standard benchmarks. We also plan to design and implement in-kernel collective I/O operations to achieve very high MPI-IO performance.

6.1.2. Grid-aware Operating System

Participants: Emmanuel Jeanvoine, Christine Morin, Louis Rilling, Isaac Scherson.

We have worked on the design of a *Grid-aware* operating system that could federate clusters in order to make them cooperate, in particular for sharing resources. It should be able to manage a large number of nodes, and to deal with the dynamicity inherent to a federation, where multiple reconfigurations may be in progress at the same time. Our proposal is based on a *peer-to-peer* infrastructure. The *Grid-aware* operating system would encompass several distributed services such as, for instance, services assembling a federation, managing and scheduling applications, controlling resource access, managing a virtual shared memory and a distributed file system, etc.

In 2004, we have carried on implementing our infrastructure for a cluster federation. This infrastructure is organized in a structured overlay network using the algorithms of Pastry. This infrastructure is resilient to simultaneous reconfigurations including joins, leaves, and failures of nodes.

On top of this infrastructure, we have implemented a data sharing service based on the consistency algorithms we have designed in 2003 [49]. This data sharing service enables the components of a distributed application to transparently share and cache data. The data sharing service ensures the consistency of the cached data despite any number of simultaneous reconfigurations. In future implementations and using the full algorithms we have designed in 2003, the data sharing service will ensure the availability of the shared data to the application despite up to a fixed number of simultaneous reconfigurations. Coupled to a mechanism of application checkpoint and restart [36][35], our data sharing service will ensure the progress of the distributed application despite up to a fixed number of simultaneous reconfigurations.

We started to evaluate our data sharing service with the *paraci* cluster of workstations, which is located at IRISA, and with a simulator. We have coupled our prototype of a system for cluster federation to a discrete event simulator. On top of the resulting simulator, we have simulated the execution of a distributed application

that uses the data sharing service. We have simulated the execution in both static and dynamic configurations. The dynamic configuration has been derived from traces of the Gnutella peer-to-peer file sharing application in the Internet.

The evaluations show that using data sharing service leads to significant speed-ups on parallelized applications. The simulations show that the dynamic aspect of the configuration has a greater impact on communication delays than on the number of restarts an application undergoes [49].

In October, we have started the study of a resource allocation service which will be able to discover available resources in a cluster federation and to allocate these resources to applications. The resource allocation service is being designed to be fully decentralized and to be able to cope with the dynamicity of a cluster federation.

Finally, we also started studying security issues in federations of clusters with the internship of Jamal Ghaffour, a master student [56]. This internship aimed at providing a decentralized mechanism to authenticate nodes accessing to shared resources. A public-key-based protocol with no single point of failure has been designed and validated by model checking.

6.1.3. Mome and openMP

Participants: Yvon Jégou, Christian Pérez.

The OpenMP specification targets SMP architectures: shared memory multiprocessors. In the OpenMP model, all variables are implicitly shared. The private variables (one instance per thread) must be explicitly specified. It is not possible through static analysis to decide at compile-time which objects are shared and which ones are private.

The MOME DSM implementation and the associated runtime system have been adapted in order to support standard OpenMP codes without adding complexity to compilers: the thread stacks can be allocated in the shared space, the signal handlers are executed on private stacks, the DSM internal code never read or write in the application space, the distributed synchronization objects are allocated in the shared space but the primitives do not touch the objects etc.

A new implementation of the |nth_lib| runtime system from the IST POP project on the MOME DSM has been done and the benchmarking and experimentations are planned for the end of 2004. The integration of the release consistency model in MOME is in progress.

6.2. Middleware for computational grids

6.2.1. The PadicoTM framework

Keywords: *CORBA*, *Communication framework*, *MPI*.

Participants: Alexandre Denis, Christian Pérez, Thierry Priol, André Ribes.

Computational grids exhibit parallel and distributed aspects: it is a set of various and widely *distributed* computing resources, which are often *parallel*. Therefore, a grid usually contains various networking technologies — from system area network through wide area network. PADICOTM is a communication framework that decouples application middleware systems from the actual networking environment. Hence, applications become able to transparently and efficiently utilize any kind of communication middleware (either parallel or distributed-based) on any network that they are deployed on. Moreover, to support advanced grid programming models, PADICOTM is able to concurrently support several communication middleware systems.

The year 2004 has been devoted to the stabilization of PADICOTM, the port of JXTA and to the design and implementation of a module selection framework.

The availability of the C version of JXTA (JXTA-C) enabled us to do a porting of JXTA on top of PADICOTM with the goal of enabling JUXMEM of top of PADICOTM. As JXTA-C is based on the Apache Portable Runtime (APR), the porting consisted in improving the support of PADICOTM with respect to the C library (Database module). Benchmarking are expected by the end of 2004.

The major improvement of PADICOTM in 2004 is the development of a module selection framework. In previous version of PADICOTM, only a best-effort strategy was defined and hard-coded into PADICOTM:

Madeleine was used if available; otherwise plain TCP/IP. So, it was not possible to simply select advanced modules like parallel streams, on-line compression, etc.

To face the diversity of available communication selection strategies, we decided to virtualize the communication selection module. The principle is to delegate this role to an oracle module that returns an actual communication assembly (i.e. the list of module) the first time PADICOTM needs to send or receive a message to/from a node. Using a configuration parameter (currently a communication tag), the oracle looks for a selection module that has been registered for this tag. Hence, a CORBA communication can use a different module assembly than a MPI communication. We have implemented two such selection modules: a best-effort module that provides the same behavior than the previous versions of PADICOTM and a complete description-based module that allows a user to fully describe the assemblies between nodes. The configuration is done through an XML file that needs to describe all the nodes and the assembly to use for each possible communication.

6.2.2. Parallel CORBA objects and components

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

Participants: Christian Pérez, Thierry Priol, André Ribes, Hinde-Lilia Bouziane.

The concept of (distributed) parallel object/component appears to be a key technology for programming (distributed) numerical simulation. It joins the well known object/component oriented model with a parallel execution model. Hence, a data distributed across a parallel object/component can be sent and/or received almost like a regular piece of data while taking advantage of (possible) multiple communication flows between the parallel sender and receiver. The PARIS Project-Team has been working on such a topic for several years. PACO was the first attempt to extend CORBA with parallelism. PACO++ is a second attempt that supersedes PACO in several points. It targets a portable extension to CORBA so that it can be added to any implementation of CORBA. It advocates the parallelism of an object is mainly an implementation issue: it should not be visible to users but in some special occasions. Hence, the OMG IDL is no longer modified. GRIDCCM is the evolution of PACO++ into the component model of CORBA.

The work carried out in 2004 is related to the definition of an abstract model, the support of communication scheduling plug in, the proposition of a model for handling exception between parallel objects/components and the finalization of a software distribution of PACO++.

First, we have proposed an abstract model defining the notion of parallel distributed entities. A parallel CORBA object is an example of such entities. The model contains three sub-models defining the communication phase (notion of entities reference, connection phase, communication and disconnection), the management of distributed data during a communication between two parallel distributed entities and, last, the management of exception between such entities. The abstract model can then be derived into concrete models such as to define the abstract model of PACO++ and GRIDCCM. Working with this abstract model enable us to focus on the problem of managing parallel entities. Then, these results can be quiet directly applied to an object oriented model (like CORBA object) or a component model (like CORBA component). It should be also possible to applied it to Web Service so as to define parallel Web Services.

Second, we have finalized our distributed data model to handle data distribution functionalities as well as communication scheduling functionalities. The model has been implemented in PACO++. Hence, a data distribution library like RedSym developed by the Scalaplix project or a communication library like the library done by the Algorille research team (Nancy, France) are easily and properly integrated into PACO++. Benchmarking is currently in progress. Preliminary results show that the overhead is quite small and the observed behavior is the one expected.

Third, we have finalized a first model for handling exception between parallel entities. The model contains three parts: the definition of three kind of parallel exceptions (simple, aggregated and complex), the definition of the behavior implemented by the server and the definition of the client view. A simple exception is an exception raised by only one node or an SPMD exception. An aggregated exception is made of the set of exception raised by several nodes of the server. A complex exception is an aggregated exception with (possibly) incomplete data. A server needs to specify the kinds of exception it may raise and to sort such

exceptions according to some priority order. A client needs to specify the kinds of exception it supports. Then, the middleware system has to filter the exception raised by a server to conform to the choice of a client. The feasibility of the model has been validated into PACO++: the overhead is negligible for non-exceptional method invocation.

Future work can be divided into three parts. First, we will continue to support PACO++ in particular for the ACI GRID HydroGrid project. Second, we would like to develop an operational prototype of GRIDCCM. Third, we would like to investigate advanced features like hierarchical components or the support of dynamicity in the creation and/or connection of components.

6.2.3. Parallel component deployment for computational grids

Participants: Sebastien Lacour, Christian Pérez, Thierry Priol.

The deployment of parallel component based applications is a critical issue in the utilization of computational Grids. It consists in selecting a number of nodes and in launching the application on them. A first issue was to accurately describe the resources. We have proposed a description model for grid networks that provides a *synthetic* view of the network topology. It is complementary to previous work that succeeds in describing properly the compute nodes (CPU speed, memory size, operating system, etc), but generally fails to describe the network topology and its characteristics in a simple, synthetic and complete way.

In 2004, we have focus on specifying an architecture for an automatic deployment of application in a Grid environment. The architecture aims at describing the entities needed for an automatic deployment as well as their relationships. These entities can be grouped into three parts, each of them actually corresponding to a phase of the deployment process: the inputs (the component assembly and a grid resource description), the planner, which selects the resources and maps each component on a computer, and the actual deployment of the components on the selected resources.

We have started the development of ADAGE, an implementation of the proposed architecture. It is currently able to deploy standard CORBA Component based application on Grids manage by the Globus Toolkit 2.

We have just started to deal with parallel-based code. It seems that our approach can be extended to support the automatic deployment of MPICH-G2 application. An important goal is the deployment of GridCCM based applications that make use of PADICOTM. Not only, we have to face the complexity of deployment parallel component but, their are two levels of components to handle: CORBA component as well as PADICOTM component.

6.2.4. Adaptive components

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

Since Grid architectures are also known to be highly dynamic, using resources efficiently on such architectures is a challenging problem. 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.

We have defined a parallel self-adaptable component as a parallel component which is able to change its behavior according to the changes of the environment [18]. Such a component includes an adaptation *policy*, a set of available implementations, called *behaviors*, and a set of *reactions*. Reactions are the means by which the component adapts itself. It can be for example the replacement of the active implementation, the tuning of some parameters, the redistribution of arrays.

In order to adapt dynamically a parallel software component, we needed to coordinate all its processes before the execution of a reaction. We have formally defined and implemented an agreement algorithm to find the next point where an adaptation can be achieved [43].

In the next future, we will build the external mechanisms to manage parallel adaptable components and we will experiment our ideas with different real applications and environments. Then we will extend our work to the adaptation of several distributed cooperating components which will require coherent adaptation decisions.

Adaptation also involves the management of the resources. In this area we will study the relations between adaptation, allocation and scheduling policies.

This will constitute an axis for further work in order to provide a generic adaptation framework able to efficiently manage multi-applications in heterogeneous environments.

6.3. P2P System Foundations

6.3.1. Clustering in peer-to-peer systems

Keywords: *Peer-to-peer file sharing systems*, *clustering*, *semantic links*.

Participant: Anne-Marie Kermarrec.

Peer-to-peer file sharing systems have grown to the extent that they now generate most of the Internet traffic, way ahead of Web traffic. Understanding workload properties of peer-to-peer systems is necessary to optimize their performance. We studied clustering properties [30] of peer-to-peer file sharing workload along two directions: *(i)* we analyzed a workload gathered by crawling the eDonkey network, a dominant file sharing system, for over 50 days and, *(ii)* we exploited the clustering properties of peer-to-peer file sharing workload by creating additional semantic connections between related peers. This work has been done in tight collaboration with Fabrice Le Fessant (COMETE Project-Team, INRIA Futurs). During this analysis, we confirmed the presence of some well-known features, such as the prevalence of free-riding and the Zipf-like distribution of file popularity. We then focused on clustering properties. First, we analyzed the overlap between contents offered by different peers. We found that peer contents tend to be clustered, which may be taken as evidence that peers possess specific interests [34]. Then we leveraged this property by maintaining a list of *semantic neighbors*, i.e., peers with similar interests [28]. We evaluated the relevance of these semantic links, and therefore the presence of clustering, on P2PSim and observed convincing results. We plan to go further in that direction by using meta data associated with items in order to detect the clustering between peers.

6.3.2. Querying peer-to-peer systems

Keywords: *Search in peer-to-peer systems*, *publish-subscribe systems*, *range queries*.

Participants: Anne-Marie Kermarrec, Etienne Rivière.

Efficient search algorithms are crucial for a wide range of distributed applications. We worked in this area along two main directions: content-based publish-subscribe systems and generic query mechanisms. In publish-subscribe systems, subscribers register their interest in an event or a pattern of events in order to be be asynchronously notified of any event published matching their subscription. On the contrary, query mechanisms are symmetric: items are stored permanently and queries are the events looking for matching items. While existing P2P generic infrastructures provide a scalable support for topic-based publish-subscribe systems, they are not well adapted to content-based ones (in which events are filtered according to their content). We started a new collaboration with Sidath Handurukande and Rachid Guerraoui (EPFL, Switzerland) in this area. We defined a dedicated overlay network, where the overlay structure reflects the actual structure of the underlying application properties. In this approach, subscription filters are arranged according to a dictionary-based semantic. This overlay relies on gossip messages to construct a structure eventually similar to a perfect Skip list, preserving the semantic locality of the items stored in the overlay. We plan to extend this approach to support range queries in the future.

Our research in generic query mechanisms in large-scale systems has just started. Our long term goal is to build a unified search platform able to deal with the various kind of queries in peer-to-peer applications. A wide range of applications should benefit from such a framework including resource discovery in Grid computing.

6.3.3. Unstructured peer-to-peer overlays

Keywords: *Unstructured peer-to-peer overlays*, *application-level multicast*, *gossip-based algorithms*.

Participant: Anne-Marie Kermarrec.

Peer-to-peer self-organizing unstructured overlays networks have proven to provide good support for several distributed applications [16]. In this area, we worked along two directions. First we designed an efficient tree-based multicast system relying on a peer-to-peer unstructured overlay. In this approach, the unstructured

overlays is refined in a way that reflects geographic locality and evenly balances the number of neighbors of each node in the overlay, thereby sharing the load evenly as well as improving the resilience to random node failures or disconnections. We then build trees as sub-graphs of the optimized overlay, and use them to perform efficient application-level multicast.

Second, in collaboration with Maarten van Steen (VU, Amsterdam), Mark Jelasity (University of Bologna, Italy) and Rachid Guerraoui (EPFL, Switzerland), we compared various gossip-based protocols to build unstructured overlay network [29]. We plan to further compare these approaches in the context of specific applications such as application-level multicast and aggregation.

6.4. Large-scale data management for grids

6.4.1. Mome e-Toile

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 production sites to the execution sites. Moreover, in high performance simulation domain, the applications are themselves parallel programs and the grid sites are clusters of computation nodes. Each process of the parallel application needs only part of the input data and produces a part of the results. Duplicating the input data from a central server and then gathering the results after the execution can be expensive.

The participation of the PARIS Project-Team to the *e-Toile* project (http://www.urec.cnrs.fr/etoile/, ended June 2004) aimed at the experimentation of Distributed Shared Memory technology for the implementation of uniform data-naming and data-sharing services for grid computing. The current implementation is based on the MOME DSM. A MOME daemon process is launched in the background on each node of the grid. When the execution of some application starts on MOME-aware computation nodes, each of its parallel processes connects to local MOME daemon. The data-repository interface provides entry-points for the creation and for the localization of segments in the DSM (through a kind of directory), and for the mapping of these segments in the local address space of the process. The data repository is persistent: the segments retain their data after all application processes have disconnected. The application processes can fail safely (or be killed) without impacting the DSM. The system provides a kind of uniform data space to the grid applications.

The current version of MOME considers a flat organization of the DSM nodes. On a grid infrastructure, the performance of the communication system inside a grid node (a cluster) is higher than between grid nodes. The DSM should be aware of this structure. A new hierarchical organization of the MOME DSM has been defined and currently being implemented.

6.4.2. The JuxMem data-sharing service

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

Participants: Gabriel Antoniu, Luc Bougé, Mathieu Jan.

With JUXMEM, we propose the concept of *data-sharing service* for grid computing, as a compromise between two rather different kinds of data sharing systems: (1) *DSM systems*, which propose consistency models and protocols for efficient transparent management of *mutable data, on static, small-scaled configurations (tens of nodes)*; (2) *P2P systems*, which have proven adequate for the management of *immutable data* on *highly dynamic, large-scale configurations (millions of nodes)*.

The main challenge in this context is to define appropriate models and protocols allowing to guarantee both *consistency* of replicated data and *fault tolerance* in *large-scale, dynamic environments*.

To tackle the issues described above, we have defined an architecture proposal for a data sharing service. This architecture mirrors a federation of distributed clusters and is therefore *hierarchical* and is illustrated through a software platform called JUXMEM (for *Juxtaposed Memory*). A detailed description of this architecture is given in [15]. It consists of a network of peer groups (|cluster| groups), each of which generally

corresponds to a cluster at the physical level. All the groups are inside a wider group which includes all the peers which run the service (the |juxmem| group). Each |cluster| group consists of a set of nodes which provide memory for data storage (called *providers*). All providers which host copies of the same data block make up a |data| group, to which is associated an ID. To read/write a data block, clients only need to specify this ID: the platform transparently locates the corresponding data block. This architecture is illustrated by a software prototype (development started in February 2003, currently in progress). The prototype is based on the JXTA [62] generic peer-to-peer framework, which provides basic building blocks for user-defined peer-to-peer services. In 2004 we have stabilized and refined this architecture, especially in order to add the necessary generic support for multiple consistency protocols (see Section 6.4.3). We have also designed and implemented a more efficient data allocation algorithm.

6.4.3. Fault-tolerant consistency protocols

Keywords: *consistency protocols*, *fault-tolerance*, *grid data sharing*, *peer-to-peer*.

Participants: Gabriel Antoniu, Luc Bougé, Jean-François Deverge, Sébastien Monnet.

We enriched JUXMEM's architecture so as to decouple consistency management from fault-tolerance management and to define a thin interface between these two aspects. Critical entities in consistency protocols are made fault-tolerant using an enriched version of the *group membership* abstraction. Each such entity (e.g. home node) is replaced by a set of nodes with the following properties: 1) All messages sent to such a group are received *by all members of the group, in the same order* (atomic multicast); 2) The groups are self-organizing: they maintain some user-specified replication degree by dynamically and adding new members when necessary in a "smart" way. This approach has been illustrated and by implementing and evaluating a fault-tolerant consistency protocol for the entry consistency model. Details are given in [20].

Future work will address extensions of our approach, in order to define an extended semantics of the consistency protocols. The goal is take into account cases where the assumptions made by the low-level fault-tolerant building blocks about the fault types and about the upper bound on the number of concurrent faults are not satisfied.

6.4.4. Large-scale deployment tools for P2P experiments.

Keywords: *JXTA*, *deployment*, *peer-to-peer*.

Participants: Gabriel Antoniu, Luc Bougé, Mathieu Jan, Sébastien Monnet.

Within the context of the experimental evaluation of JUXMEM in large-scale environments, we also focused on tools for large-scale experimentation of P2P systems. We stated *five commandments* which should be observed by a deployment and control tool to successfully support large-scale P2P experiments. Our contribution consists in enhancing the *JXTA Distributed Framework* (JDF) to fulfill some of these requirements. This enhancement mainly includes a more precise and concise specification language describing the virtual network of JXTA peers, the ability to use various batch systems, and also to control the volatility conditions during large-scale tests. Details including some preliminary performance measurements for the basic operations are reported in [19].

We used JDF to evaluate the performance of JXTA communication layers on top of Fast Ethernet and Myrinet networks [45] (internship of David Noblet, Univ. of New Hampshire).

Further enhancements are needed for JDF to fully observe the five commandments. A hierarchical, tree-like scheme for the |ssh|/|rsh| commands could be used to balance the load of copying files from the control node to other physical nodes. We plan to integrate a synchronization mechanism for peers to support more complex distributed tests. The final goal is to have a rich generic tool allowing to deploy, configure, control and analyze large-scale distributed experiments on a *federation of clusters* from a single control node. We intend to further develop this tool, in order to use it for the validation of JUXMEM. JDF could also be very helpful for other JXTA-based services, and the approach can be easily generalized to other P2P environments.

6.5. Advanced computation models for the Grid

Participants: Jean-Pierre Banâtre, Yann Radenac.

We are considering unconventional approaches for Grid programming and, more generally, for the programming of distributed applications.

It is well known that the task of programming is very difficult in general and even harder when the environment is distributed. As usual, the best way to proceed is by separation of concerns. Programs are first expressed in a model independent of any architecture, and then are refined taking into account the properties of the (distributed) environment. Several properties have to be taken into account, such as correctness, coordination/cooperation, mobility, load balancing, migration, efficiency, security, robustness, time, reliability, availability, computing/communication ratio, etc.

Our present work relies on the chemical reaction paradigm and more precisely on the Gamma model of programming. We believe that this model can be a nice basis for the construction of applications exploiting GRID technology. This work is carried out in close cooperation with Pascal Fradet, now at INRIA Rhône-Alpes (Project-Team *POP ART*).

Our recent contributions include the extension of Gamma to higher-order and the generalization of multiplicity. The extension of the basic Gamma model to a higher-order Gamma makes it possible to consider a Gamma program as a member of a multiset, thus eligible for reactions as any other element of the multiset. We have called this model, the γ-calculus. Actually, we have defined a hierarchy of γ-calculi, from the most basic one (multisets, basic γ-expressions) to a very rich higher-order γ-calculus, much richer than the original Gamma model. A paper [42] was presented at the *5th International Workshop on Rule-Based Programming (RULE '04)*.

A Gamma program is now a first class citizen and can be manipulated as any other kind of data. So, it can move across the GRID, it can be added to a chemical solution, it can be removed from a solution, etc., thus providing a very flexible way of programming the behavior of applications. We have demonstrated such an approach by programming an autonomic mail system in [22] presented at the *13th International Conference on Intelligent and Adaptive Systems, and Software Engineering (IASSE '04)*. This work has also been presented [23] at the *Unconventional Programming Paradigms (UPP '04)* workshop (see below).

Another generalization of the Gamma language stands in the introduction of multisets with infinite cardinality and multisets with a negative cardinality. These new kind of data structures, combined with the above high order properties, provide a very general and powerful tool for expressing very general (and original) coordination schemes. A presentation of this work will be available as an INRIA Research Report.

Apart from completing the above research activities, our present work concerns the design of a Chemical Middleware for GRIDs and its implementation. The programmer would express his/her application in a language which could be JAVA, JAVA-like (or Chemical-JAVA), or GAMMA and after some adhoc transformations (guided by the programmer) the application would be executed on the *Chemical GRID*.

A major event this year has been the organization of the *Unconventional Programming Paradigms (UPP '04)* workshop from September 14 to 17 in Le-Mont-Saint-Michel. This workshop (on invitation only) was organized under the auspices of the EU (Future and Emerging Technologies Program) and NSF. About 35 people gathered on topics such as Autonomic Computing, Amorphous Computing, Generative Computing, Bio-Inspired Computing and Chemical Computing. Discussions were very rich and lively. The pre-proceedings are available [9] and a book version will be published by Springer in Spring 2005.

6.6. Experimental Grid Infrastructure

Participants: Yvon Jégou, David Margery, Guillaume Mornet.

The PARIS Project-Team manages an experimental computation platform dedicated to operating system, runtimes, middleware, grid and P2P research. This platform is currently being integrated to the nation-wide grid infrastructure GRID 5000. In order to significantly increase the resources available for GRID 5000, our project-team received financial support from ACI GRID, INRIA, UNIVERSITY RENNES 1, and from the Brittany Regional Council. During 2004, 66 dual Intel Xeon from Dell (January), 33 dual Xserve G5 from Apple (September) and 66 dual AMD Opteron from Sun (October) have been added to our platform. Further extensions are planned in 2005.

In the mean-time, a prototype platform for the GRID 5000 architecture has been deployed and experimented between the seven GRID 5000 sites. Our GRID 5000 prototype is implemented using 12 Pentium III PCs running Linux. Different services for the GRID 5000 platform have been experimented on this prototype: file sharing (NFS), data synchronization between sites, naming services (DNS), networking (routing), security, shared data-bases (NIS, LDAP), home-dir and accounting, operating system deployment, distant reboot etc. The hardware of some of our *old* Pentium 3 has been modified in order to allow computer reset from distant sites.

The networking architecture of GRID 5000 exploits level-2 VLANs from Renater. GRID 5000 sites can use privates IP addresses if they wish to be unreachable from the external world. They can however use public IP addresses if they plan to collaborate with non-GRID 5000 platforms (this is our choice). The GRID 5000 VLANs protect all communications between GRID 5000 sites from the external world and facilitate the deployment of the experiments on the platforms (no authentication, no need for crypting).

All our computational resources have been integrated into this platform by the end of October and have been made available to GRID 5000 for the SuperComputing 2004 (SC 2004) event in Pittsburg, PA, November 6-12.

7. Contracts and Grants with Industry

7.1. VTHD ++

Participants: Yvon Jégou, Christian Pérez.

Program: The VTHD Project (http://www.vthd.org/) aims at deploying a broadband IP test platform to develop the technological bricks that will be required to deploy New Generation Internet and Intranet networks.

Starting time: March 2002.

Ending time: December 2004.

Partners: *France Télécom Recherche & Développement* (FT R&D), INRIA, *École Nationale Supérieure des Télécommunications* (ENST), *École Nationale Supérieure des Télécommunications de Bretagne*, IMAG and EURECOM institute.

Support: RNRT funding, Platform program

Project contribution: The PARIS Project-Team is involved in the VTHD ++ *Metacomputing* Sub-Project. We study code coupling and high-bandwidth data transfer between distant clusters. We have demonstrated a sustained transfer rate of 1.9 Gb/s between a PC cluster located in Rennes and another cluster located in Sophia-Antipolis (1000 km far apart) within a coupled numerical simulation.

7.2. e-Toile

Participant: Yvon Jégou.

Program: The *e-Toile* Project (http://www.urec.cnrs.fr/etoile/) aims at deploying a high-performance Grid platform. The project has investigated extensions to pre-existing Grid software, to make it suitable for HPC production Grids and has evaluated the costs and benefits of running applications on a Grid platform.

Starting time: December 2001.

Ending time: June 2004.

Partners: CEA, CNRS, CS (Communication & Systems), EDF, ENS Lyon (LIP), Université de Versailles Saint-Quentin (PRISM), INRIA (IRISA, ID-IMAG, RESO), Sun France, IBCP Lyon (since May 2002).

Support: RNTL funding, Platform program

Project contribution: The contribution of the PARIS Project-Team to the e-Toile Project focuses on the development of a *Distributed Shared Memory* (DSM) environment for the implementation of a persistent and distributed data repository for the Grid.

7.3. CASPer

Participants: Guillaume Mornet, Jean-Louis Pazat.

Program: The CASPER Project aims at defining a *Web-based computing portal* to use distributed computing resources.

Starting time: October 2002

Ending time: May 2005

Partners: EADS CCR, ALCATEL Space Industries, IDEAMECH, Université de Paris Sud (LRI)

Support: RNTL funding

Project contribution: The PARIS Project-Team defines the overall architecture and implements an OGSA based system for the core services of CASPER.

7.4. Edf 1

Participants: Christine Morin, Geoffroy Vallée.

Program: The collaboration with EDF R&D aims at designing and implementing an environment and tools for PC cluster management and use in the area of high performance computing.

Starting time: December 1st, 2000

Ending time: February 29th, 2004

Partners: EDF R&D, INRIA (IRISA, RESO)

Support: EDF R&D funding, PhD CIFRE Grant (Geoffroy Vallée)

Project contribution: The work carried out by the PARIS Project-Team relates to the design and implementation of KERRIGHED Single System Image (SSI) operating system for high-performance computing on clusters. In the framework of the EDF project, KERRIGHED configurable global scheduler has been designed and implemented as well as efficient global process management mechanisms to replicate, migrate and checkpoint processes. A development framework facilitating the implementation of dynamic scheduling policies in KERRIGHED has been developed. Experimentations carried out with applications provided by EDF R&D (HRM1D, Aster, Cyrano3, Cathare) have been conducted.

7.5. Edf 2

Participants: Christine Morin, Geoffroy Vallée.

Program: The collaboration with EDF R&D and ORNL aims at creating the SSI-OSCAR package in the OSCAR software suite for high performance computing on clusters and integrating KERRIGHED in OSCAR as the first SSI-OSCAR package.

Starting time: March 15th, 2004

Ending time: March 14th, 2005

Partners: EDF R&D, ORNL

Support: EDF R&D and ORNL funding, INRIA industrial post-doc grant (Geoffroy Vallée)

Project contribution: The work carried out by the PARIS Project-Team relates to the packaging of KERRIGHED and its integration in SSI-OSCAR. We also carry out experimentations with real industrial applications provided by EDF R&D. The first release of SSI-OSCAR has been presented at SC04 in November 2004.

7.6. Edf 3

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 1st, 2004

Ending time: September 30th, 2007

Partners: EDF R&D, INRIA

Support: EDF R&D funding, PhD CIFRE grant (Emmanuel Jeanvoine)

Project contribution: The work carried out by the PARIS Project-Team relates to the design and implementation of a Grid-aware operating system for cluster federations. As part of this contract, we design a resource discovery and allocation service based on an underlying peer-to-peer overlay network to cope with the decentralized and dynamic nature of a cluster federation. We also study application scheduling policies for cluster federations that will be evaluated experimentally with workloads provided by EDF R&D.

7.7. Dga

Participants: Renaud Lottiaux, David Margery, Christine Morin.

Program: The COCA contract comprises of two parts. The first one aims at designing, evaluating and optimizing a prototype high performance computing infrastructure well-suited for scientific numerical simulation. The second one relates to the problematic of the re-usability of numerical models. The PARIS Project-Team contributes to the first part of the COCA contract.

Starting time: January 15th, 2003

Ending time: November 14th, 2005

Partners: DGA, CGEY, ONERA-CERT

Support: DGA Funding

Project contribution: The high-performance computing infrastructure considered in the COCA contract is a federation of medium-size clusters, each cluster running a Single System Image (SSI) operating system. The work carried out by the PARIS Project-Team relates to the design and implementation of KERRIGHED SSI cluster operating system. Four successive releases of KERRIGHED will be delivered as part of the COCA contract with an increasing set of functionalities: (1) Global memory management (V0.70); (2) Global management of memory, processes, data streams and files (V1.0); (3) Checkpointing mechanisms for parallel applications (V.1.10, based on V1.0); and (4) Full-fledged SSI, highly available system (V2.0, based on V.1.10). Moreover, the PARIS Project-Team will study extensions to KERRIGHED operating system to make it a *Grid-aware* operating system for cluster federations.
In 2004, we have worked on the design and implementation of Kerrighed V1.0 and on the improvement of the system robustness.

8. Other Grants and Activities

8.1. Regional grants

GRID 5000: the PARIS Project-Team received a 40,000 Euros grant from the Brittany Regional Council to acquire additional network equipments for the GRID 5000 Platform.

PhD grants: The Brittany Regional Council provides half of the financial support for the PhD theses of Mathieu Jan (starting on October 1, 2003, for 3 years) and André Ribes (starting on October 1, 2001, for 3 years). This support amounts to a total of 28,000 Euros/year.

8.2. National grants

8.2.1. ACI GRID: Globalisation des Ressources Informatiques et des Données

The PARIS Project-Team is deeply involved in national initiatives related to the Grid. An initiative was launched by the *Ministry of Research* through the ACI program (*Action Concertée Incitative*). The ACI GRID (for *Globalisation des Ressources Informatiques et des Données*) aims at fostering French research activities in the area of Grid computing by providing financial support to the best research groups. The ACI GRID initiative was launched in 2001 and issued three calls for proposal (one every year). The PARIS Project-Team submitted proposals for each of them. The following paragraphs present an overview of the projects funded by the ACI GRID in which the project-team is involved.

8.2.2. ACI GRID ANIM

Participant: Thierry Priol.

T. Priol was asked by the Ministry of Research to be the director of the ACI GRID from January 2004 as M. Cosnard (the former director) was named as chairman of INRIA. T. Priol is responsible of a project funded by the ACI GRID to support the management of the whole ACI GRID initiative.

8.2.3. ACI GRID HydroGrid

Participants: Christian Pérez, André Ribes.

The HydroGrid project is a 3-year multidisciplinary project, started in September 2002. It aims at modeling and simulating fluid and solute transport in subsurface geological media using a multiphysic approach. Such multiphysic numerical simulation involve code featuring different languages and communication libraries (FORTRAN, MPI, OpenMP, etc.), to be run on a commun computational Grid. Therefore, the project relies on the results of the ACI GRID RMI project. A strong point of the HydroGrid project is to group together teams with different areas of expertise (from applications, scientific computing and computer science). The partners are the PARIS, Aladin and Estime Project-Teams at IRISA, the *Hydrodynamique et Transferts en Milieux Poreux* Team (IMFS Strasbourg) and the *Transferts physiques et chimiques* Team (Géosciences Rennes).

During the first two years of the project, different numerical coupling schema were studied and some of them were experimentally evaluated but mainly with sequential code. For the last year of the project, we plan to continue working on the numerical schema but we mainly target to be able to execute realistic simulations made of the coupling of parallel codes.

8.2.4. ACI GRID DataGRAAL

Participants: Gabriel Antoniu, Luc Bougé, Mathieu Jan, Sébastien Monnet, Thierry Priol.

This project gathers together the national research communities interested in large-scale data management. This includes communities involved in grid computing, operating systems, distributed systems and databases. The projects aims at understanding the common research issues and at defining a common terminology. An important goal of the project is encourage the emergence of software prototypes resulting from collaboration efforts of these communities.

The setup of the GDS Project of the ACI MD, coordinated by the PARIS Project-Team, in collaboration with the ReMaP/GRAAL and REGAL Research Groups, is a result of the discussions that took place within the framework provided by DataGRAAL. The DataGRAAL community is the initiator of a project for a Spring School on *large-scale data management* (DRUIDE 2004), which took place in May 2004 at Le Croisic, Brittany (85 participants). Gabriel Antoniu chaired the Program Committee and the Organizing Committee of this school. DataGRAAL started in November 2002 and ends in November 2004. Gabriel Antoniu is local correspondent of DataGRAAL for the PARIS Project-Team.

8.2.5. ACI GRID GRID2

Participants: Jean-Louis Pazat, Christian Pérez.

Jean-Louis Pazat is at the head of the GRID2 project. At many as 10 laboratories from various parts of France are involved in this 150,000-Euro project granted by the Ministry of Research for 3 years. Christian Pérez is in charge of the *Run-Time System and Middleware* Working Group. The objective of this project is to federate the Computing GRID research community by organizing meetings between researchers, teaching for young researchers and by achieving information dissemination.

GRID2 is divided in the following working groups: (1) Software architecture and languages; (2) Run-time systems and middleware; (3) Algorithms and models; (4) Algorithms and high performance applications. This project has organized a *Winter School on Grid Computing* in Aussois in December 2002 and two workshops during the *RenPar* Conference in 2002 and 2003. A number of *Hands-On Days* have taken place this year, enabling researcher to gain practical experience of topics such as *JXTA*, CORBA, numerical computing, etc.

8.2.6. ACI GRID Alta

Participants: Alexandre Denis, Christian Pérez.

Alta is a 2-year joint project funded by the ACI GRID of the French Ministry of Research, in cooperation with the INRIA Cooperative Research Initiatives. The PARIS Project-Team coordinates the project. It also involves *Runtime* Project-Team in Bordeaux, and the *Distribution and Parallelism* Team in Lille. It aims at studying the impact of tolerant loss-control in the context of asynchronous iterative algorithm. An objective is to define and to implement a dedicated API.

After almost two years of work, the project is reaching its objectives. A tolerant loss-control protocol has been proposed and implemented. Its usage is quite simple thanks to an extension of the Madeleine API. It has been validated into an application though more experimentation and validation is required.

8.2.7. ACI GRID Grid 5000

Participants: Yvon Jégou, David Margery, Guillaume Mornet.

GRID 5000 is a nation-wide initiative to build a research platform (ca. 5000 processors) for Grid computing. This large-scale distributed platform will enable experimentations on operating systems, middlewares, and communications libraries by the computer-science research community in France. In 2003, the PARIS Project-Team submitted a proposal for building a GRID 5000 node in Rennes. The project has been selected by the French Ministry of Research (ACI GRID) to be one of the 8 initial nodes of the GRID 5000 Computing Infrastructure and received a three-year grant of 200 kEuros. The integration of the first 66 processor boards (dual Xeon) to the PARIS cluster (50 PC and Xserve dual-processor nodes) has been initiated during November 2003. The exploitation of these node started beginning of January 2004. A 33-dual Apple Xserve G5 cluster running MacOS X was delivered end of September 2004, and a 66 dual Opteron V20z cluster from Sun Microsystems was installed by the end of October. All these clusters were in production for the Super Computing SC '04 presentation at Pittsburg, PA, November 6-12.

Handling communication inside clusters is an active research activity in the PARIS project. In the future, we plan to equip parts of our clusters with various high performance system networks: Ethernet, InfinityBand, SCI and Myrinet.

In 2004, the PARIS Project-Team submitted a proposal to ACI GRID for the equipment of the project GRID 5000 clusters with local system high-performance networks. The project has been selected by the French

Ministry of Research (ACI GRID) and received a three-year grant of 120 kEuros for high-performance system network equipments.

8.2.8. ACI MD: Masses de Données

The PARIS Project-Team is involved in the ACI MD (for *Masses de Données*). It aims at fostering research activities in the area of large-scale data management, including Grid computing. The first call for proposal was issued in 2003. The following paragraphs give a short overview of the project-team involvement in this initiative.

8.2.9. ACI MD GDS

Participants: Gabriel Antoniu, Luc Bougé, Mathieu Jan, Sébastien Monnet, Thierry Priol.

The GDS Project of the ACI MD gathers 3 research teams: PARIS (IRISA), REGAL (LIP6) and ReMaP/GRAAL (LIP). The main goal of this project is to specify, design, implement and evaluate a data sharing service for mutable data and integrate it into the DIET ASP environment developed by ReMaP/GRAAL. This service will be built using the generic JUXMEM platform for peer-to-peer data management (currently under development within the PARIS Project-Team, see section 6.4.2). JUXMEM will serve to implement and compare multiple replication and data consistency strategies defined together by the PARIS and REGAL research groups. The project started in September 2003 and will end in September 2006. It is coordinated by Gabriel Antoniu (PARIS). Project site: http://www.irisa.fr/GDS/.

In 2004, we mainly made progress on 3 topics. First, we defined the interaction between consistency protocols and fault-tolerance components. A hierarchical failure detector developed by REGAL has been integrated into JUXMEM (PARIS). Second, we enhanced the JDF deployment tool, in order to be able to run large-scale experiments of the data sharing service, while controlling the volatility conditions. Third, we studied how to integrate the deployment and visualization tools used by DIET and JuxMem, in order to have a unique control framework for joint experiments.

8.2.10. ACI MD MDP2P

Participant: Yvon Jégou.

The main objective of the ACI MD MDP2P project is to provide high-level services for managing text and multimedia data in *large-scale P2P systems*. The PARIS Project-Team contributes for the development of DSM-based (MOME and KERRIGHED) data management techniques on clusters of clusters for large-scale multimedia indexing.

8.2.11. ACI MD GdX

Participants: Gabriel Antoniu, Luc Bougé, Mathieu Jan, Sébastien Monnet, Thierry Priol.

The *Data Grid Explorer* (GdX) Project aims to implement a large-scale emulation tool for the communities of a) distributed operating systems, b) networks, and c) the users of Grid or P2P systems. This large-scale emulator consists of a database of experimental conditions, a large cluster of 1000 PCs, and tools to control and analyze experiments. The project includes studies concerning the instrument itself, and others that make use of the instrument. The GDS project of the ACI MD, coordinated by PARIS, is partner of GdX, as a user project. The project started in September 2003 and will end in September 2006. Our interaction with GdX will effectively start in 2005, once the physical platform becomes available. Gabriel Antoniu is local correspondent of GdX for the PARIS Project-Team. Project site: http://www.lri.fr/~fci/GdX/.

8.2.12. ACI CE: Support à la soumission de propositions de réseaux d'excellence

8.2.13. ACI CE CoreGRID

Participant: Thierry Priol.

This project (http://www.coregrid.net) aims at helping the PARIS Project-Team to prepare a *Network Of Excellence* proposal in the area of *Grid and Peer-to-Peer Computing*. This proposal was positively evaluated end of 2003. Negotiation started in March 2004 and ended in June. The funding given by the ACI CE

COREGRID was spent to cover the preparation of the proposal in 2003 and the negotiation phase in 2004. The official starting date of COREGRID is September 1, 2004.

8.2.14. Other grants

8.2.15. ARC RedGrid

Participants: Yvon Jégou, Christian Pérez, Thierry Priol, André Ribes.

This 2-year project is funded by the INRIA Cooperative Research Initiative (ARC) whose partners are the ReMaP, PARIS, Algorille and Scalapplix Project-Teams. Its objective is to study the issues related to data redistribution in a Grid environment, to develop data redistribution libraries and to apply the results in the environments develop by the partners (DIET, PACO++, GridCCM and EPSN).

8.2.16. CNRS AS 114, RTP 8

Participant: Yvon Jégou.

This one-year project, called *Étude préparatoire pour une plate-forme de grille expérimentale*, aims to identify issues (scientific and technical) and propose solutions in the perspective of building an experimental Grid platform gathering nodes geographically distributed in France.

8.2.17. CNRS AS 115 RTP 8

Participant: Christian Pérez.

This is a one-year project, called *Méthodologies de programmation des grilles*, that aims at identifying future research directions related to computational Grids programming. It encompass partners involved in applications, algorithms, runtimes/middleware systems and network protocols.

8.2.18. CNRS AS Distributed Algorithms

Participant: Anne-Marie Kermarrec.

This is a one-year project, called *Algorithmiques distribuées*, aiming at identifying future research directions in distributed systems and algorithms, peer-to-peer computing, mobile computing, etc. This working group is composed of French academics partners.

8.3. European grants

8.3.1. IST POP

Participants: Yvon Jégou, Christian Pérez.

The POP Project (IST Project 2000-29245) targets performance portability of OpenMP application. It is a 3-year project which has started in December 2001. The partners are the *European Center of Parallelism of Barcelona* (CEPBA-UPC, Barcelona, Spain), the *Istituo di Cibernitica* (IC-CNR, Naples, Italy), the *High Performance Information System Laboratory* (LHPCA-UP, Patras, Greece) and INRIA.

The POP Project was motivated by the adoption by the industry of the OpenMP language as a standard for shared memory programming. However, this standard is restricted to hardware shared memory machine. The POP project objective is to build an environment that, starting from an OpenMP application, is able to generate efficient code for different kind of machine architectures. In addition to hardware shared memory machine, the targeted architectures include distributed memory machines and multithreaded machined.

In particular, the project focus on three main goals. The first goal deals with the extension of OpenMP expressiveness to exploit parallelism in irregular task graphs, to improve work-distribution schemes among groups of processors so as to enforce data locality and to add a support for inspector/executor techniques. The second goal is to study the dynamic adaptability of the runtime to use self-analysis to modify the behavior of the application on runtime and to run the same binary file regardless of the underlying architecture, the input data, and the dynamic variation of available resources. The third goal concerns architectural modifications to efficiently execute OpenMP application on distributer memory machine or multithreaded machine.

The POP Project is based on the results of the Nanos European project. In particular, an OpenMP compilation and execution environment was developed for shared memory machines like the Origin 2000.

The PARIS Project-Team focus on the architectural modifications of existing software DSM to provide an adequate support for an efficient execution of OpenMP application on cluster. The set of critical SDSM features, we have identified during Year 2002, are being applied to the MOME SDSM, used in conjunction with PADICOTM. A first complete prototype of the POP Runtime is available on top of MOME.

8.3.2. CoreGRID

Thierry Priol is the Scientific Coordinator of a *Network of Excellence* proposal, called COREGRID, in the area of Grid and Peer-to-Peer (P2P). This network started on September 1, 2004. As many as 42 partners, mostly from 17 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. It is the primary objective of the COREGRID Network of Excellence 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, and more specifically distributed systems and middleware, programming models, knowledge discovery, intelligent tools, and environments.

The research program is structured around six complementary research areas 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 model, system architecture, Grid information and monitoring services, resource management and scheduling, problem solving environments, tools and Grid systems.

8.3.3. GridCoord

The *Specific Support Action* (SSA) *ERA pilot on a co-ordinated Europe-wide initiative in Grid Research* addresses the Strategic Objective 2.3.2.8 *Grid-based Systems for solving complex problems* and the Strategic Objective 2.3.6 *General Accompanying actions* as described in the IST Work Programme 2003-04. It has been launched in July 2004 for 18 months.

Currently several Grid Research initiatives are on-going or planned at national and European Community level. These initiatives propose the development a rich set of advanced technologies, methodologies and applications, however enhanced co-ordination among the funding bodies is required to achieve critical mass, avoid duplication and reduce fragmentation in order to solve the challenges ahead. However, if Europe wishes to compete with leading global players, it would be sensible to attempt to better coordinate its various, fragmented efforts toward achieving a critical mass and the potential for a more visible impact at an international level.

The goal of the GRIDCOORD SSA proposal is namely to achieve such a coordinated approach. It will require both: (1) Co-ordination among the funding authorities; (2) Collaboration among the individual researchers; (3) A visionary research agenda. This proposal is thus tightly connected to the COREGRID Network of Excellence proposal above, led by Thierry Priol at the European level.

The GRIDCOORD SSA proposal is led by Marco Vanneschi, University of Pisa, Italy. It includes 13 institutional partners from 9 European countries. The French partners are INRIA and University of Nice Sophia-Antipolis. The INRIA partnership is made of Isabelle Attali (INRIA Sophia-Antipolis, leader), L. Bougé and Th. Priol.

8.4. International bilateral grants

8.4.1. Europe

University of Ulm, Germany. A bi-lateral research collaboration with the distributed system group of the University of Ulm has been started in the 2004 *Procope* Program. We design and implement new checkpointing strategies for real applications running in different DSM environments. Three

DSM systems are considered in the study: *Plurix* system, developed at the University of Um, which is based on a DSM implemented at the lowest possible level; the KERRIGHED system, which implements a kernel-level DSM in Linux; and the MOME DSM, implemented in user space on top of Linux. The two latter systems are developed in the PARIS Project-Team.
As part of this collaboration, Peter Schultess, Michael Schöttner, Stefen Frenz and Ralf Göttermann from Ulm University participated to a workshop organized on March 4th and 5th at IRISA. Stefen Frenz was hosted during a week in the PARIS Project-Team in June 2004. In September 2004, Yvon Jégou, Renaud Lottiaux and Christine Morin were hosted for a two-day visit at Ulm University.

8.4.2. North-America

NSF/INRIA contract, Univ. New Hampshire, USA. This funding has supported our collaboration with the Parallel Computing group of the CS Department of Univ. of New Hampshire (Phil Hatcher and Bob Russell, Professors at UNH). The collaboration has focused on the *Hyperion* project, a distributed compiling and execution environment for clusters. David Noblet (one of Phil Hatcher's undergraduate students) has visited IRISA during the summer 2004 for a 2-month internship within the PARIS Project-Team. He worked on the evaluation of JXTA communication protocols. He has been supervised by Gabriel Antoniu and Mathieu Jan. The internship has been funded by the *International Research Opportunities Program* (IROP) de UNH (http://www.unh.edu/urop/discoverirop.html).

Rutgers University, USA. We collaborate with the *Discolab* Research Team leaded by Liviu Iftode at Rutgers University. Pascal Gallard has worked on the use of the R-DMA technology for the implementation of a highly-available cluster architecture. An *équipe associée* proposal for further collaboration on the design and implementation of a novel, highly-available cluster architecture based on the concept of *remote healing* has been submitted in November 2004.

MIT, Boston, USA. Chester Tse, 3rd year undergraduate student at MIT (Boston, Massachussets, USA) has spent a 3-month internship within the PARIS Project-Team, from June to August 2004. He has been supervised by Gabriel Antoniu and Mathieu Jan. He designed and implemented a graphical visualization tool for the JuxMem platform. The internship was co-funded by the MIT-France program and by INRIA.

8.4.3. Middle-East, Asia, Oceania

Seoul National University, Korea. The PARIS Project-Team, with the GRAAL Project-Team located at INRIA Rhône-Alpes, have been selected by the STAR program of the French Embassy in Seoul to conduct a 2-year cooperation with the Department of Aerospace Engineering (Prof. Seung Jo Kim) of the Seoul National University. This cooperation, starting in June 2003, aims at experimenting a Grid infrastructure, made with the computing equipments of the two participants, with aerospace applications (SNU) and middleware and programming tools designed by INRIA. In July 2004, Prof. Seung Jo Kim visited IRISA with two researchers from his team. A workshop was organized to discuss scientific and technical progress on both side. A group of six INRIA researchers visited SNU with the objective of setting up a Grid infrastructure made of computing resources from INRIA and SNU. During the visit, two applications from SNU were executed using the grid infrastructure and middleware provided by both the PARIS and GRAAL Project-Teams.

8.5. Visits and invitations

University of New Hampshire, USA. Phil Hatcher, Professor at the University of New Hampshire, visited the PARIS Project-Team in June 2004 and presented his latest work on bio-informatics within a seminar organized by the PARIS Research Group.

University of California, Irvine, USA. Isaac Scherson, Professor at the UCI University, USA, was invited by the PARIS Project-Team in January 2004 and presented two seminars: *The Rate of Change Load Balancing in Cluster Computing*, on January 12th 2004 and *Measuring beyond FLOPS: A seminar in Performance Evaluation*, on January 13, 2004. Isaac Scherson has been hosted by the PARIS Project-Team during five months (March - July 2004) funded by a one-month grant from the University of Rennes 1 and a 4-month grant from IRISA/INRIA. He contributed to the KERRIGHED and Grid-aware OS research activities.

Oak Ridge National Laboratory (ORNL), USA. Stephen Scott is a Senior Researcher at Oak Ridge National Laboratory. He was invited by the PARIS project in July 2004. He visited us in the framework of our on-going collaboration between INRIA, EDF R&D and ORNL for the integration of KERRIGHED into OSCAR (http://oscar.openclustergroup.org/), a distribution for high performance computing on clusters.

Universidade Catolica de Santos, Brazil Fabricio Silva, professor at the Universidade Catolica de Santos, Brazil, was invited by the PARIS Project-Team on May 18th and 19th. He gave a seminar entitled *The DM-Grid Project: Running Data-mining Applications on Computational Grids* on May 18th.

Karlsruhe University,Germany Florin Isaila, research assistant at the University of Karlsruhe, Germany, was invited by the PARIS Project-Team on November 29th and 30th. He gave a talk entitled *ParadisNet: A Network Interface for Parallel and Distributed Systems Development*.

9. Dissemination

9.1. Community animation

9.1.1. Leaderships, Steering Committees and community service

European COREGRID Network of Excellence. Th. Priol is the *Scientific Coordinator* of the COREGRID Network of Excellence (http://www.coregrid.net/). This network started in September 1, 2004. Ch. Pérez is the INRIA Scientific Correspondent of COREGRID NoE.

ACI GRID, Ministry of Research. Th. Priol has been appointed in January 2004 as new director of ACI GRID Program, funded by the French National Ministry of Research. The ACI GRID is the national French initiative in the area of Grid computing (http://www.recherche.gouv.fr/recherche/aci/grid.htm). It was formerly led by Michel Cosnard, INRIA Sophia. This initiative was launched in April 2001. Since then, several call for proposals (one per year) were issued, and evaluation was carried out by the Scientific Committee. L. Bougé is member of the Scientific Committee if this program.

National GRID 5000 Project. The ACI GRID Program has launched the GRID 5000 Project in order to build a national Grid infrastructure for research in Computer Science. The objective is to set up a constellation of large clusters in 8 major research laboratories throughout the country, amounting altogether to 5,000 processors, interconnected by a high-performance large-area network. As the chairman of the ACI GRID Program, Th. Priol is member of the steering committee. Y. Jégou is member of the steering committee as a representative of IRISA within this initiative.

CNRS, GDR ARP. L. Bougé chairs the CNRS Research Co-operative Federation (*Groupement de recherche*, GDR) on Architecture, Networks and Systems, and Parallelism (ARP, http://www.arp.cnrs.fr/). He has been serving since Year 2000. This GDR GDR is run by the STIC CNRS Department. It is one of the 6 nation-wide animation networks (*GDR d'animation*) run by the Department. It has been renewed for another 4-year term in 2002. Virtually all the French academic researchers active in these areas are registered in the GDR. As of today, this amounts to ca. 900 persons.
J.-L. Pazat is the coordinator of the G2C (*Grids and Clusters for Computing*) Working Group of GDR ARP. This working group aims at information dissemination and contacts between researchers in the area of Cluster and Grid computing.

CNRS, Inter-GDR Co-ordination Committee. L. Bougé chairs the (informal!) Co-ordination Committee of the 6 GDR of the CNRS STIC Department. He has been serving since Year 2001.

Euro-Par Annual Conference. L. Bougé serves as the Vice-Chair of the *Steering Committee* of the *Euro-Par* annual conference series on parallel computing (ca. 250 attendees, http://www.europar.org/).

RenPar Annual Conference. J.-L. Pazat serves as the Chair of the Steering Committee of the RenPar (*Rencontres francophone du parallélisme*, http://www.renpar.org/) annual conference series. The last edition of RenPar was held in La Colle-sur-Loup, near Nice, in 2003.

IEEE IPDPS Conference Series. L. Bougé is a member of the *Steering Committee* of the IPDPS (*International Parallel and Distributed Processing Symposium*, http://www.ipdps.org/) annual conference series. Luc Bougé served as a *Workshop Co-Chair* for IPDPS 2004, held in Santa-Fe, NM, in April 2004.

ACI GRID GRID2. J.-L. Pazat heads the GRID2 Project http://www.irisa.fr/grid2/ of ACI GRID. This project is devoted to dissemination and co-ordination of academic French research groups interested in Grid computing.
Ch. Pérez co-ordinates of the *Communication and middleware systems* Working Group in the GRID2 project.

ACM ICS 2004 Conference. Several members of the PARIS Project-Team have been involved in the Organization Committee of the *18th ACM International Conference on Supercomputing* (ICS '04). It has been held in Saint-Malo, France, in June 2004, co-supported by IRISA/INRIA and ENS Lyon. Ch. Morin held the *Local Arrangement Co-Chair*, L. Bougé the *Finance Chair*, Ch. Pérez the *Publication Chair*, and Th. Priol the *Workshop Chair*. About 110 participants attended (http://graal.ens-lyon.fr/ICS04/).

DRUIDE 2004 Thematic School. Several members of the PARIS Project-Team have been involved in organization of the DRUIDE 2004 Thematic School on Large-Scale Distributed Data, held in Le Croisic, in May 2004 (http://druide2004.irisa.fr/). This school was supported by CNRS, INRIA and the GDR ARP. G. Antoniu chaired the Organization Committee and the Program Committee. L. Bougé has been a member of the Organization committee. Th. Priol and Ch. Pérez participated to the Program Committee. 85 participants attended this school.

COSET-1 Workshop. Ch. Morin co-organized with Stephen Scott, ORNL, the *First International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters*. It was held in Saint-Malo in June 2004, in conjunction with ICS 2004. About 20 researchers attended the workshop (http://www.irisa.fr/manifestations/2004/coset/).

European IST POP. Ch. Pérez is the INRIA Scientific Correspondent of the European IST POP project, which started in December 2001 for 3 years.

European SSA GRIDCOORD. L. Bougé and Th. Priol participate to the GRIDCOORD *Specific Support Action* (SSA) through the INRIA institutional member. Isabelle Attalli, OASIS Project-Team, INRIA Sophia-Antipolis, is in charge of leading the contribution of INRIA members to this SSA, in close co-ordination with the COREGRID NoE. The GRIDCOORD SSA has been launched on July 2004, for 18 months (http://www.gridcoord.org/).

ACI GRID/INRIA Alta. Ch. Pérez heads the Alta Project, co-supported by ACI GRID and INRIA. Alta started in 2003, for 2 years (http://www.irisa.fr/alta/).

ACI MD GDS. G. Antoniu heads the GDS (*Grid Data Service*) Project supported by ACI MD. GDS started in September 2003, for 3 years (http://www.irisa.fr/GDS/).

ACI MD GdX. G. Antoniu is the local correspondent of the GdX (*Data Grid Explorer*) Project supported by ACI MD. GdX started in September 2003, for 3 years (http://www.lri.fr/~fci/GdX/).

RTP STIC CNRS. L. Bougé was a member of the Steering Committee of CNRS Thematic Committee (RTP 8) *High-Performance and Distributed Computing* led by Yves Robert, Lyon, and Brigitte Plateau, Grenoble. This RTP has been stopped by the end of 2004.

AS STIC CNRS. Ch. Pérez was the local correspondent of CNRS Specific Action (AS 115) of RTP 8 *Methodology of Grid programming*, let by Raymond Namyst, Bordeaux. This RTP has been stopped by the end of 2004.

9.1.2. Editorial boards, steering and program committees

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

Cluster 2004: San Diego, USA, September 2004.

CCGrid 2005: *IEEE/ACM International Symposium on Cluster Computing and the Grid*, to be held in May 2005, in Cardiff, UK.

J.-P. Banâtre was Co-chairman of the EU/NSF workshop *Unconventional Programming Paradigms*, in Le Mont Saint Michel, September, 2004.
He served in the Program Committees for the following conferences:

ISORC 2004 7th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, held in May 2004, in Vienna, Austria

SRDS 2004: *23rd International Symposium on Reliable Distributed Systems*, held in May 2004, in Florianpolis, USA.

L. Bougé belongs to the *Editorial Advisory Board* of the *Scientific Programming* Journal, IOS Press.
He chairs the Program Committee of the *International Conference on High Performance Computing* (HiPC 2004), to be held in Bangalore, India, in December 2004.

A.-M. Kermarrec is the Global Chair Topic *Peer-to-peer and Web Computing* of Euro-Par 2005, to be held in Lisbon, Portugal in August 2005.
She co-organized with Pierre Sens and Luciana Arantes (LIP6) a one-day workshop on *Peer-to-peer computing* (*Journée Thème Emergent*, JTE) in the context of the French Chapter of ACM SIGOPS, held in December 2004.
She co-organized with Pierre Fraigniaud, the peer-to-peer track of the workshop on Distributed Algorithms, held in Porquerolles in September 2004.
She served in the Program Committees for the following conferences:

Algotel 2004: *6es Rencontres francophones sur les aspects algorithmiques des télécommunications*, held in Batz-sur-mer, France, May 2004.

DOA 2004: *Distributed Objects and Applications*, held in Agia Napa, Cyprus, October 2004.

MediaNet 2004: *Second International Conference on Intelligent Access of Multimedia Documents on Internet*, held in Tunisia, November 2004.

HiPC 2004: *International Conference on High performance Computing*, held in Bangalore, India, December 2004.

IPTPS '05: *International workshop on peer-to-peer Computing*, to be held in Ithaca, NY, USA, February 2005.

ICDCS '05: *International Conference on Distributed Computing Systems* , Peer-to-Peer Networking Track to be held in Colombus, USA, May 2005.

CFSE 2004: *4e Conférence Française sur les Systèmes d'Exploitation*, to be held in Le Croisic, Presqu'île de Guérande, France, April 2005.

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

DSM 2004: *International Workshop on Distributed Shared Memory on Clusters*, organized in conjunction with IEEE International Symposium on Cluster Computing and the Grid (CCGrid '04), Chicago, IL, USA, May 2004.

ICS '04: *18th ACM International Conference on Supercomputing*, Saint-Malo, France, June 2004.

1st International workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters, Saint-Malo, France, June 2004.

ICWL 2004: 2nd International Conference on Web-Based Learning, Tsinghua University, Beijing, China, August 2004.

RenPar 16: 16e Rencontres francophones du parallélisme, Le Croisic, Presqu'île de Guérande, France, April 2005.

DSM 2005: *International Workshop on Distributed Shared Memory on Clusters*, organized in conjunction with *IEEE International Symposium on Cluster Computing and the Grid* (CCGrid '05), Cardiff, UK, May 2005.

ICA3PP-2005: *6th International Conference on Algorithms and Architectures*, Melbourne, Australia, June 2005.

EGC 2005: *European Grid Conference 2005*, Amsterdam, The Netherlands, February 2005.

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

ACM Workshop on Component Models and Systems for Grid Applications, Saint-Malo, France, June 2004, in conjunction with ICS 2004.

European Grid Conference, to be held in February 2005, Science Park Amsterdam, The Netherlands.

HIPS 2005: *10th International Workshop on High-Level Parallel Programming Models and Supportive Environment*, to be held in April 2005, Denver, Colorado, USA.

Th. Priol was the Global Chair of Topic *Grid and Cluster Computing* of the Euro-Par 2004 Conference that was held in Pisa, Italy, in August 2004.
Th. Priol is a member of the Editorial Board of the *Parallel Computing* journal.
He served in the Program Committees of the following conferences:

2nd European Across Grids Conference , Nicosia, Cyprus, January 2004.

CCGRID 2004: *IEEE International Symposium on Cluster Computing and the Grid*, Chicago, IL, USA, April 2004.

VecPar 2004: *International Meeting on High Performance Computing for Computational Science*, Valencia, Spain, June 2004.

Fifth EuroGraphics Workshop on Parallel Graphics and Visualization, Grenoble, France, June 2004.

Third Workshop on Advanced Collaborative Environments, Seattle, WA, USA, June 2004

First International Workshop on Programming Paradigms for Grids and Metacomputing Systems, Krakow, Poland, June 2004.

5th Austrian-Hungarian Workshop on Distributed and Parallel Systems, Budapest, Hungary, September 2004.

SCC04 IEEE: *International Conference on Services Computing*, Shanghai, China, September 2004.

5th IEEE/ACM Intl. Workshop on Grid Computing, Pittsburgh, USA, November 2004.

Workshop on Access to Knowledge through Grid in a Mobile World, Vienna, Austria, December 2004.

European Grid Conference , Amsterdam, The Netherlands, February 2005.

2nd Workshop on Programming Grids and Metasystems, Atlanta, USA, May 2005.

HPDC-14: *14th IEEE International Symposium on High-Performance Distributed Computing*, Research Triangle Park, USA, July 2005.

9.1.3. Evaluation committees, consulting

G. Antoniu has served as a reviewer for the evaluation of a NSERC (Canada) project submission.

L. Bougé served in the *Evaluation Committee* of the RNTL Program till September 2004.
He serves in the *Scientific Committees* of ACI GRID and ACI MD Programs of the Ministry of Research.
He served as an international expert of the Evaluation Committee of the Dutch ASCI Research School (*Advanced School for Computing and Imaging*), let by Andy Tanenbaum. The Evaluation was held in September 2004 on behalf of the Royal Netherlands Academy of Arts and Sciences (KNAW).

A.-M. Kermarrec is a member of the committee *Prix de thèse Spécif*, 2004.
She served as a reviewer for the Council of Physical Sciences of the Netherlands Organization for Scientific research (NWO).
She served as a reviewer for the Swedish scientific council.
She served as a reviewer of the MMAPPS EC-funded projects.
She acted as a consultant for Microsoft Research, Cambridge, UK.

J.-L. Pazat is a member of the Selection Committee (*Commission de spécialistes*, CSE) for Computer Science at INSA Rennes and University of South Brittany (UBS, Vannes).

Ch. Pérez has served as an external expert for the ACI MD programs of Ministry of Research.

Th. Priol has been member of an *Expert Group* convened by the DG Information Society of the European Commission (EC) to outline a vision for Grid research priorities over the period of 2005-2010. (ftp://ftp.cordis.lu/pub/ist/docs/ngg2_eg_final.pdf).
He served as a reviewer to the following EC-funded projects: IST EUROGRID and P2PEOPLE.
He is member of the Scientific Committee of the PRST Intelligence Logicielle (Contrat de Plan Etat-Région Lorraine 2003-006).
He is member of the Evaluation Committee of the LINA (Nantes).

9.2. Academic teaching

G. Antoniu has taught the tutorials of the *Operating Systems* module of the *DESS CCI* Master Program (IFSIC). He 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.

L. Bougé 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.

A.-M. Kermarrec is responsible for a graduate teaching module *peer-to-peer systems and applications* of the Master Program in Computer Science,UNIVERSITY RENNES 1.

Ch. Morin is responsible for a graduate teaching module *High Performance Computing on Clusters and Grids* of the Master Program, UNIVERSITY RENNES 1. Within this module, she gave lectures on distributed operating systems for clusters.
She gave a lecture on clusters, taught in the final year of the *Network Architecture* Track at the Institut National des Télécommunications (INT) in Évry in December 2004.

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. Within this module, he gave lectures on Enterprise Java Beans.

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*.

Th. Priol gave lectures on Distributed Shared Memory within the *High Performance Computing on Clusters and Grids* Module of the Master Program, UNIVERSITY RENNES 1.

9.3. Conferences, seminars, and invitations

Only the events not listed elsewhere are listed below.

RNTL Information Day. J.-P. Banâtre gave an invited presentation entitled *Future Programming Techniques* in Rennes, October 2004.

ACI Securité Information Day. J.-P. Banâtre gave an invited presentation entitled *Security in the next EU Framework Programme* in Toulouse, November 2004.

FSR 2003. Ch. Morin was invited to participate to the *Free Software Research* workshop (FSR '03) in Soissons, in December 2003. She presented a talk entitled *Experience with Kerrighed free software*.

Bull. Ch. Morin and D. Margery were invited to present KERRIGHED at Bull, Echirolles, in February 2004.

GGF, CPR-WG Ch. Morin was invited to participate to the *Checkpoint recovery working group* meeting at the Global Grid Forum meeting, held in Berlin, Germany, in March 2004. She presented an invited talk entitled *Overview of* PARIS *project-team Activities in Checkpoint Recovery*.

Deakin University, Australia. Ch. Morin gave an invited talk in the Computer Science Department, at Deakin University, in April 2004. Title: *Kerrighed: a Genuine Single System Image Cluster Operating System based on Linux*.

COSET-1 Workshop. G. Vallée gave an invited talk at the COSET-1 Workshop organized in conjunction with ICS '04 in Saint-Malo, France, in June 2004. Title: *SSI-OSCAR: A Single System Image for OSCAR Clusters*.

Workshop France-Korea. Ch. Morin presented a talk entitled *The Kerrighed SSI Operating System* at the France-Korea workshop organized at IRISA in Rennes, France in July 2004.

University of Tennessee, USA. G. Vallée gave a talk entitled *The Kerrighed Operating System: a Single System Image for Cluster* for students at the University of Tennessee, UT, Knoxville, Tennessee, USA, in September 2004.

ORNL, USA. G. Vallée gave a talk at Oak Ridge National Laboratory, USA, in October 2004. Title: *Kerrighed: A single system image for clusters*. Ch. Morin gave an invited talk entitled *Beyond Kerrighed: research perspectives in cluster computing*. R. Lottiaux gave two talks: *penMosix, OpenSSI and Kerrighed: A comparative study* and *KerFS: Kerrighed Distributed File System* at ORNL in November 2004.

EDF R&D, Clamart. G. Vallée gave a talk entitled *Conception d'un ordonnanceur de processus adaptable pour la gestion globale des ressources dans les grappes de calculateurs* at EDF R&D in Clamart, France, in March 2004.

Dagstuhl Seminar. Th. Priol, L. Bougé, G. Antoniu and C. Pérez were invited and gave talks at the Dagstuhl Seminar on Future Generation Grids - FGG 2004 in November 2004.

Workshop on Distributed Algorithmics, Porquerolles. G. Antoniu was invited to give a talk at the *Workshop on Distributed Algorithmics* held in Porquerolles, in September 2004. Title: *JuxMem: How to Handle Fault-tolerance and Data Consistency in a Grid Data-sharing Service?*.

GiGn '04. Th. Priol gave a invited presentation on Grid research activities in France and in Europe during the GiGn'2004 conference, Clermont-Ferrand, January 2004.

SwissGRID. Th. Priol gave two presentations (French ACI GRID and the COREGRID NoE) at the SwissGRID workshop, Lugano, Switzerland, April 2004.

AFNeT '04. Th. Priol gave an invited talk to the AFNeT congress in Paris, France, April 2004.

Café des Techniques au Musée des Arts et Métiers. Th. Priol gave a presentation on Grids in Paris, November 2004.

CCGSC '04. Th. Priol gave an invited talk entitled *Objects, Components, Services for grid middleware: pros & cons* at the CCGSC '04 workshop in Chateau de Faverges organized by Jack Dongarra, September 2004.

Joint EDF/INRIA/CEA Summer School on Model Coupling and Code Coupling Tools. Th. Priol gave a presentation entitled *Une approche par composant logiciels pour le couplage de codes* in Centre de Séminaires Port-Royal, June 2004.

Sino-French Workshop. Th. Priol gave a presentation related to the COREGRID Network of Excellence during a Sino-French Workshop organized by the French Embassy in Beijing, China, June 2004.

EGC'05. Th. Priol will be one of the keynote speaker of the European Grid Conference to be held in Amsterdam, The Netherlands, February 2005.

Workshop Scheduling for large scale distributed platforms. Aussois, France. Ch. Pérez gave an invited talk entitled *On the deployment and the execution of component applications on the Grid* at the Workshop *Scheduling for large scale distributed platforms* in Aussois, organized by the ENS Lyon.

IRIT Seminar. Ch. Pérez gave an invited talk entitled at a seminar organized by the Institut de Recherche en Informatique de Toulouse (IRIT), Toulouse, France, December 2004.

France/Japan Workshop on Grid. Ch. Pérez gave a talk presenting the result of the ACI GRID RMI and the ARC/ACI GRID ALTA at the French/Japan Workshop on Grid Computing at CNRS headquarters, Paris, March 2004.

Microsoft Research, Cambridge, UK. A.-M. Kermarrec has been invited for a 3-day visit at the Microsoft Research Lab by Laurent Massoulié.

ENS CACHAN. Anne-Marie Kermarrec gave a seminar at the Computer Science Department of ENS CACHAN on *Application-level multicast in large scale distributed systems* in September 2004.

ACI MD Pair-à-pair. Anne-Marie Kermarrec gave a seminar on *SplitStream: High bandwidth multicast in Cooperative Environnements* at the *ACI Masses de données* Project *Pair-à-pair* Workhop in October 2004.

9.4. Administrative responsibilities

J.-P. Banâtre is in charge of the European Affairs within the Department for European and International Relations (DREI) at INRIA.

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.

Ch. Morin is a member of the INRIA Evaluation Committee. She was a member of the 2004 selection committee for the Junior Researcher permanent position (CR2) at the INRIA Futurs research unit, and of the 2004 selection committee for the Senior Researcher permanent position (DR2).
She chairs the local IRISA Computing Infrastructure User Committee (*Commission des utilisateurs des moyens informatiques,CUMI*).

J.-L. Pazat is a member of the Administrative Committee of INSA of Rennes.

A.-M. Kermarrec is co-responsible of the local IRISA International Relations committee (INRIA).

9.5. Miscellaneous

L. Bougé is a member of the Project-Team Committee of IRISA, standing for the ENS CACHAN partner. He serves as the Vice-Chairman of the Selection Committee (*Commission de spécialistes*, CSE) for Computer Science at ENS CACHAN, and as an external deputy-member of the one at UNIVERSITY RENNES 1.

A.-M. Kermarrec is a member of the local IRISA Communication Committee.
She is a member of the Selection Committee (*Commission de Spécialistes*) for computer Science of ENS CACHAN.
She is a member of the working group *Prospective* of the INRIA *Conseil d'Orientation Scientifique et Technologique*.

Ch. Morin is a member of the editorial board of *Inedit*, the INRIA Newsletter.
She is an external member of the *Course Advisory Board* of the Information Technology School of Deakin University (Australia). She participated to the *Course Advisory Board* annual meeting, in Melbourne, Australia in April 2004.

Th. Priol is a deputy-member of the Selection Committee (*Commission de spécialistes*, CSE) for Computer Science at UNIVERSITY RENNES 1, since December 2001.

Ch. Pérez is member of the Selection Committee (*Commission de spécialistes*, CSE) for Computer Science at ENS CACHAN.
He is a member of the IRISA Committee (*Conseil de laboratoire* since March 2004.

10. Bibliography

Major publications by the team in recent years

[1] 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.

[2] 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://www.inria.fr/rrrt/rr-4108.html.

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

[4] 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://www.inria.fr/rrrt/rr-4554.html.

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

[6] 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, http://www.irisa.fr/paris/biblio/Papers/Bouge/KieHatBouBal01CACM.ps.gz.

[7] 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://www.inria.fr/rrrt/rr-1634.html.

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

Books and Monographs

[9] J.-P. BANÂTRE, J.-L. GIAVITTO, P. FRADET, O. MICHEL (editors). *Unconventional Programming Paradigms (UPP '04), pre-proceedings*, Available on request, September 2004.

[10] L. BOUGÉ, V. K. PRASANNA (editors). *Proc. 11th Intl. Conf. on High Performance Computing (HiPC 2004)*, Lect. Notes in Computer Science, vol. 3296, Springer-Verlag, Bangalore, India, December 2004.

Doctoral dissertations and Habilitation theses

[11] P. GALLARD. *Conception d'un service de communication pour systèmes d'exploitation distribués pour grappes de calculateurs: mise en oeuvre dans le système à image unique Kerrighed*, Thèse de doctorat, IRISA, Université de Rennes 1, IRISA, Rennes, France, December 2004, http://www.irisa.fr/paris/Biblio/Papers/Gallard/Gal04PhD.pdf.

[12] A. RIBES. *Contribution à la conception d'un modèle de programmation parallèle et distribué et sa mise en oeuvre au sein de plates-formes orientées objet et composant*, Thèse de doctorat, IRISA, Université de Rennes 1, IRISA, Rennes, France, December 2004, http://www.irisa.fr/paris/Biblio/Papers/Ribes/Rib04Phd.ps.

[13] G. VALLÉE. *Conception d'un ordonnanceur de processus adaptable pour la gestion globale des ressources dans les grappes de calculateurs : mise en oeuvre dans le système d'exploitation Kerrighed*, Thèse de doctorat, IFSIC, Université de Rennes 1, France, March 2004, http://www.irisa.fr/bibli/publi/theses/2004/vallee/vallee.html.

Articles in referred journals and book chapters

[14] G. ANTONIU, L. BOUGÉ, M. JAN. *JuxMem: An Adaptive Supportive Platform for Data Sharing on the Grid*, in "Journal of Parallel and Distributed Computing Practices", To appear. Special issue on the Workshop on Adaptive Grid Middleware (AGridM 2003), New Orleans, Louisiana, September 2003. Preliminary electronic version available as INRIA Research Report RR-4917, 2004, http://www.inria.fr/rrrt/rr-4917.html.

[15] G. ANTONIU, L. BOUGÉ, M. JAN. *JuxMem: Weaving together the P2P and DSM paradigms to enable a Grid Data-sharing Service*, in "Kluwer Journal of Supercomputing", To appear. Preliminary electronic version available as INRIA Research Report RR-5082, 2004, http://www.inria.fr/rrrt/rr-5082.html.

[16] P. EUGSTER, R. GUERRAOUI, A.-M. KERMARREC, L. MASSOULIÉ. *From Epidemics to Distributed Computing*, in "IEEE Computer", vol. 37, n° 5, May 2004, p. 60–67, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/EugGueKerMas04IEEEComp.pdf.

[17] C. MORIN, P. GALLARD, R. LOTTIAUX, G. VALLÉE. *Towards an Efficient Single System Image Cluster Operating System*, in "Future Generation Computer Systems", vol. 20, n° 2, January 2004, http://www.irisa.fr/paris/Biblio/Papers/Morin/MorGalLotVal03FGCS.pdf.

Publications in Conferences and Workshops

[18] F. ANDRÉ, J. BUISSON, J.-L. PAZAT. *Dynamic adaptation of parallel codes: toward self-adaptable components for the Grid*, in "ICS'04 Workshop on Component Models and Systems for Grid Applications", June 2004, http://www.irisa.fr/paris/Biblio/Papers/Buisson/AndBuiPaz04CMSGA.pdf.

[19] G. ANTONIU, L. BOUGÉ, M. JAN, S. MONNET. *Large-scale Deployment in P2P Experiments Using the JXTA Distributed Framework*, in "Euro-Par 2004: Parallel Processing, Pisa, Italy", Lect. Notes in Comp. Science, n° 3149, Springer-Verlag, August 2004, p. 1038–1047, http://www.irisa.fr/paris/Biblio/Papers/Antoniu/AntBouJanMon04EuroPar.ps.gz.

[20] G. ANTONIU, J.-F. DEVERGE, S. MONNET. *Building Fault-Tolerant Consistency Protocols for an Adaptive*

Grid Data-Sharing Service, in "Proc. ACM Workshop on Adaptive Grid Middleware (AGridM 2004), Antibes Juan-les-Pins, France", To appear, September 2004, http://www.inria.fr/rrrt/rr-5309.html.

[21] R. BADRINATH, C. MORIN. *Locks and Barriers in Checkpointing and Recovery*, in "Proc. Intl. Workshop on Distributed Shared Memory on Clusters (DSM 2004), Chicago", Held in conjunction with CCGrid 2004. IEEE TFCC, April 2004, p. 471–477, http://www.irisa.fr/paris/Biblio/Papers/Badrinath/BadMor04DSM2004.pdf.

[22] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Chemical Specification of Autonomic Systems*, in "Proceedings of the 13th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE'04)", July 2004, http://www.irisa.fr/paris/Biblio/Papers/Banatre/BanFraRad04IASSE.pdf.

[23] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Higher-order Chemical Programming Style*, in "Proceedings of Unconventional Programming Paradigms (UPP'04)", An extended version will be published by Springer in spring 2005, September 2004, http://www.irisa.fr/paris/Biblio/Papers/Banatre/BanFraRad04UPP.pdf.

[24] A. BOHRA, I. NEAMTIU, P. GALLARD, F. SULTAN, L. IFTODE. *Remote Repair of Operating System State Using Backdoors*, in "International Conference on Autonomic Computing (ICAC-04), New-York, NY", Initial version published as Technical Report, Rutgers University DCS-TR-543, May 2004, http://discolab.rutgers.edu/bda/remrepair04.ps.

[25] J. BUISSON, F. ANDRÉ, J.-L. PAZAT. *Adaptation dynamique de codes parallèles*, in "Journée Composants 2004", March 2004, http://www.irisa.fr/paris/Biblio/Papers/Buisson/BuiAndPaz04JC.ps.

[26] A. DENIS, C. PÉREZ, T. PRIOL. *Network Communications in Grid Computing: At a Crossroads Between Parallel and Distributed Worlds*, in "18th International Parallel and Distributed Processing Symposium (IPDPS 2004), Santa Fe, NM, USA", IEEE Computer Society, April 2004, 95a, http://www.irisa.fr/paris/Biblio/Papers/Denis/DenPerPri04IPDPS.pdf.

[27] A. DENIS, C. PÉREZ, T. PRIOL, A. RIBES. *Bringing High Performance to the CORBA Component Model*, in "SIAM Conference on Parallel Processing for Scientific Computing", February 2004, http://www.irisa.fr/paris/Biblio/Papers/Perez/DenPerPriRib04PP.txt.

[28] S. HANDURUKANDE, A.-M. KERMARREC, F. LE FESSANT, L. MASSOULIÉ. *Exploiting Semantic Clustering in the eDonkey P2P network*, in "SIGOPS European Workshop, Leuven, Belgium", September 2004, p. 109–114, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/HanKerLefMas04EWSIGOPS.pdf.

[29] M. JELASITY, R. GUERRAOUI, A.-M. KERMARREC, M. VAN STEEN. *The Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations*, in "ACM/IFIP/USENIX 5th International Middleware Conference (Middleware), Toronto, Canada", October 2004, p. 79-98, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/JelGueKerSte04MIDDLEWARE.pdf.

[30] A.-M. KERMARREC. *Self-clustering in Peer-to-Peer overlays*, in "International Workshop on Self-* Properties in Complex Information Systems, Bertinoro, Italy", February 2004, p. 89–92, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/Ker04SELF.pdf.

[31] S. LACOUR, C. PÉREZ, T. PRIOL. *A Network Topology Description Model for Grid Application Deployment*, in "Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (GRID 2004), Pittsburgh,

PA, USA", R. BUYYA (editor)., Held in conjunction with Supercomputing 2004 (SC2004), November 2004, p. 61–68, http://www.irisa.fr/paris/pages-perso/Sebastien-Lacour/publis/papers/LacPerPri2004grid.pdf.

[32] S. LACOUR, C. PÉREZ, T. PRIOL. *A Software Architecture for Automatic Deployment of CORBA Components Using Grid Technologies*, in "Proceedings of the 1st Francophone Conference On Software Deployment and (Re)Configuration (DECOR 2004), Grenoble, France", October 2004, p. 187–192, http://www.irisa.fr/paris/pages-perso/Sebastien-Lacour/publis/papers/LacPerPri2004decor.pdf.

[33] S. LACOUR, C. PÉREZ, T. PRIOL. *Deploying CORBA Components on a Computational Grid: General Principles and Early Experiments Using the Globus Toolkit*, in "Proceedings of the 2nd International Working Conference on Component Deployment (CD 2004), Edinburgh, Scotland, UK", W. EMMERICH, A. L. WOLF (editors)., Lect. Notes in Comp. Science, Held in conjunction with the 26th International Conference on Software Engineering (ICSE 2004), nº 3083, Springer-Verlag, May 2004, p. 35–49, http://www.irisa.fr/paris/pages-perso/Sebastien-Lacour/publis/papers/LacPerPri20040521CD.ps.

[34] F. LE FESSANT, S. HANDURUKANDE, A.-M. KERMARREC, L. MASSOULIÉ. *Clustering in Peer-to-Peer File Sharing Workloads*, in "3rd International Workshop on Peer-to-peer systems (IPTPS 04), San Diego, CA", February 2004, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/LefHanKerMas04IPTPS.pdf.

[35] S. MONNET, C. MORIN, R. BADRINATH. *A Hierarchical Checkpointing Protocol for Parallel Applications in Cluster Federations*, in "9th IEEE Workshop on Fault-Tolerant Parallel, Distributed and Network-Centric Systems, Santa Fe, New Mexico", IEEE, Held in conjunction with IPDPS 2004, April 2004, http://www.irisa.fr/paris/Biblio/Papers/Monnet/MonMorBad04FTPDS.pdf.

[36] S. MONNET, C. MORIN, R. BADRINATH. *Hybrid Checkpointing for Parallel Applications in Cluster Federations*, in "4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, IL, USA", Electronic version, IEEE, CCGrid 2004, April 2004, http://www.irisa.fr/paris/Biblio/Papers/Monnet/MonMorBad04CCGRID.pdf.

[37] C. MORIN, R. LOTTIAUX, G. VALLÉE, P. GALLARD, D. MARGERY, J.-Y. BERTHOU, I. SCHERSON. *Kerrighed and Data Parallelism: Cluster Computing on Single System Image Operating Systems*, in "Proc. of Cluster 2004", IEEE, September 2004, http://www.irisa.fr/paris/Biblio/Papers/Morin/MorLotVal04Cluster.pdf.

[38] C. PÉREZ, T. PRIOL, A. RIBES. *PaCO++: A parallel object model for high performance distributed systems*, in "Distributed Object and Component-based Software Systems Minitrack in the Software Technology Track of the 37th Hawaii International Conference on System Sciences (HICSS-37), Big Island, Hawaii, USA", IEEE Computer Society Press, January 2004, 274a, http://www.irisa.fr/paris/Biblio/Papers/Ribes/PerPriRib04HICSS.pdf.

[39] C. PÉREZ, A. RIBES, T. PRIOL. *Handling Exceptions Between Parallel Objects*, in "Proc. 10th Intl. Euro-Par Conference (EuroPar 04), Pisa, Italia", Lect. Notes in Comp. Science, nº 3149, Springer-Verlag, August 2004, p. 671–678, http://www.irisa.fr/paris/Biblio/Papers/Perez/PerRibPri04EuroPar.ps.

[40] G. VALLÉE, J.-Y. BERTHOU, P. GALLARD, R. LOTTIAUX, D. MARGERY, C. MORIN. *Kerrighed: a Single System Image Providing High Availability Capabilities to Applications*, in "High Availability and Performance Computing Workshop 2004 (HAPCW04), Santa Fe, New Mexico, USA", October 2004, http://www.irisa.fr/paris/Biblio/Papers/Vallee/ValBerGal04HAPCW.pdf.

[41] S. VOULGARIS, A.-M. KERMARREC, L. MASSOULIÉ, M. VAN STEEN. *Exploiting semantic proximity in peer-to-peer content searching*, in "10th International Workshop on Future Trends in Distributed Computing Systems (FTDCS 2004), China", May 2004, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/VouKerMasSte04ftdcs.pdf.

Internal Reports

[42] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Principles of Chemical Programming*, Research Report, n° AIB-2004-04, RWTH Aachen, June 2004, http://www.irisa.fr/paris/Biblio/Papers/Banatre/BanFraRad04RULE.pdf.

[43] J. BUISSON. *Un protocole de négociation multi-parties*, Technical report, n° PI-1659, IRISA, November 2004, http://www.irisa.fr/paris/Biblio/Papers/Buisson/Bui04PI1659.pdf.

[44] R. GUERRAOUI, S. HANDURUKANDE, A.-M. KERMARREC. *GosSkip: a Gossip-based Structured Overlay Network for Efficient Content-based Filtering*, Technical Report, n° IC/2004/95, EPFL, Lausanne, 2004, http://ic2.epfl.ch/publications/documents/IC_TECH_REPORT_200495.pdf.

[45] M. JAN, D. A. NOBLET. *Performance Evaluation of JXTA Communication layers*, Research Report, n° RR-5350, INRIA, IRISA, Rennes, France, October 2004, http://www.inria.fr/rrrt/rr-5350.html.

[46] A.-M. KERMARREC, L. MASSOULIÉ, A. GANESH. *Efficient application-level multicast on a network-aware self-organizing overlay*, Research Report, n° 1657, IRISA, 2004.

[47] R. LOTTIAUX, B. BOISSINOT, P. GALLARD, G. VALLÉE, C. MORIN. *OpenMosix, OpenSSI and Kerrighed: A Comparative Study*, Technical report, n° RR-5399, INRIA, November 2004, http://www.inria.fr/rrrt/rr-5399.html.

[48] D. MARGERY, R. LOTTIAUX, C. MORIN. *Capabilities for per Process Tuning of Distributed Operating Systems*, Rapport de Recherche, n° RR-5411, INRIA, IRISA, Rennes, France, December 2004, http://www.inria.fr/rrrt/rr-5411.html.

[49] L. RILLING, C. MORIN. *A Fault-Tolerant Transparent Data Sharing Service for the Grid*, Research Report, n° RR-5427, INRIA, IRISA, Rennes, France, December 2004, http://www.inria.fr/rrrt/rr-5427.html.

[50] F. SULTAN, A. BOHRA, P. GALLARD, S. SMALDONE, Y. PAN, B. NATH, L. IFTODE. *Citadel: Defensive Architectures for Computer Systems and Networks*, Submitted for publication, Technical report, n° DCS-TR-554, Rutgers University, March 2004, http://discolab.rutgers.edu/bda/DCS-TR-554.ps.

[51] J. D. TERESCO, J. E. FLAHERTY, S. B. BADEN, J. FAIK, S. LACOUR, M. PARASHAR, V. E. TAYLOR, C. A. VARELA. *Approaches to Architecture-Aware Parallel Scientific Computation*, Submitted to Proceedings of the 11th Conference on Parallel Processing for Scientific Computing of the Society for Industrial and Applied Mathematics (SIAM-PP2004): Frontiers of Scientific Computing, Research Report, n° CS-04-09, Williams College Department of Computer Science, Williamstown, MA, USA, 2004, http://www.cs.williams.edu/~terescoj/research/publications/pp04/pp04.pdf.

[52] G. UTARD, C. MORIN. *Conception et Mise en œuvre d'un système d'entrées/sorties efficaces pour grappe*, Rapport de recherche, n° 5416, INRIA, December 2004, http://www.inria.fr/rrrt/rr-5416.html.

[53] G. VALLÉE, J.-Y. BERTHOU, R. LOTTIAUX, D. MARGERY, C. MORIN. *Ghost process: a Sound Basis to Implement New Mechanisms for Global Process Management in Linux Clusters*, Research report, n° 1664, IRISA, IRISA, Rennes, France, December 2004, http://www.inria.fr/rrrt/rr-5476.html.

Miscellaneous

[54] B. BOISSINOT. *Systèmes à image unique pour grappes : une étude comparative*, Stage effectué sous la direction de Christine Morin, September 2004, http://www.irisa.fr/paris/Biblio/Papers/Boissinot/Boi04internship.pdf.

[55] J.-F. DEVERGE. *Cohérence et volatilité dans un service pair-à-pair de partage de données*, Rapport de stage de DEA, DEA d'informatique de l'IFSIC, Université de Rennes 1, France, June 2004, http://www.irisa.fr/paris/Biblio/Papers/Deverge/Dev04DEA.pdf.

[56] J. GHAFFOUR. *Sécurité dans les grilles de calcul*, In French, Rapport de stage de DEA, IFSIC, Université de Rennes 1, France, June 2004, http://www.irisa.fr/paris/Biblio/Papers/Ghaffour/Gha04Master.pdf.

[57] D. MARGERY, R. LOTTIAUX, C. MORIN. *Kerrighed : Manuel de référence V1.0*, May 2004, Deliverable COCA contract.

[58] C. PÉREZ. *A Component-Based Software Infrastructure for Grid Computing*, vol. 59, October 2004, http://www.ercim.org/publication/Ercim_News/enw59/perez.html, ERCIM News.

[59] E. RIVIERE. *Gestion dynamique d'applications parallèles à mémoire partagée au sein du système d'exploitation pour grappes de calculateurs Kerrighed*, Rapport de stage de DEA, DEA d'informatique, IFSIC, Université de Rennes 1, France, June 2004, http://www.irisa.fr/paris/Biblio/Papers/Riviere/Riv04Master.pdf.

[60] G. VALLÉE. *Creation of the SSI-OSCAR Distribution*, August 2004, Deliverable EDF contract.

Bibliography in notes

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

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

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

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

[65] 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.

[66] O. AUMAGE, L. BOUGÉ, A. DENIS, L. EYRAUD, J.-F. MÉHAUT, G. MERCIER, R. NAMYST, L. PRYLLI. *A Portable and Efficient Communication Library for High-Performance Cluster Computing (extended version)*, in "Cluster Computing", vol. 5, n° 1, January 2002, p. 43–54.

[67] R. BUYYA. *High Performance Cluster Computing: Architectures and Systems*, Prentice-Hall PTR, 1999.

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

[69] 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.

[70] 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.

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

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

[73] 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.

[74] 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.

[75] 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.

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

[77] 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.

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

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

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

[81] A. ORAM. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly, 2001.

[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] A. ROWSTRON, P. DRUSCHEL. *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*, in "IFIP/ACM Intl. Conf. on Distributed Systems Platforms (Middleware)", November 2001, p. 329–350.

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