INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team PARIS

Programming Parallel and Distributed Systems for Large Scale Numerical Simulation Applications

Rennes

THEME NUM

Activity Report

2005

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.*

Head of project-team
Thierry Priol [DR INRIA]

Administrative assistants
Maryse Auffray [TR INRIA]
Päivi Palosaari [CoreGRID INRIA]

Staff members Inria
Gabriel Antoniu [CR]
Yvon Jégou [CR]
Anne-Marie Kermarrec [DR]
David Margery [IR (à 50% dans le projet)]
Christine Morin [DR]
Christian Pérez [CR]

Staff members University Rennes 1
Françoise André [Professor]
Jean-Pierre Banâtre [Professor]
Pascal Morillon [IE (since December 2005)]

Staff members Insa de Rennes
Marin Bertier [Assistant professor (since September 2005)]
Jean-Louis Pazat [Professor]

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

Project technical staff
Pascal Gallard [INRIA, DGA *COCA* Contract (till November 2005)]
Vincent Lefevre [IE INRIA (May–June 2005), IE UNIVERSITY RENNES 1 (July–November 2005)]
Renaud Lottiaux [INRIA, DGA *COCA* Contract (till November 2005)]
Guillaume Mornet [INRIA, PRIR Brittany Regional Council]
Jean Parpaillon [IA INRIA (since October 2005)]

PhD students
Yann Busnel [ENS CACHAN student (since October 2005)]
Loïc Cudennec [INRIA and Brittany Regional Council Grant (since October 2005)]
Mathieu Fertré [MENRT Grant (since October 2005)]
Lilia Hinde Bouziane [INRIA Grant]
Jérémy Buisson [MENRT Grant]
Mathieu Jan [INRIA and Brittany Regional Council Grant]
Emmanuel Jeanvoine [Cifre EDF industrial Grant]
Sébastien Lacour [INRIA Grant]
Erwan Le Merrer [Cifre France Telecom industrial Grant]
Sébastien Monnet [MENRT Grant]
Yann Radenac [MENRT Grant)]
Étienne Rivière [MENRT Grant]
Louis Rilling [MENRT Grant]
Gaël Utard [INRIA Grant (until March 2005)]

Post-doctoral fellow

Zsolt Nemeth [ERCIM Post-Doc (until March 2005)]
André Ribes [INRIA Post-Doc Engineer (until december 2005)]
Geoffroy Vallée [INRIA Post-Doc Engineer (until September 2005))]
Aline C. Viana [INRIA Post-Doc (since October 2005)]

2. Overall Objectives

2.1. General objectives

The PARIS Project-Team aims at contributing to the programming of parallel and distributed systems for large scale numerical simulation applications. Its goal is to design operating systems and middleware to ease the use of such computing infrastructure for the targeted applications. Such applications enable the speed-up of the design of complex manufactured products, such as cars or aircrafts, thanks to numerical simulation techniques. As computer performance 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 simulations of these different aspects will not be carried out by a single computer due to the lack of computing and memory resources. Instead, several clusters of inexpensive PCs, and probably 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 managing a national grid computing infrastructure (GRID 5000) enabling large-scale experiments.

2.1.1. Parallel processing to go faster

Given the significant increase of the performance of microprocessors, computer architectures and networks, clusters of standard personal computers now provide the level of performance to make numerical simulation a handy tool. This tool should not be used 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.

2.1.2.1. Large-scale numerical simulation

Large scale numerical simulation will without a doubt become one of the key technologies to meet such constraints. In traditional numerical simulation, only one simulation code is executed. In contrast, it is now required to *couple* several such codes together in a single simulation. A large-scale numerical simulation application is typically composed of several codes, not only to simulate one physics, but to perform multi-physics simulation. One can imagine that the simulation times will be in the order of weeks and sometimes months depending on the number of physics involved in the simulation, and depending on the available computing resources. Parallel processing extends the number of computing resources locally: it cannot significantly reduce simulation times, since the simulation codes will not be localized in a single geographical location. This is particularly true with the global economy where complex products (such as cars, aircrafts, etc.) are not designed by a single company, but by several of them, through the use of subcontractors. Each of these companies brings its own expertise and tools such as numerical simulation codes, and even 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

2.1.2.2. Resource aggregation

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

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 hiding to the programmers and the users the fact that resources (processors, memories, disks) are distributed. A PC cluster with such an operating system looks like a traditional multi-processor running a Single System Image (SSI).

Middleware 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 [117], PVM [103] or MPI [112]. 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 (SSI OS) is designed to perform global, dynamic and integrated resource management.

As the execution time of scientific applications may be larger than the cluster mean time between failures, checkpoint/restart facilities need to be provided not only for sequential applications but also for parallel applications. This is independent of the underlying communication paradigm. Even though backward error recovery (BER) has 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 [95]. 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.

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

Promoting a programming model that simultaneously supports distributed as well as parallel middleware systems, independently of the actual resources, raises three new issues. First, middleware systems should be decoupled from the actual networks so as to be deployed on any kind of network. Second, several middleware systems should be *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. Moreover, we have to deal with applications integrating several distributed middleware systems, like MPI, CORBA, JXTA, etc.

The last issue deals with the dynamic nature of computational grids. As targeted applications may 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 dealing 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.Therefore, adequate strategies need to be set up in order for the service to efficiently interact with the resource management system of the grid.

On the other hand, it should be noted that the algorithms proposed for parallel computing have often been studied on small-scale configurations. Our target architecture is typically made of thousands of computing nodes, say tens of hundred-node clusters. It is well-known that designing low-level, explicit MPI programs is most difficult at such a scale. In contrast, peer-to-peer approaches have proved to remain effective at 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 [99].

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 research 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. This platform is integrated to Grid 5000, a national computing infrastructure dedicated to large-scale Grid and peer-to-peer experiments. The Grid 5000 infrastructure federates experimental platforms (currently eight platforms) across France. These platforms are connected through Renater using dedicated Gigabit Ethernet links.

Our experimental platform is heterogeneous: PowerPC and PC families of processors, 32-bit and 64-bit architectures, Linux and Mac OS X operating systems. Various high performance interconnection technologies such as Myrinet and InfiniBand are available on groups of nodes. Heterogeneity allows realistic validation of interoperability of middleware and P2P systems. On the other hand, our platform is composed of sufficiently large groups of homogeneous computation nodes: 66 dual Xeon, 166 dual Opterons, 33 Xserve G5. This enables to evaluate the scalability of operating systems, runtimes and applications on various architectures.

Our experimental platform is dedicated to operating system and middleware experimentations: it is possible to repeat experiments in the same environment (same machines, same network, etc.). The allocation of our resources to the experiments is handled through *GridPrems*, a collaborative resource manager developed in our group and through *OAR*, a job manager developed by the Grenoble group of Grid 5000.

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* [109] is an example of a memory model for which all memory operations are consistent with a total order. Sequential Consistency requires that a parallel system having a global address space appears to be a multiprogramming uniprocessor system to any program running on it. Such a strict definition impacts on the performance of shared virtual memory systems due to the large number of messages that are required (page access, invalidation, control, etc.). Moreover Sequential Consistency is not necessarily required to 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* [104] 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 [107]: an *eager* one, for which modifications are propagated at the time of a release operation; and a *lazy* one, for which modifications are propagated at the time of an acquire operation. These alternative implementations differ in the number of messages that needs to be sent/received, and in the complexity of the implementation [108]. Implementations of Release Consistency rely on the use of a logical clock such as a vector clock [111]. 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* [105] 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* [110] 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 an all or nothing operation. In the event of a failure during the update of a group of data stored in stable storage, either all data remain in their initial state or they all take their new value.

3.4. 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 *peer-to-peer* (P2P) systems [116] 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 approach is illustrated by *Chord* (MIT) [120], or *Pastry* [118] (Microsoft Research).

3.5. High-performance communication

High-performance communication [100] 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 [98] 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 data management

Past research on distributed data management led to three main approaches. Currently, the most widely-used approach to data management for distributed grid computation relies on *explicit data transfers* between clients and computing servers. As an example, the *Globus* [93] 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 [113]. 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 [121] 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 [115], which is part of the latest CORBA [115] 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 [97] 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 [96]. 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 [106] 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 [101] that provides generic mechanisms for the adaptation process and for the definition of the adaptation rules is needed.

4. Application Domains

4.1. Application Domains

Keywords: *Scientific computing, cooperative applications, large-scale computing.*

The research conducted in the project-team targets several types of applications

4.1.1. Scientific computing

The project-team research activities address scientific computing and specifically numerical applications that require the execution of several codes simultaneously. This kind of applications requires both the use of parallel and distributed systems. Parallel processing is required to address performance issues 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.

4.1.2. Large-scale cooperative applications

Many other applications might benefit from the research conducted in the area of peer to peer computing. More specifically, a wide spectrum of large-scale cooperative applications spanning from publish-subscribe applications to sensor networks applications or genomic applications. Publish-subscribe systems are asynchronous event notification systems and can be used for resource discovery in large system, persistent queries (news-like applications). Sensor networks intrinsically form challenging constraint-based distributed systems in which a large set of data might be aggregated and stored. Typically sensor networks are used in monitoring applications. Finally genomic applications deal with extremely large amount of data that should be indexed: indexes as well as data themselves have to be distributed, searched and accessed easily and efficiently.

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: Matthieu Fertré, Pascal Gallard, Emmanuel Jeanvoine, Renaud Lottiaux, Christine Morin, Jean Parpaillon, Geoffroy Vallée.

Contact: Christine Morin

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

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 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 (mailto:kerrighed.users@irisa.fr) and a technical forum are available to provide a support to KERRIGHED users.

Current status: KERRIGHED (version V1.0.2) includes 90,000 lines of code (mostly in C). It represents 200 person-months of effort. The development of KERRIGHED started in late 1999. The stable release of KERRIGHED is Version V1.0.2 (April 2005) based on Linux 2.4.29. 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. In 2005, Kerrighed has also been ported to Linux 2.6.11. The code has been significantly been improved during this port resulting in a more compact software (70,000 lines of code). Moreover, a liveCD of Kerrighed based on Knoppix has been created to ease Kerrighed installation when Kerrighed is used for demonstrations or for evaluation by users not familiar with Linux installation process. Kerrighed LiveCD has been made publicly available since October 2005. Since May 2005, Kerrighed is also distributed as an *official* spin-off OSCAR package with the SSI-OSCAR package. SSI-OSCAR 3.1 is based on Kerrighed V1.0.2 and OSCAR 4.1 for RedHat 9 and Fedora Core 2 Linux distributions. A port to Debian Linux distribution is on-going.

Several demonstrations of KERRIGHED have been presented this year at *Linux Expo* (Paris, February 2005, Pascal Gallard, Renaud Lottiaux and Christine Morin), *Euro-Par 2005* (Lisbon (Portugal), August 2005, Renaud Lottiaux and Christine Morin), *ADIS meeting* (Arcueil, September 2005, Renaud Lottiaux and Christine Morin), *IRISA 30th anniversary* (Rennes, October 2005, Pascal Gallard, Renaud Lottiaux, Christine Morin)), *Supercomputing 2005 Conference* (Seattle (USA), November 2005, 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*, Seoul National University (Korea),NEC-HPCE Europe (Germany), Ulm University (Germany) and ORNL (USA). More than 1200 external downloads of KERRIGHED have been recorded in 2005.

5.2. PadicoTM

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

Participants: Christian Pérez, Thierry Priol.

Contact: Christian Pérez

URL: http://runtime.futurs.inria.fr/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 (mailto: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: 179 external downloads with 136 unique IPs between July 2002 and October 2005.

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 50 person-month effort. The first public version, referenced as PACO++ 0.1 has been released in November 2004. The second version (0.2) has been released in March 2005. It has been successfully tested on top of three CORBA implementations: *Mico*, *omniORB3* and *omniORB4*. Moreover, it supports PADICOTM.

The version 0.2 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++ has been supported by the French ACI GRID RMI. It has been used or it is used by several French projects: ACI GRID HydroGrid, ACI GRID EPSN, RNTL VTHD ++ and INRIA ARC RedGrid.

5.4. Adage

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

Participants: Sébastien Lacour, Christian Pérez, Thierry Priol.

Contact: Christian Pérez

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

Status: In development

License: GNU General Public License version 2.

Presentation: ADAGE (*Automatic Deployment of Applications in a Grid Environment*) is a research prototype that aims at studying the deployment issues related to multi-middleware applications. One of its originality is to use a generic application description model (GADe) to be able to handle several middleware systems.

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

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

ADAGE currently deploys only static applications. It supports standard programming models like MPI (MPICH1-P4 and MPICH-G2), CCM and JXTA, as well as more advanced programming models like GridCCM. The current support of GridCCM is restricted to MPI-based parallel components.

Current status: The current unstable version of PACO++ includes 35,000 lines of C++. Non-public version are used within the ACI GRID HydroGrid project.

5.5. 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.

5.6. 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 MOME DSM has been used in various contexts in the past: code coupling for the VTHD project, parallel application checkpointing during the ALCATEL collaboration, DSM-based Grid data repository in the *e-Toile* project and OpenMP runtime in the POP project. For each of these projects, new features have been integrated to the DSM software: heterogeneity of the applications and page aliasing for code coupling, support for application checkpoints, dynamic connection of the applications for Grid data-repository or fast shared memory allocator for OpenMP. In order to consolidate the integration of these new features, a new kernel has been developed for the DSM. The next release of MOME (MOME 0.1) will integrate more dynamicity (adding and removing nodes), better scalability (using a hierarchical implementation), multiple consistency models (sequential consistency, release consistency, application-managed consistency, parallel reduction), support for background checkpoint consolidation, and better support for for the POSIX SMP computation model.

Current status: MOME is implemented in C (50,000 lines) and represents a 24-person-month effort. The current stable release is MOME 0.8. The next major release MOME 1 will integrate all current developments.

5.7. JuxMem

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

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

Contact: Gabriel Antoniu,

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

License: GNU Lesser General Public License version 2.1.

Status: Registered at APP, under Ref. IDDN.FR.001.180015.000.S.P.2005.000.10000.

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 6.5.2.

Current status: Two implementations are in progress, in Java and C.It is based on the *JXTA* generic platform for P2P services (Sun Microsystems, http://www.jxta.org/). 16,000 lines of Java code and 10,000 lines of C code. Implementation started in February 2003. The first public version, referenced as JUXMEM 0.1 has been released in April 2005.

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 8.2.7). 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). An industrial collaboration with Sun Microsystems has been started in August 2005. JUXMEM is also used within other several international collaborations started in 2005: AIST (Tsukuba, Japan), University of Illinois a Urbana Champaign, University of Pisa.

5.8. 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. During 2005, GRIDPREMS has been made aware of OAR, a job scheduler deployed on the GRID 5000 platforms: nodes reserved through GRIDPREMS are removed from OAR during the reservation and nodes scheduled by OAR are visible in GRIDPREMS's global view.

5.9. Other software

5.9.1. *Peer to peer systems simulators*

A.-M. Kermarrec, Erwan Le Merrer and Etienne Rivière (Contact: mailto:Anne-Marie.Kermarrec@irisa.fr, License: Not yet defined, Keywords: P2P, simulation, unstructured overlays. Status: Under development) Several simulators were developed in Java to evaluate the proposed peer to peer systems.

The SizeWalker simulator provides a generic framework to simulate unstructured peer to peer overlays. The simulator enable to set the way the unstructured peer to peer overlay is built as well as the associated counting algorithm. This simulator has been used to evaluate the SizeWalker algorithm as well as two competitors.

Second, we developed a large scale simulator for Voronet, that can handle up to millions of nodes. This simulator is implementing both the protocol and a set of tools to examine the behavior of the system under different workloads or nodes behaviors.

5.9.2. *Workload Generator*

A.-M. Kermarrec and Etienne Rivière (Contact: mailto:Anne-Marie.Kermarrec@irisa.fr License: Not yet defined, Keywords: P2P, simulation, P2P workloads . Status: Under development)

This software has been developed in collaboration with Marteen Van Steen Vrije Universiteit in Amsterdam, and provides a workload generator for comprehensive publish and subscribe systems evaluation and comparison. This workload generator is highly configurable, is currently used to evaluate Sub-2-Sub behavior, and will be used to evaluate several existing peer to peer approaches.

5.9.3. *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.

5.9.4. *Vigne:*

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

Vigne is a prototype of a Grid-aware system for cluster federations which goal is to ease the use of computing resources in a grid for executing distributed applications. Vigne is made up of a set of operating system services based on a peer-to-peer infrastructure. This infrastructure currently implements a structured overlay network inspired from *Pastry* [119] and an unstructured overlay network inspired from *Scamp* [102] for join operations. On top of the structured overlay network, a transparent data sharing service based on the sequential consistency model and able to handle an arbitrary number of simultaneous reconfigurations has been implemented. Resource discovery and allocation services have also been implemented. Vigne prototype has been developed in C and includes 55,000 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. In 2005, Vigne system has been experimented on several sites of Grid 5000 platform.

6. New Results

6.1. Introduction

Research results are presented according to the six scientific challenges of the PARIS Project-Team. This year we added a new section that describes how our research activities are integrated within the CoreGRID Network of Excellence within which PARIS is actively involved.

6.2. Operating system and runtime for clusters and cluster federations

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

6.2.1. Kerrighed

Participants: Matthieu Fertré, Pascal Gallard, Emmanuel Jeanvoine, Renaud Lottiaux, Christine Morin, 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 [114]. 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 [47]; and (4) *Scalability*, i.e., dynamic system reconfiguration, node addition and eviction, transparently to applications.

6.2.1.1. Current achievements with Kerrighed

In 2005, two major releases have been delivered. KERRIGHED V1.0.2, released in April 2005, is the Kerrighed stable release based on Linux 2.4 kernel. It implements the resource distribution and high performance SSI properties and process checkpointing. KERRIGHED V1.1, released in November 2005, is the first release based on Linux 2.6 kernel. A major restructuring of Kerrighed code has been done between the V1.0.2 and 1.1 versions to enhance the maintainability of the software. In October 2005, a LiveCD based on Kerrighed V1.0.2 has been released as a result of a summer internship.

A port of Kerrighed on User Mode Linux (UML) architecture of virtual machine has also been started in 2005. The UML version of Kerrighed will be useful to facilitate the debugging of the system and for demonstration purposes.

The robustness of KERRIGHED has been significantly enhanced, and several new functionalities have been implemented such as high availability mechanisms to automatically reconfigure Kerrighed services in the event of a hot node addition or eviction [83].

KERRIGHED V1.0.2 has been used for the execution of wide range of applications, including legacy OpenMP and MPI applications (HRM1D, Cathare, Gorf 3D, Ligase, ...) provided by our industrial partners (EDF, DGA). Apart from scientific applications, we have also started to evaluate the potentialities of Kerrighed for other application domains: bio-informatics (Jérôme Gallard internship) and Web services (internship of Robert Guziolowski).

A start-up, KerLabs, is currently being created by Pascal Gallard and Renaud Lottiaux in order to transfer Kerrighed technology.

6.2.1.2. Integration of Kerrighed in SSI-OSCAR package

As an INRIA industrial post-doc co-funded by EDF R&D, Geoffroy Vallée has been a member of the OSCAR Team at the Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA in order to create and maintain SSI-OSCAR package which integrates Kerrighed in OSCAR software suite. 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. By combining OSCAR with Kerrighed single system image operating system, a cluster becomes easy to install, administrate, use and program [57].

The first version of SSI-OSCAR released in 2004 was a complete spin-off suite because of important modifications required to the OSCAR suite. In 2005, the OSCAR infrastructure and the SSI-OSCAR package have been modified. As a result, SSI-OSCAR package has become a standard third party OSCAR package. As such SSI-OSCAR is now automatically proposed as all other official OSCAR packages (only core packages are directly included in the OSCAR suite, third-party packages being available through on-line repositories). Three major releases of SSI-OSCAR have been made available in 2005 based on OSCAR 4.0, OSCAR 4.1 and OSCAR 5.0 for RedHat9 and Fedora core 2 Linux distributions. Geoffroy Vallée has coordinated the work related to the port of OSCAR on Debian Linux distribution [54].

6.2.1.3. Distributed file system

KerFS, Kerrighed distributed file system, has been designed and implemented to exploit the disks attached to cluster nodes. KerFS 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.

In 2005, KerFS has been enhanced to allow efficient execution of I/O intensive parallel applications. During his Master internship [81], Boris Daix has extended KerFS by integrating file splitting and data redundancy mechanisms. The MPI-IO interface for parallel I/O has been implemented on top of KerFS based on ROMIO software. KerFS performance has been compared experimentally with those of traditional files systems for collective I/O operations: NFS representative of distributed file systems and PVFS representative of parallel file systems.

6.2.1.4. Parallel Application Checkpoint/Restart

Clusters are widely used to execute scientific applications that are often message passing parallel applications with long execution times. Since the number of nodes in clusters is growing, the probability of a node failure during the execution of an application increases and the application execution time may be greater than the cluster mean time between failures (MTBF). Checkpoint/restart mechanisms can be used to avoid restarting an application from the beginning in the event of a failure. Currently, checkpoint/restart mechanisms are either implemented directly in the application source code by applications programmers or are integrated in communication environments such as MPI or PVM. We have proposed a new approach in which checkpoint/restart mechanisms for parallel applications are implemented in a cluster single system image operating system. While this kernel level approach is more complex to implement than other approaches, it is more generic because it does not require any modification, recompilation or relinking of the applications whatever the communication environment they rely on. A convenient command-line interface has been designed to trigger parallel applications checkpointing and restarting. This interface allows system initiated checkpoints[47]. All system objects of a parallel application can be detected from the identifier of one of the application processes [46]. We have implemented a global coordinated checkpointing strategy based on parallel algorithms. Communication streams can be restored without any overhead after restarting since, thanks to Kerrighed dynamic data streams, direct communication is preserved between communicating processes even if processes are not restored on their original execution node. Our approach has been prototyped in KERRIGHED single system image operating system [55] based on Linux. A preliminary performance evaluation have been conducted during Matthieu Fertré's Master internship [82]. We plan to further validate our approach with MPI and multithreaded applications and to address the scalability issues.

6.2.1.5. High availability

Kerrighed being a distributed system made up of cooperating kernels executing on the cluster nodes, a node failure has a significant impact on the operating system itself, not only on the applications being executed on top of the system. We have designed a generic service to be used by the various services composing the Kerrighed operating system to allow their automatic reconfiguration when a node is added or removed in the cluster [83], [56]. We also have studied in the context of the Phenix associated team (collaboration with Rutger University) self-healing mechanisms allowing the Kerrighed system to automatically reconfigure itself in the event of a failure. One of the goals of this work is to avoid that a node failure prevents the whole cluster to function.

6.2.2. Grid-aware Operating System

Participants: Matthieu Fertré, Emmanuel Jeanvoine, Christine Morin, Louis Rilling.

Our research aims at easing the execution of distributed computing applications on computational grids. These grids are composed of a great number of geographically distributed computing resources. This distribution and the large scale of the system make the system dynamic: failures of single resources are frequent (inter-connecting network failures and machine failures), and any participating entity may decide at any time to add or remove nodes from the grid. To ease the use of such dynamic distributed systems, the approach we are defending is to build a distributed operating system that provides single system image, that is self-healing, and that can be adapted to the users' needs. Such an operating system is composed of a set of distributed services that each provides single system image for a specific type of resource in a fault-tolerant manner. We are implementing this system on a research prototype called Vigne. Experimental evaluations are made on the Grid5000 research grid.

The work of year 2005 is twofold. First we kept on designing a second fault-tolerant consistency protocol allowing to implement a data sharing service targeted to private and volatile data of applications. Compared to the first consistency protocol we developed earlier [52], [53], this protocol allows to obtain better performance for applications that do not perform concurrent accesses on a data that include write accesses [15]. Such access schemes are the most frequent in well programmed applications.

Second, we have worked on the design and implementation of grid operating system services for resource and application management. The objective of our work is to allow users to launch applications to be reliably executed in a grid without having to know where the applications are executed and without dealing with failures. When launching an application, the user only provides a description of the application execution constraints such as for example the processor architecture, memory requirements, operating system.

A resource discovery service based on an unstructured peer-to-peer network has been designed. In contrast to structured networks, unstructured networks allow to perform complex queries like range queries or multi-attributes queries needed for resource discovery. The Vigne peer-to-peer infrastructure that was initially based on a structured overlay network has been extended with an unstructured overlay network inspired from *Scamp* [102] for join operations. The two networks have been designed to cooperate resulting in a balanced unstructured network at no cost. A first evaluation of the Vigne infrastructure scalability in a dynamic environment has been performed by simulation using Grid 5000 platform. The evaluations with 100, 500, 1000 and 2000 nodes show that whatever the number of nodes, the bandwidth cost for overhauling the infrastructure in presence of reconfigurations in the grid (node addition, eviction or failure) is constant.

In a first step, we have implemented flooding and random walk traditional policies to discover grid resources fulfilling the requirements of an application. In a second step, we have designed a new discovery algorithm based on learning strategies in order to decrease the bandwidth consumption generated by the resource discovery service.

An application manager service has been designed to take in charge the application life cycle and the link with the user. When an application is submitted to Vigne system, an application manager is created by the Vigne the system at a random but known place in the grid. The application manager can thus be retrieved by the Vigne system for handling user queries (the user may want to know the current status of the application execution). The application manager is in charge of dealing with reliable application execution in the event of failures, according to the strategy defined by the user. We have initiated a study on checkpoint/restart mechanisms that could be used by the Vigne system in a grid. One of the issue is to deal with process identifiers that are saved in application checkpoints and that are not unique in the grid. Another issue is to design checkpointing strategies for distributed applications that can be used in a large scale dynamic system.

The Vigne prototype has been extended to implement the above mentioned services. Vigne prototype is composed of a client and a set of daemons. The client part of Vigne is designed to be executed on the user workstation and provides a user interface for job submission. An instance of the Vigne daemon is executed on each node of the grid, a node being a single computer or a cluster running either a single system image operating system such as Kerrighed or a batch system on top of Linux. A system interface used to link Vigne daemon to cluster schedulers such as OpenPBS, LSF or Kerrighed has been designed. Currently, this interface has only been implemented for Kerrighed.

A European IP project proposal, XtreemOS[86], coordinated by Christine Morin and related to Section 2.5.4 *Advanced Grid Technologies, Systems and Services* of the 2006 Work Programme has been submitted in September 2005 to the IST Call 5. The overall objective of the XtreemOS project is the design, implementation, evaluation and distribution of an open source Grid operating system with a native support for virtual organizations (VO). The proposed approach is the construction of a Grid OS made up of a set of system services based on the traditional general purpose OS Linux, extended as needed to support VO and to provide appropriate interfaces to the Grid OS services. The XtreemOS consortium comprises of 18 academic and industrial partners. Different end-users are involved in the XtreemOS consortium providing various test cases in scientific and business computing domains.

6.2.3. 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.

6.3. Middleware for computational grids

6.3.1. The PadicoTM framework

Keywords: *CORBA, Communication framework, MPI.*

Participants: Mathieu Jan, Christian Pérez, Thierry Priol.

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.

With the employment of Alexandre Denis in the RUNTIME project, PADICOTM is an activity shared between the RUNTIME and PARIS projects. With respect to the PARIS project, the year 2005 has been devoted to the continuation of the stabilization of the port of JXTA-C on top of PADICOTM. The port has been updated as long with the evolution of JXTA-C as well as the Apache Portable Runtime (APR), which is used by JXTA-C.

6.3.2. Parallel CORBA objects and components

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

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

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 2005 was related to the improvement of PACO++. First, a more stable release has been produced. Second, the RedSym library from the Scalaplix project has been integrated into PACO++. Third, a port of the core of the Salome platform on top of PACO++ has been studied.

The ACI GRID HydroGrid project has produced a version of an application simulating a coupling between a transport and a flow codes through the used of PACO++.

PACO++ has reached a stable point which validates its purpose.

6.3.3. Dynamic software component models

Keywords: *CORBA Component Model (CCM)*, *Grid*, *dynamic behavior*, *software component*.

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

Software component models are succeeding in handling another level of the software complexity by dealing with its architecture. However, a current limitation is to only handle *static* architecture. Dynamic behaviors, like the well-known master-worker pattern, are not expressible. It is an important lack as many applications require such a feature. Our objective is to study how to capture such dynamic behavior within a component model.

Started with a very simple and well-known dynamic behavior, the master-worker design pattern, we have introduced the concept of collection of components and of communication between a component and such a collection. Hence, from an application point of view, the master-worker behavior is well captured. This architecture is captured in an *abstract* ADL which currently is an ADL enriched with this notion of collection. An abstract ADL is transformed into a concrete ADL by selecting an actual mechanism of request transport from the master to the worker. This choice is being formalized as the application of a template to the abstract ADL.

The proposed solution has been applied to two software components, CORBA Component Model and Fractal. A first hand-coded prototype has convinced us of the feasibility of the approach.

Three activities are envisioned. First, we are working on integrating existing request framework, like DIET or XtremWeb, in our component model. Second, we will continue to increase the level of dynamic behavior support, by for example, supporting a dynamic set of workers. Third, the links between the deployment model and such a dynamic component model have to be studied.

6.3.4. Application deployment on 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. This is complementary to previous works 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 2005, we have proposed specifications for describing MPI applications as well as GridCCM applications and a generic application description model.

The specification GridCCM applications consists in an extension to the CCM specification which allows an implementation to be a parallel application. For example, if the parallel component has based on MPI, its description follows the MPI description. It was a motivation for specifying such an MPI description as well as the need to be able to deploy plain MPI applications.

The generic application description model (GADe) enables to decouple most of the deployment tool from a specific application description. Translating a specific application description into the generic description is a simple task. Then, developing new planning algorithms and re-using them for different application types becomes much easier as they rely on the generic description. Moreover, the generic description model allows to deploy applications based on a programming model combining several models, as parallel components encompass component-based and parallel programming models for instance.

The development of ADAGE has been continued. ADAGE is currently able to deploy MPICH-P4, MPICH-G2, standard CORBA Component, GridCCM components and JXTA based applications on Grids managed by the Globus Toolkit 2 or by SSH/SCP.

Our 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, there are two levels of components to handle: CORBA component as well as PADICOTM component. Another objective is to support dynamic applications.

6.3.5. 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. 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 need 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.

Based on our model for the adaptation of parallel component, on our previous experience and on a collaboration with University of Pisa [27], we have defined a generic model of dynamic adaptation.

We have started the development of AFPAC, an implementation of the proposed model in the case of parallel components. Experiments have been conducted that consist in dynamically changing the number of processes of the NAS Parallel Benchmark MPI-FFT standard code. Results have been published in [39].

AFPAC has also been used by students in the context of a student project at INSA-Rennes that aimed at providing a resource manager for adaptive components.

In order to ease the integration of AFPAC into components, we have investigated aspect-oriented programming [58]. The specific TACO aspect weaver has been developed in the context of a Master student project. Aspects for integrating AFPAC have been written in the context of a Master summer internship funded by ARC COA. This aspect-oriented approach has been experimented with the Gadget-2 cosmological N-body/smoothed particle hydrodynamics simulation code.

6.4. P2P System Foundations

6.4.1. Clustering in peer-to-peer systems

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

Participants: Yann Busnel, 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 carried on the study of the clustering properties [70] of an eDonkey peer-to-peer file sharing workload.

We also refined a semantic proximity measure [68] capturing peer generosity and file popularity. These factors had been previously identified as potential biases on the genuine proximity measure. We also proposed a decentralized popularity tracking algorithm, based on epidemic algorithms. Those approaches have been evaluated against a real eDonkey trace.

6.4.2. Querying peer-to-peer systems

Keywords: *Search in peer-to-peer systems, Voronoi diagram, epidemic algorithms, 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 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).

6.4.2.1. GosSkip: gossip-based peer to peer publish-subscribe system

This work has been done in collaboration with Sidath Handurukande and Rachid Guerraoui (EPFL, Switzerland). GosSkip is an efficient attribute-based publish-subscribe system. GosSkip is a structured overlay which structure reflects the actual structure of the underlying application properties. GosSkip [69] 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. In GosSkip, events are delivered to matching subscriptions in O(log N) routing hops, N being the total number of subscriptions. Experimental results based on a real P2P trace convey the scalability, the failure resilience, the efficiency and the fairness of the approach both in static and dynamic scenarios. We plan to extend this approach to support range queries in the future. We are also currently working on leveraging the presence of multiple subscriptions (logical peers) on a physical node.

6.4.2.2. VoroNet: a distributed storage system for multidimensional data based on Voronoi diagrams

The Voronet project aims at building a fully distributed overlay network for data storage system with strictly proven algorithmic costs. This work is done in collaboration with Loris Marchal (ENS Lyon) and Olivier Beaumont (LABRI Bordeaux). The idea of Voronet is to generalize the Kleinberg model where each peer in an overlay is connected to its neighbors on a grid as well as to a remote node picked at random with a probability proportional to its distance. Every data item is specified by a set of values over k attributes, positioning the item in an k-euclidean space. Each data item represents a logical peer in the Voronet overlay and is connected to a set of neighbor, sharing vertexes in the Voronoi tessellation of the Euclidean space. These links are eventually forming the Delaunay complex of the set of elements. Each peer also knows a remote peer to provide efficient polylogarithmic routing between any two peers in the overlay, independently of the distribution of peers in the space. We are currently extensively evaluating Voronet. We plan to study the extension of this work to higher dimensions.

6.4.2.3. Sub-2-Sub: large scale cooperative publish-subscribe system

This work has been done in collaboration with Spyros Voulgaris and Pr. Marteen Van Steen from the Vrije Universiteit in Amsterdam and started while Spyros Voulagris was visiting the PARIS research group for two months in May and June 2005. The Sub-2-Sub project aims at providing a self-organizing overlay network for content-based publish-subscribe systems. Sub-2-Sub builds an unstructured peer to peer overlay linking subscriptions using epidemic algorithms so that subscribers having similar interests are automatically clustered. Events are then routed towards clusters and efficiently disseminate within such clusters. In this context, we developed a workload generator and simulation platform to highlight problematic behaviors of such a system or potential target applications.

6.4.3. Unstructured peer-to-peer overlays

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

Participants: Anne-Marie Kermarrec, Erwan Le Merrer.

6.4.3.1. Epidemic algorithms

Peer-to-peer self-organizing unstructured overlays networks have proven to provide good support for several distributed applications. 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. This year, we conducted an extensive experimental study of various gossip-based algorithms .

6.4.3.2. Decentralized estimation size algorithms

Peer to peer systems are characterized by the fact that peers only have a limited knowledge of the system. Therefore, no peer is aware of the global membership and able to compute the system size. We propose two algorithms, *the random tour* and the *sample and collide* algorithms to estimate the system size. These algorithms are fully decentralized, and based on random walks. The basic idea of the *Random tour* algorithm is that a peer initiates a random walk in the system. The message, associated to this random walk carries a density information (number of neighbor) across the network and provides an accurate estimation when it returns to the process initiator. In the *sample and collide* approach, an initiator iterates on a sampling approach. the estimation of the system size is based on the redundancy observed in the samples. In collaboration with Laurent Massoulié (Microsoft Research, UK), we provided a theoretical proof of the properties of those algorithms. We are currently conducting an extensive simulation study to compare this approach to two others recent approaches (Aggregation and probabilistic polling approaches).

6.4.4. Peer to peer sensor networks

Keywords: *gossip-based algorithms, peer-to-peer overlays, sensor networks*.

Participants: Marin Bertier, Yann Busnel, Anne-Marie Kermarrec, Aline Viana.

We are currently investigating the use of peer to peer epidemic algorithms in sensor network systems. This very recent work takes place in the context of a specific interest group (Groupement d' intérêt scientifique) composed of scientist coming from other areas (physics, chemistry, biology, computer science and signal processing). We conduct this research in the context of monitoring children activities to detect obesity pathologies. The experiment should be conducted on large populations to be valid. Many issues arise from such an application context: distributed data gathering, identification, overlay network maintenance, high-level data management (integrity, consistency, search, computations).

6.5. Large-scale data management for grids

6.5.1. Mome data-repository

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 domains, 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 implementation resulting from this project is based on MOME 0.8. When a parallel application is launched on a group of nodes, each process of the application connects to a local daemon and gets access to the data repository. The application processes and the DSM daemons run in different address spaces and communicate through Unix pipes and shared segments. The application processes can fail safely (or be killed) without impacting the DSM. The data repository is persistent: the segments retain their data after all application processes have disconnected.

The major limitations of this implementation come from the poor organization of the computation nodes in the DSM: MOME 0.8 considers a static and flat organization of the nodes.

To tackle these limitations, a new DSM kernel has been implemented for MOME 1. Computation nodes can now be added and removed dynamically. The nodes can be organized in a hierarchical way, reflecting the structure of clusters of clusters, and allowing better scalability of the system.

6.5.2. The JuxMem data-sharing service

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

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

Since 2003, we work on the concept of *data-sharing service* for grid computing, that we defined as a compromise between two rather different kinds of data sharing systems: (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)*. We illustrated this concept through the JUXMEM software platform. 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 [17]. 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 [94] generic peer-to-peer framework, which provides basic building blocks for user-defined peer-to-peer services.

In 2005 we have improved and refined this architecture, in order to add new consistency protocols (see Section 6.5.3). We have performed a preliminary experimental validation of the JuxMem data sharing service, connected to the DIET grid computing environment (ENS Lyon), using the Grid-TLSE application (IRIT, Toulouse). We have also used JUXMEM in conjunction with the Mico implementation of CCM, to illustrate the enhancement of the component model with a data sharing functionality.

6.5.3. Fault-tolerant consistency protocols

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

Participants: Gabriel Antoniu, Loïc Cudennec, Sébastien Monnet.

JUXMEM's architecture decouples consistency management from fault-tolerance management and defines 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.

In 2005, we refined the above architecture [19], and we designed and implemented a new fault-tolerant consistency protocol adapted to visualization in grid code-coupling applications [44]. This protocol introduces the *relaxed read* operation, an efficient read which does not require lock-based synchronization. This operation may not return the latest version of the data, but it allows the user to control the "freshness" of the data returned. Such operations may be useful for efficient, non-intrusive visualization. Extensive tests of this protocol have been performed on the Rennes cluster of Grid'5000. Further multi-cluster experiments are in progress.

In parallel, in collaboration with Indranil Gupta's team from the University of Illinois at Urbana Champaign, we started working on ways to use probabilistic fault-tolerance strategies to support data consistency in distributed applications based on dynamic collaborative groups. This results in the definition of an application-driven peer-to-peer overlay. A paper is in progress on this topic. We intend to evaluate the approach using JUXMEM.

6.5.4. *High-performance JXTA communications for grids*

Keywords: *JXTA*, *communication optimization*, *peer-to-peer*, *performance evaluation*.

Participants: Gabriel Antoniu, Mathieu Jan.

Based on a few benchmarks that we started writing in 2004 (internship of David Noblet, Univ. of New Hampshire), we have performed an extensive evaluation of the performance of JXTA's communication layers [30], [29], [29] on top of grid networking infrastructures: SANs (Myrinet, Giga-Ethernet), LANs (Fast-Ethernet) and WANs (Giga-Ethernet). The goal of this evaluation was to measure to what extent JXTA's communication layers are able to efficiently take advantage of Gigabit/s networks, in order to properly meet the performance constraints of scientific grid applications. We show how these layers can be tuned and configured for such a use and we conclude that JXTA can be used on grids not only for resource discovery (which was its most obvious use), but it can also efficiently be utilized for data communication. These study is important in the context of an increasing use of JXTA on grid infrastructures by various international research projects, as an ever stronger convergence can be noticed between grid computing and P2P computing. A paper [29] on this topic received the Best Presentation Award at the Global and P2P Computing Workshop held in conjunction with the CCGRID 2005 conference.

6.6. Advanced computation models for the Grid

Participants: Jean-Pierre Banâtre, Yann Radenac.

This work is carried out in close cooperation with Pascal Fradet, from INRIA Rhône-Alpes (Project-Team *POP ART*).

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.

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. This work has been published this year in [33].

The proceedings of last year major event, the *Unconventional Programming Paradigms (UPP '04)* workshop, have been published by Springer-Verlag in Spring 2005 [11], [32]. The *Grand Challenge in Non-Classical Computation* workshop has been a great occasion to expose our model [31] and to have a large overview on non-conventional models of computation. We have also posed some fundamental questions [36] about non-classical programming languages.

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 higher-order properties, provide a very general and powerful tool for expressing very general (and original) coordination schemes. This work has been presented in the *International Workshop on Developments in Computational Models* which will be published in an ENTCS volume [35]. A complete version is under construction and will be available as an INRIA Research Report [66].

Our most recent work concerns task coordination on Grids within the chemical framework [34]. In a first step, applications are programmed in an abstract manner describing essentially the chemical coordination between (not necessarily chemical) software components. In a second step, chemical service programs are specifically provided to the run-time system in order to obtain from the resources the expected quality of service in terms of efficiency, reliability, security, etc. Moreover, the implementation of a prototype is underway.

6.7. Experimental Grid Infrastructure

Participants: Yvon Jégou, Vincent Lefevre, David Margery, Pascal Morillon, Guillaume Mornet.

The PARIS Project-Team manages an experimental computation platform dedicated to operating system, runtimes, middleware, grid and P2P research. This platform is now 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. Finally, 100 dual AMD Opteron from HP were integrated in November 2005. Our platform contains now more than the targeted 500 processors.

The interconnection of the computation nodes inside our platform as well as the interconnection of our platform to the other GRID 5000 platforms have also been been upgraded during 2005. A direct connection between our platform and Renater using a dedicated Gigabit link was deployed in January 2005. Renater plans to deploy a 10 Gigabit/s interconnection of the GRID 5000 sites end of 2005 or beginning of 2006. We have already initiated the acquisition of new Ethernet switches in order to exploit this future communication capacity. The exploitation of intra-cluster high performance networks is a major research subject of our project team at the operating system level as well as for runtime systems. In November 2005, 66 nodes have been equipped with InfiniBand at 10 Gigabit/s. Another group of 33 nodes were connected through Myrinet 10G in December.

6.8. CoreGRID Network of Excellence

Participants: Françoise André, Gabriel Antoniu, Hinde Bouziane, Jeremy Buisson, Christian Perez, Thierry Priol.

The PARIS Project-Team participates to two virtual institutes of CoreGRID, the Institute on Programming Model and the Institute on Problem Solving Environment, Tools and GRID Systems.

Within the Institute on Programming Model, the PARIS Project-Team was involved in a common deployment model, in a data management architecture and in a common adaptability framework.

We worked on the definition of a common deployment model for software component based application. This model, jointly achieved with UNIPI, highlights important features requested to deploy application to a grid such as rich possibility to describe applications, multi-middleware application description, dynamic application support, network-enhanced resource description, resource and execution constraints and many grid middleware supports.

We worked on the problem of dynamic, self adaptation of components to their execution environment. On that subject, a collaboration has been started with UNIPI. The two teams have different approaches to dynamic adaptation. In the PARIS Project-Team, adaptability is considered to be under the responsibility of the developer that has a full control over it, while frameworks and design methods are provided to the developer to help him in that task. In the UNIPI team, adaptability is hidden to the developer, generated by a compiler that relies on high-level constructs. Starting from the two independent approaches, we have jointly proposed a common model for dynamic adaptation that synthesizes the two approaches. This common model has been shown in [27] to apply well to the two initial approaches. We have also studied the integration of adaptation aspect in the Generic Component Model (GCM).

In collaboration with UNIPI, we have sketched out a preliminary data management architecture allowing grid component models to be enhanced with inter-component data sharing. Transparent access to data is provided to components via a hierarchical data sharing system, designed by integrating UNIPI's ad-HOC cluster-level data sharing system (used by ASSIST) with the JuxMem grid-level data-sharing service built at INRIA. This integration work is currently in progress within the framework of a MS thesis at UNIPI.

Last, we contribute to the definition of the properties of the Grid Component Model.

Within the Institute on Problem Solving Environment, Tools and GRID Systems, a first activity was to study the usage of software component technology within GridSuperScalar as well as the communication abstraction that needs to be supported. A second action has been initiated to study the usage of the UPC's resource description format into an INRIA's Grid-aware application deployment tool.

7. Contracts and Grants with Industry

7.1. 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.

7.2. Edf 1

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: September, 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. The most recent release of SSI-OSCAR has been presented at SC05 in November 2005.

7.3. Edf 2

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.

7.4. Edf 3

Participants: Christian Perez, André Ribes.

Program: The collaboration with EDF R&D aims at designing,

Starting time: May 1st, 2005

Ending time: November 1st, 2005

Partners: EDF R&D, INRIA

Support: EDF R&D funding, post-doc grant (André Ribes)

Project contribution: The work carried out by the PARIS Project-Team relates to the design and implementation of parallel object and component models. As part of this contract, the benefit of using technologies from INRIA, namely PACO++ and the RedSym library, to realize a parallel implementation of the Salome platform will be evaluated.

7.5. Dga

Participants: Pascal Gallard, Renaud Lottiaux, Christine Morin, Louis Rilling.

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 has studied extensions to KERRIGHED operating system to make it a *Grid-aware* operating system for cluster federations.

In 2005, we have worked on the design and implementation of Kerrighed V2.0 and on the improvement of the system robustness. We have also work on the design of a *Grid-aware* operating system for cluster federations. A research prototype of such a system has been built on top of Kerrighed including a data sharing service and a basic resource allocation service.

7.6. Sun Microsystems

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

Program: This collaboration with Sun aims at studying how to efficiently use peer-to-peer techniques available with JXTA on grid testbeds such as Grid’5000.

Starting time: October, 2005

Ending time: September, 2008

Partners: Sun Microsystems,INRIA

Support: Sun funding, Ph.D. grant (Loïc Cudennec)

Project contribution: The work relates to the use of high-performance communication techniques, able to optimize JXTA’s communication performance on grid testbeds. A second goal is to adequately support the interaction between JXTA-based applications and grid resource management systems, in order to cope with dynamic resources.

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 Loïc Cudennec (starting on October 1, 2005, 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 is director of the ACI GRID from the Ministry of Research. He 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: Hinde-Lilia Bouziane, 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 have realized an implementation of the application above a parallel object model (PACO++).

8.2.4. 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.

8.2.5. 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.

8.2.6. ACI GRID Grid 5000

Participants: Yvon Jégou, Vincent Lefevre, David Margery, Pascal Morillon, 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 enables experimentations on operating systems, middlewares, and communication 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 experimental platform was 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 2004. All these clusters were in production for the Super Computing SC ’04 presentation at Pittsburg, PA, November 6-12. Finally, the PARIS platform reached the 500 processors target in November 2005 after the integration of a 100 dual Opteron nodes cluster from HP.

Handling communication inside clusters is an active research activity in the PARIS project. 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. The integration of high performance networking in our GRID 5000 platform has been initiated in 2005: dedicated 1 Gigabit/s link to Renater in January, 66 nodes interconnected through InfiniBand (10 Gigabit/s) in October, 33 nodes interconnected through Myrinet 10G in December and GRID 5000 platform interconnection upgrade to 10 Gigabit/s planned in December 2005/January 2006.

8.2.7. 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.

8.2.8. 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 6.5.2). 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 2005 we have improved and refined the global GDS architecture, by specifying how DIET needs to interact with JUXMEM. We have performed a preliminary experimental validation of the JuxMem data sharing service, connected to the DIET grid computing environment (ENS Lyon), using the Grid-TLSE application (IRIT, Toulouse). We performed an extensive evaluation of the JXTA communication layers, as a first step towards a full performance evaluation of the service. Scalability tests are in progress, using the Grid'5000 testbed.

8.2.9. 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.

8.2.10. 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. In 2005, preliminary scalability experiments for GDS have started and are in progress. Gabriel Antoniu is local correspondent of GdX for the PARIS Project-Team. Project site: http://www.lri.fr/~fci/GdX/.

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

8.2.12. ACI CE CoreGRID

Participant: Thierry Priol.

This project (http://www.coregrid.net) aims at helping the PARIS Project-Team to set up the COREGRID *Network Of Excellence*.

8.2.13. Other grants

8.2.14. 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).

8.3. 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 and ended in February 2005. 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

Participants: Françoise André, Gabriel Antoniu, Hinde Bouziane, Jeremy Buisson, Päivi Palosaari, Christian Pérez, Thierry Priol.

Thierry Priol is the Scientific Coordinator of a *Network of Excellence* proposal, called COREGRID, in the area of Grid and Peer-to-Peer (P2P). The COREGRID network started on September 1, 2004. As many as 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. The primary objective of the COREGRID Network of Excellence is to build solid foundations for Grid and Peer-to-Peer computing both on a methodological basis and a technological basis. This will be achieved by structuring research in the area, leading to integrated research among experts from the relevant fields, more specifically distributed systems and middleware, programming models, knowledge discovery, intelligent tools, and environments.

The research programme is structured around six complementary research areas, i.e. work packages that have been selected on the basis of their strategic importance, their research challenges and the European expertise in these areas to develop next generation Grids: knowledge and data management, programming model, system architecture, Grid information and monitoring services, resource management and scheduling, problem solving environments, tools and Grid systems.

INRIA is managing the network in collaboration with the ERCIM office which is in charge for administrative and financial management while Thierry Priol as a scientific coordinator (SCO) is leading the Network with regard to the scientific aspects and the overall running of the project. At the beginning of the project he established an SCO office for him and his assistant Päivi Palosaari, who began her work on December 1, 2004. The main tasks of the SCO office during the first year were coordination and monitoring of the activities related to the scientific and technical workpackages, coordinating the COREGRID Scientific Advisory Board, performing the first ranking of partners activity, coordinating the preparation of the second Joint Programme of Activities and providing the first internal assessment of the network. In addition, the SCO office participated in dissemination tasks by giving presentations, contributing to the COREGRID Newsletters etc.

Christian Pérez is the responsible for the COREGRID contract with INRIA. He is responsible for managing the four INRIA teams (PARIS, GRAND-LARGE, OASIS and SARDES) with regard to periodic reporting etc. His main tasks were to represent INRIA in the COREGRID Members General Assembly meetings and votes.

8.3.3. GridCoord

Participants: Luc Bougé, Thierry Priol.

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.

8.4. International bilateral grants

8.4.1. Europe

University of Ulm, Germany. The bi-lateral research collaboration with the distributed system group of the University of Ulm initiated in 2004 has been continued in the 2005 *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, Michael Schöttner and Stefen Frenz from Ulm University participated to a workshop organized on May 3rd at IRISA. In December 2005, Yvon Jégou and Christine Morin were hosted for a two-day visit at Ulm University.

Vrije Universiteit, Amsterdam, The Netherlands. The collaboration between the PARIS project-team (A.-M. Kermarrec) and The “Large-scale Distributed Systems” group of Vrije Universiteit (VU) (Maarten van Steen) takes place in the context of the Van Gogh program (PAI) with Vrije Universiteit since January 2005. The research program is aiming at addressing the problem of effective and efficient content-based searching by improving zero-knowledge semantic overlay construction in the context of resource discovery in a grid-like environment.

In this context, Elth Ogston, Spyros Voulgaris and Stevens Leblond visited the PARIS project-team for respectively 15 days, 6 weeks and 6 weeks during Spring 2005. Etienne Rivière and Yann Busnel spent respectively 4 and 2 weeks at the VU during Fall 2005.

8.4.2. North-America

UIUC-INRIA-CNRS In 2005 we started a 2-year collaboration with Indranil Gupta's team from the University of Illinois at Urbana Champaign (UIUC). Sébastien Monnet visited UIUC in June 2005 (one month) and Ramses Morales (UIUC) visited IRISA in July 2005 (one month). During these visits, we started working on the use of probabilistic fault-tolerance strategies to support data consistency in distributed applications based on dynamic collaborative groups. This results in the definition of an application-driven peer-to-peer overlay. The approach will be experimentally evaluated using the JUXMEM software.

Rutgers University, USA. We have collaborated with the *Discolab* Research Team leaded by Liviu Iftode at Rutgers University in the framework of the Phenix associated team funded by INRIA since January 2005. We investigate the design of a novel, highly-available cluster architecture based on the concept of *remote healing*. A workshop has been organized at IRISA on December 1st and 2nd. Liviu Iftode and four of his students participated to this workshop.

MIT, Boston, USA. Arvind Saraf, 3rd year undergraduate student at MIT (Boston, Massachussets, USA) has spent a 3-month internship within the PARIS Project-Team, from June to August 2005. He has been supervised by Gabriel Antoniu and Mathieu Jan. He implemented a prototype allowing the JUXMEM data sharing service to interact with the P3 global computing environment, developed by AIST (Tsukuba, Japan).

ORNL, Oak Ridge, USA. Geoffroy Vallée has been hosted by ORNL during his INRIA industrial post-doc co-funded by EDF R&D (until September 2005 [1]. In the framework of this collaboration, Kerrighed has been integrated as an official spin-off package in the OSCAR software suite for high performance computing on Linux clusters.

ORNL is involved in two FastOS projects http://www.cs.unm.edu/~fastos/ funded by the DOE: PetascaleSSI and Molar. PetascaleSSI aims at creating a Petascale SSI. Molar (MOdular Linux and Adaptive Runtime support for HEC OS/R research) aims at adaptive, reliable, and efficient operating and runtime system solutions for ultra-scale high-end scientific computing on the next generation of supercomputers. SSI-OSCAR may be used to provide some SSI features in these projects. A solution for cluster virtualization based on Xen for kernel research and testing has also been studied.

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, ended in June 2005, 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 June 2005, four researchers from SNU visited ENS-Lyon to complete the activities started with the GRAAL and the PARIS Project-Team.

[1] Since October 2005, Geoffroy Vallée has been employed by ORNL)

8.5. Visits

Jorge Corral Arean: Master student at the Universidad de la Republica in Montevideo (Uruguay) spent 15 weeks in the PARIS research team and worked in collaboration with Françoise André and Jérémy Buisson in the area of contracts and adaptation for software components in January-May 2005.

Stevens Leblond: Master student at the Vrije Universiteit in Amsterdam (NL) spent 6 weeks in the PARIS research team and worked in collaboration with A.-M. Kermarrec in the area of clustering analysis of peer to peer file sharing system trace in May-June 2005.

Ramses Morales: Ph.D student at UIUC spent 4 weeks in the PARIS research team and worked in collaboration with Gabriel Antoniu and Sebastien Monnet in the area of data consistency in distributed applications in June 2005.

Elth Ogston: Post-doctorate researcher at the Vrije Universiteit in Amsterdam (NL) spent 2 weeks in the PARIS research team and worked in collaboration with A.-M. Kermarrec in the area of semantic peer to peer networks in April 2005.

Spyros Voulgaris: Ph.D student at the Vrije Universiteit in Amsterdam (NL) spent 6 weeks in the PARIS research team and worked in collaboration with A.-M. Kermarrec in the area of epidemic-based publish-subscribe systems in May–June 2005.

9. Dissemination

9.1. Community animation

9.1.1. Leaderships, Steering Committees and community service

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

ACI GRID, Ministry of Research. Th. Priol is the Director of the 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. 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. 1000 persons. The term of L. Bougé will end at the end of 2005.

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 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 Le Croisic in 2005.

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.

COSET-2 Workshop. Ch. Morin co-organized with Stephen Scott, ORNL, the *Second International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters*. It was held in Boston in June 2005, in conjunction with ICS 2005. About 20 researchers attended the workshop (http://coset.irisa.fr).

European IST POP. Ch. Pérez is the INRIA Scientific Correspondent of the European IST POP project, which ended in February 2005.

European GRIDCOORD IST-FP6 SSA. L. Bougé and Th. Priol participate to the GRIDCOORD *Specific Support Action* (SSA) through the INRIA institutional member. After the tragic disparition of Isabelle Attali, INRIA Sophia-Antipolis, L. Bougé 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 24 months (http://www.gridcoord.org/).

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/).

PGAMS Workshop Series. T. Priol is a member of the *Steering Committee* of the *The Second International Workshop on Programming Grids and Metacomputing Systems and Distributed Processing Symposium*, http://www.mathcs.emory.edu/dcl/meetings/pgams2005).

9.1.2. Editorial boards, steering and program committees

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

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

CDUR 2005: *Journées Francophones de la Cohérence des Données en Univers Réparti*, November 2005, in Paris, France.

CCGrid 2006: *IEEE/ACM International Symposium on Cluster Computing and the Grid*, to be held Singapore, May 2006.

HPDGrid 2006: *International Workshop on High-Performance Data Management*, to be held in Rio de Janeiro, July 2006, in conjunction with VecPar 2006.

J.-P. Banâtre served in the Program Committees for the following conferences:

WORDS 2005: *Workshop on Object-Oriented, real-time dependable systems*, Sedona, USA, February 2005.

ISORC 2005: *International Symposium on Object-oriented Real-time distributed Computing*, Seattle, USA, May 2005.

WWSFM 2005: *Workshop on Web Services and Formal Methods*, Versailles, France, September 2005.

L. Bougé belongs to the *Editorial Advisory Board* of the *Scientific Programming* Journal, IOS Press. He served in the following Program Committees:

NPC 2005: *IFIP International Conference on Network and Parallel Computing (NPC 2005)*, Beijing, China, November 2005.

A.-M. Kermarrec served as Global Chair for the *Peer-to-peer and Web Computing* topic of Euro-Par 2005, Lisbon, Portugal, August 2005. She served as Publicity Chair for the *20th ACM Symposium on Operating Systems Principles (SOSP)*, Brighton, UK, November 2005. She served as Publicity Chair for the *EuroSys 2006* to be held in Leuven, Belgium in April 2006.
She served in the Program Committees for the following conferences:

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

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

WCW 2005: *IEEE Tenth International Workshop on Web Content Caching and Distribution*, Sophia Antipolis, France, September 2005.

DOA 2005: *International Symposium on Distributed Objects and Applications (DOA)* Agia Napa, Cyprus, October 2005.

Middleware 2005: *ACM/IFIP/USENIX 6th International Middleware Conference*, Grenoble, November 2005.

Co-Next'05: Toulouse, France, November 2005.

C'DUR 2005: *Journées Francophones sur la cohérence des données en univers réparti*, Paris, November 2005.

IPTPS '06: *International workshop on peer-to-peer Computing*, to be held in Santa-Barbara, CA, USA, February 2006.

IEEE Global Internet Symposium 2006. To be held in conjunction with Infocom in Barcelona, Spain in April 2006

DSN 2006: *The International Conference on Dependable Systems and Networks* to be held in Philadelphia, PA, USA in June 2006.

ICDCS '06: *International Conference on Distributed Computing Systems* , Operating system and Middleware track, to be held in Lisboa, Portugal in July 2006.

IWDDS '06: *The first International Workshop on Dynamic Distributed Systems* to be held in conjunction with ICDCS in July 2006.

Algotel 2006: *The International Conference on Dependable Systems and Networks* to be held in Philadelphia, PA, USA in June 2006.

Euro-Par 2006: *Peer-to-peer and Web Computing* Topic of Euro-Par 2006, to be held in Dresden, Germany, in August 2006.

IWSOS '06: *New Trends in Network Architectures and Services: International Workshop on Self-Organizing Systems* to be held in Passau, Germany in September 2006.

Middleware 2006: *ACM/IFIP/USENIX 7th International Middleware Conference* to be held in 2006.

CFSE 2006: *Conférence Française en système d'exploitation* to be held in 2006.

R. Lottiaux served in the Program Committees for the following conferences:

COSET-2: *Second International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters* held in Cambridge, MA, USA, June 2005.

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

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

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.

COSET-2: *2 nd International workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters*, Cambridge, MA, USA, June 2005.

ISPDC 2005: *International Symposium on Parallel and Distributed Computing*, Lille, France, July, 2005.

Cluster 2005: *IEEE Cluster 2005*, Boston, USA, September 2005.

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

I-SPAN 2005: *8th International Symposium on Parallel Architectures, Algorithms, and Networks*, Las Vegas, USA, December 2005.

DCCS 2005: *International workshop on data consistency in collaborative systems*, that will be held in conjunction with the *1st International conference on Collaborative Computing (CollaboratateCom '05)*, San Jose, USA, December 2005.

DSM 2006: *International Workshop on Distributed Shared Memory on Clusters*, organized in conjunction with the *IEEE International Symposium on Cluster Computing and the Grid (CCGrid '06)*, Singapore, May 2006.

ICPP 2006: *35th International Conference on Parallel Processing*, Columbus, OH, USA, August 2006.

ISPA 2006: *Fourth International Symposium on Parallel and Processing and Applications*, Sorrento, Italy, December 2006.

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

ICGNS 2005: *International Conference on GRID Networking and Services*, San Jose, Silicon Valley, California, September, 2005.

ISPDC 2005: *International Symposium on Parallel and Distributed Computing*, Lille, France, July 2005.

Th. Priol was member of the Scientific Committee of the *European Grid Conference*, Amsterdam, The Netherlands, February 2005.

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

He served in the Program Committees of the following conferences:

CCGRID 2005: *IEEE International Symposium on Cluster Computing and the Grid*, Cardiff, UK, May 2005.

e-Science 2005: *1st IEEE International Conference on e-Science and Grid Computing*, Melbourne, Australia, December 2005.

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

GRID 2005: *6th IEEE/ACM Intl. Workshop on Grid Computing*, Seattle, USA, November 2005.

HPCC 2005: *The 2005 International Conference on High Performance Computing and Communications*, Sorrento, Italy, September 2005.

WPGaM 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.

ISSGC 2005: *3rd International Summer School for Grid Computing*, Vico Equense, Italy, July 2005.

ISPDC 2005: *The 4th International Symposium on Parallel and Distributed Computing*, Lille, France, July 2005.

WI 2005: *The IEEE/WIC/ACM International Conference on Web Intelligence, and the International Conference on Intelligent Agent Technology*, Compiègne, France, September 2005.

WDMG 2005: *VLDB Workshop on Data Management in Grids*, Trondheim, Norway, September 2005.

CCGRID06: *IEEE International Symposium on Cluster Computing and the Grid*, Singapore, May 2006.

HPDC-15: *15th IEEE International Symposium on High-Performance Distributed Computing*, Paris, France, June 2006.

WHPGC 2006: *IPDPS Workshop on High Performance Grid Computing*, Rhode Island, Greece, April 2006.

VECPAR 2006: *The 7th International Meeting on high performance computing for computational science*, Rio de Janeiro, Brasil, July 2006.

E. Rivière was the publicity chair and one of the main three organizing team coordinators of the third edition of the french-speaking conference MajecSTIC'05 to be held in Rennes in November 2005.

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

COSET-2: *Second International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters* held in Cambridge, MA, USA, June 2005.

Symposium OSCAR: *Third Annual OSCAR Symposium* held in Guelph, Canada, May 2005.

9.1.3. Evaluation committees, consulting

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

J-P. Banâtre has been member of an *Expert Group* convened by the DG Information Society of the European Commission (EC) to outline a vision for *Software, Grids, Security and Dependability* that was held in Brussels on June 2005. He has been member of the NGG3 panel of experts to outline a vision for Grid research priorities over the period of 2005–2010.
He gave a presentation on Software Intensive systems, in the framework of the ERCIM Beyond the Horizon Workshop, Coblenz, 9–10 December 2005.

L. Bougé served in the *Expert Group* convened by the Ministry of Research (MSTP 9) in July 2005 to review the application for the Doctoral Supervision and Research Awards (*Primes d'encadrement doctoral et de recherche*, PEDR).
He serves in the *Scientific Committees* of two scientific programs launched by the newly-founded *National Research Agency* (Agence nationale la recherche, ANR): Advanced Research Action on *Data Masses, Modelisation, Simulation, Applications (MDMSA)* and Program on *High-Performance Computing and Computational Grids (Calcul intensif et grilles de calcul)*.

A.-M. Kermarrec is a member of the committee *Prix de thèse Spécif*, 2005.
She served as a reviewer of the EVERGROW IP EC-funded project.
She acted as an evaluator for the FP6 CALL 4 S.O 2.4.6 *Networked Audio/Visual Systems and Home platform*.
She is the secretary of the European ACM Chapter of the SIGOPS, EuroSys (until October 2005).
She acted as a referee for the foreign PhD committees of Sidath Handurukande from EPFL, Switzerland and Costa Paolo from Polytechnic University of Milano, Italy.

Ch. Pérez served as an expert for the RNTL program of the Ministry of Research and Industry.
He acted as an expert for the program *High performance computing and computational grids* of the French National Research Agency (GIP ANR).

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. He participated to the 3rd meeting that was held in Brussels on September 2005.
He is member of the Scientific Committee for the program *High Performance Computing and Computational Grids* of the French National Research Agency (GIP ANR).
He is member of the External Advisory Committee of the EU-funded EGEE project.
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) and the LIP (Lyon) research laboratories.
He was an evaluator for The Netherlands Organization for Scientific Research (NWO).

G. Vallée is a member of the Experts Management Module for FP6 of the European Commission. He has been selected for the technical review of a European Project.

9.2. Academic teaching

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

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 and ENS CACHAN.

R. Lottiaux gave a tutorial on Kerrighed Operating System at École Normale Supérieure de Lyon in April 2005.

Ch. Morin is responsible for a graduate teaching module *High Performance Computing on Clusters and Grids* of the Master Program in Computer Science, UNIVERSITY RENNES 1. Within this module, she gave lectures on distributed operating systems for clusters.

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.

MDP2P 2005. G. Antoniu was invited to give a talk at the *Journées Masses de Données Pair-à-Pair* in Orsay, March 2005.

Workshop on Distributed Algorithmics. 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?*.

Tsinghua University. J.-P. Banâtre gave a presentation entitled *A Chemical Model of Computation* in Beijing, China, 2005.

École des Jeunes Chercheurs en Programmation. J.-P. Banâtre gave an invited presentation entitled *A Higher-Order Chemical Language* in Dinard, June, 2005.

Libr'East 2005. Pascal Gallard gave an invited talk entitled *Kerrighed : un système d'exploitation pour grappes Linux* in Marne-la-Vallée, April 2005.

Algotel 2005. A.-M. Kermarrec gave an invited talk at Algotel 2005 *Septièmes Rencontres Francophones sur les aspects Algorithmiques des Télécommunications* in May 2005, Presqu'île de Giens, France on *Peer sharing behavior in the eDonkey network and Design of Server-less File Sharing Systems*

Locality 2005. A.-M. Kermarrec gave an invited talk at Locality 2005 *Locality Preserving Distributed Computing Methods* workshop in September 2005, Krakow, Poland *Gossip-based Overlay Network for Efficient Content-based Filtering*. She also gave this talk at the one day workshop “Systèmes Dynamiques”in Rennes in September 2005.

Recap 2005. A.-M. Kermarrec gave an invited talk at the CNRS-RECAP Workshop *Premier workshop CNRS RECAP - Réseaux de capteurs*, in Nice in November 2005 on *Peer sampling Service*.

Fosdem 2005. Renaud Lottiaux gave an invited talk entitled *Kerrighed: an Overview* at the Fosdem meeting in Brussels, February 2005.

Graphotec 2005. Renaud Lottiaux gave an invited talk entitled *Kerrighed: an Overview* at the Graphotec meeting in Rennes, February 2005.

RMLL 2005. Renaud Lottiaux and Pascal Gallard gave an invited talk entitled *Kerrighed: a Single System Image System for High Performance Computing on Linux Clusters* at the Rencontre Mondiale du Logiciel Libre (RMLL) meeting in Dijon, July 2005.

RMLL 2005. Christine Morin gave an invited talk entitled *Cluster Single System Image Operating Systems: a State of the Art* at the Rencontre Mondiale du Logiciel Libre (RMLL) meeting in Dijon, July 2005.

Euro-Par 2005. Renaud Lottiaux and Christine Morin gave a tutorial entitled *Kerrighed, a Single System Image Cluster Operating System* at the EuroPar conference in Lisbon, August 2005.

Bull. Pascal Gallard, Renaud Lottiaux and Christine Morin presented the Kerrighed system to Bull research and Development unit, Les Clayes sous Bois, February 2005.

Linux Networx. Pascal Gallard, Renaud Lottiaux, Christine Morin and Geoffroy Vallée presented the Kerrighed system to Linux Networx, Clamart, February 2005.

COREGRID Workshop. Christine Morin presented a talk entitled *A Grid OS for High Performance Computing on Cluster Federations*, at the CoreGrid workshop on Network-Centric Operating Systems, in Brussels, March 2005.

SOS9. Christine Morin was invited to participate as a panelist on *Scalable, high performance Linux: dream or reality?* to the SOS 9 workshop on science and supercomputing held in Davos, Switzerland, March 2005.

IngéDev 2005. Christine Morin gave an invited presentation entitled *Kerrighed/SSI-OSCAR : retour d’expérience d’un projet Open Source* at the journées des ingénieurs de Développement et plates-formes expérimentales, Rennes, April 2005.

ADIS. Christine Morin gave an invited presentation entitled *Kerrighed : un système à image unique pour le calcul haute performance sur grappes Linux* at the ADIS meeting, held at DGA, Arcueil, September 2005.

SC ’05. Pascal Gallard, Renaud Lottiaux, Christine Morin and Geoffroy Vallée gave a tutorial entitled *Scalable SSI Clustering with Kerrighed* at SC ’05. in Seattle, USA, November 2005.

CINES. Christine Morin gave an invited talk entitled *Etude comparative des systèmes à image unique (SSI) sur grappes Linux* at CINES in Montpellier, March 2005.

Café des sciences. Christine Morin participated to a Café des sciences meeting organized by the Espace des Sciences in Rennes, October 2005.

Joint ORNL/IU Workshop on Computational Frameworks in Fusion, ORNL, Oak Ridge, TN, USA. Ch. Pérez gave an invited talk entitled *Defining, Implementing, Executing and Deploying a Parallel Component Model* at the Joint ORNL/IU Workshop on *Computational Frameworks in Fusion* at ORNL, organized by ORNL and Indiana University, June 2005.

Journée sur les outils de calculs et de communication en HPC. Ch. Pérez gave an invited talk about Padico at a seminar organized by *Structuration du calcul hautes performances en sciences pour l'ingénieur* which involves four laboratories (IUSTI, IRPHE, LMA, MSMN-GP). Marseille, France, November 2005.

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

GridCoord Workshop. Th. Priol gave an invited presentation on CoreGRID at the GridCoord Industrial Workshop and the 8th HLRS Metacomputing and Grid Workshop. Stuttgart, Germany, March 2005.

Journées Thématiques CINES. Th. Priol was the responsible of a panel session on Clusters for HPC. Montepellier, France, March 2005.

GRIDLBIO. Th. Priol gave a presentation on Grid research in Europe and France. Lyon, France, June 2005.

ISPDC 2005. Th. Priol gave an invited presentation on CoreGRID at 4th International Symposium on Parallel and Distributed Computing. Lille, France, July 2005.

2nd French-German workshop. Th. Priol gave a presentation related to single system image operating systems for clusters. Postdam, Germany, July 2005.

CoreGRID Summer School. Th. Priol gave an invited presentation entitled *Main current EU projects on Grid*. Lausanne, Switzerland, September 2005.

GRID SUMMIT. Th. Priol gave a presentation entitled *Le Grid Computing : Une nouvelle approche pour virtualiser la puissance informatique ?*. Paris, France, October 2005.

9.4. Administrative responsibilities

F. André is the vice-chair of the Administrative Committee of IFSIC, the Computer Science department of Rennes 1 University.

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.

A.-M. Kermarrec is responsible of the local IRISA International Relations Committee (*Correspondant International en UR* (INRIA).
She is an elected member of the INRIA Evaluation Committee since September 2005.
She was a member of the local IRISA Communication Committee (until October 2005)
She is a member of the working group *Prospective* of the INRIA *Conseil d'Orientation Scientifique et Technologique*.

Ch. Morin has been a member of the INRIA Evaluation Committee until September 2005. She was a member of the 2005 Selection Committee for the Junior Researcher permanent positions (CR2) at the INRIA Futurs and Rocquencourt Research Units, and of the 2005 Selection Committee for the Senior Researcher permanent positions (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.

9.5. Miscellaneous

F. André is a member of the Selection Committee (*Commission de spécialistes*, CSE) of IFSIC (Computer Science department of Rennes 1 University), of the Computer Science department of INSA of Rennes and of the Computer Science group of University of Rennes 2.

L. Bougé is a member of the Project-Team Committee of IRISA, 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 Selection Committee (*Commission de Spécialistes*, CSE) for computer Science of ENS CACHAN.

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).

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 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.

T. Priol is a member of the Project-Team Committee of IRISA.

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. LE MÉTAYER. *Programming by Multiset Transformation*, in "Communications of the ACM", vol. 36, n^o 1, January 1993, p. 98–111.

[4] M. CASTRO, P. DRUSCHEL, A.-M. KERMARREC, A. NANDI, A. ROWSTRON, A. SINGH. *SplitStream: High-Bandwidth Multicast in Cooperative Environments*, in "Symposium on Operating System principles (SOSP 2003), Bolton Landing, NY", October 2003, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/CasDruKerNanRowSin03SOSP.pd

[5] 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.

[6] P. Eugster, P. Felber, R. Guerraoui, A.-M. Kermarrec. *The Many Faces of Publish/Subscribe*, in "ACM computing Surveys", vol. 35, nº 2, June 2003, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/EugFelGueKer03ACMSur

[7] A.-M. Kermarrec, C. Morin, M. Banâtre. *Design, Implementation and Evaluation of ICARE*, in "Software Practice and Experience", nº 9, 1998, p. 981–1010.

[8] T. Kielmann, P. Hatcher, L. Bougé, H. Bal. *Enabling Java for High-Performance Computing: Exploiting Distributed Shared Memory and Remote Method Invocation*, in "Communications of the ACM", Special issue on Java for High Performance Computing, vol. 44, nº 10, October 2001, p. 110–117.

[9] Z. Lahjomri, T. Priol. *KOAN: A Shared Virtual Memory for iPSC/2 Hypercube*, in "Proc. of the 2nd Joint Int'l Conf. on Vector and Parallel Processing (CONPAR'92)", Lecture Notes in Computer Science, vol. 634, Springer Verlag, September 1992, p. 441–452, http://www.inria.fr/rrrt/rr-1634.html.

[10] 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

[11] J.-P. Banâtre, J.-L. Giavitto, P. Fradet, O. Michel (editors). *Unconventional Programming Paradigms, International Workshop UPP 2004, Le Mont Saint Michel, France, September 15-17, 2004, Revised Selected and Invited Papers*, Lecture Notes in Computer Science, vol. 3566, Springer, Le Mont-Saint-Michel, France, 2005.

[12] 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.

[13] P. M. A. Sloot, A. G. Hoekstra, T. Priol, A. Reinefeld, M. Bubak (editors). *Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Revised Selected Papers*, Lecture Notes in Computer Science, vol. 3470, Springer, 2005.

Doctoral dissertations and Habilitation theses

[14] S. Lacour. *Contribution à l'automatisation du déploiement d'applications sur des grilles de calcul*, in French, Thèse de Doctorat, Université de Rennes 1, IRISA, Rennes, France, December 2005.

[15] L. Rilling. *Système d'exploitation à image unique pour une grille de composition dynamique : conception et mise en oeuvre de services fiables pour exécuter les applications distribuées partageant des données*, In French, Thèse de doctorat, Université de Rennes 1, IRISA, Rennes, France, November 2005.

Articles in refereed journals and book chapters

[16] G. Antoniu, M. Bertier, E. Caron, F. Desprez, L. Bougé, M. Jan, S. Monnet, P. Sens. *Future Generation Grids*, V. Getov, D. Laforenza, A. Reinefeld (editors). , CoreGRID, chap. GDS: An Architecture Proposal for a grid Data-Sharing Service, Springer, 2006.

[17] G. Antoniu, L. Bougé, M. Jan. *JuxMem: An Adaptive Supportive Platform for Data Sharing on*

the Grid, in "Scalable Computing: Practice and Experience", vol. 6, n^o 3, November 2005, p. 45–55, http://www.irisa.fr/paris/Biblio/Papers/Antoniu/AntBouJan05SCPE.pdf.

[18] 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, 2005, http://www.inria.fr/rrrt/rr-5082.html.

[19] G. ANTONIU, J.-F. DEVERGE, S. MONNET. *How to bring together fault tolerance and data consistency to enable grid data sharing*, in "Concurrency and Computation: Practice and Experience", n^o 17, September 2005, p. 1–19, http://www.interscience.wiley.com.

[20] M. BERTIER, L. ARANTES, P. SENS. *Distributed Mutual Exclusion Algorithms for Grid Applications: a Hierarchical Approach*, in "To appear Journal of Parallel and Distributed Computing (JPDC)", 2005, http://www.irisa.fr/paris/Biblio/Papers/Bertier/BerAraSen05JPDC.pdf.

[21] A. DENIS, S. LACOUR, C. PÉREZ, T. PRIOL, A. RIBES. *Engineering the Grid: status and perspective*, B. D. MARTINO, J. DONGARRA, A. HOISIE, L. T. YANG, H. ZIMA (editors). , ISBN: 1-58883-038-1, chap. Programming the Grid with components: models and runtime issues, American Scientific Publishers, 2005.

[22] I. GUPTA, A.-M. KERMARREC, A. GANESH. *Efficient and Adaptive Epidemic-style Protocols for Reliable and Scalable Multicast*, in "IEEE Transactions on Parallel and Distributed Systems", To appear, 2005, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/GupKerGan05tpds.pdf.

[23] P. HATCHER, M. RENO, G. ANTONIU, L. BOUGÉ. *Cluster Computing with Java*, in "Computing in Science and Engineering (CISE)", IEEE Computer Society, March 2005, p. 22–27, http://csdl.computer.org/comp/mags/cs/2005/02/c2toc.htm.

[24] C. MORIN, A. DENIS, R. NAMYST, O. AUMAGE, R. LOTTIAUX. *Encyclopédie de l'informatique et des systèmes d'information*, J. AKOKA, I. COMYN-WATTIAU (editors). , Collection informatique, to appear, vol. Architectures et systèmes distribués, chap. Des réseaux de calculateurs aux grilles de calcul, Vuibert, 2006.

[25] F. SULTAN, A. BOHRA, P. GALLARD, I. NEAMTIU, S. SMALDONE, Y. PAN, L. IFTODE. *Recovering Internet Services from Operating System Failures*, in "IEEE Internet Computing", March 2005, http://www.cs.rutgers.edu/~iftode/recovery05.pdf.

[26] A. C. VIANA, M. D. AMORIM, S. FDIDA, J. F. REZENDE. *Self-organization in spontaneous networks: the approach of DHT-based routing protocols*, in "Ad Hoc Networks Journal, special issue on Data Communications and Topology Control in Ad Hoc Networks", vol. 3, n^o 5, September 2005, p. 589–606, http://www.irisa.fr/paris/Biblio/Papers/Viana/ViAmoFdiRez05AdHocNetJ.pdf.

Publications in Conferences and Workshops

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

[28] F. ANDRÉ, J. BUISSON, J.-L. PAZAT. *Dynamic adaptation of parallel codes: toward self-adaptable*

components for the Grid, in "Component Models and Systems for Grid Applications", V. Getov, T. Kielmann (editors). , Proceedings of the Workshop on Component Models and Systems for Grid Applications held June 26, 2004 in Saint Malo, France, Springer, June 2005, p. 145–156, http://www.irisa.fr/paris/Biblio/Papers/Buisson/AndBuiPaz04CMSGA.pdf.

[29] G. Antoniu, P. Hatcher, M. Jan, D. A. Noblet. *Performance Evaluation of JXTA Communication Layers*, in "Proc. Workshop on Global and Peer-to-Peer Computing (GP2PC 2005), Cardiff, UK", Held in conjunction with the 5th IEEE/ACM Int. Symp. on Cluster Computing and the Grid (CCGRID 2005) - IEEE TFCC, May 2005, http://www.irisa.fr/paris/Biblio/Papers/Antoniu/AntHatJanNob05GP2PC.pdf.

[30] G. Antoniu, M. Jan, D. A. Noblet. *Enabling the P2P JXTA Platform for High-Performance Networking Grid Infrastructures*, in "Proc. of the first Intl. Conf. on High Performance Computing and Communications (HPCC '05), Sorrento, Italy", Lect. Notes in Comp. Science, n^{o} 3726, Springer-Verlag, September 2005, p. 429–440.

[31] J.-P. Banâtre, P. Fradet, Y. Radenac. *Higher-order Chemical Model of Computation*, in "The Grand Challenge in Non-Classical Computation", April 2005, http://www.cs.york.ac.uk/nature/workshop/papers/BanatreFradetRadenac.p

[32] J.-P. Banâtre, P. Fradet, Y. Radenac. *Higher-order Chemical Programming Style*, in "Unconventional Programming Paradigms (UPP'04)", J.-P. Banâtre, J.-L. Giavitto, P. Fradet, O. Michel (editors). , Lecture Notes in Computer Science, vol. 3566, Springer, 2005, p. 84–95.

[33] J.-P. Banâtre, P. Fradet, Y. Radenac. *Principles of Chemical Programming*, in "Proceedings of the 5th International Workshop on Rule-Based Programming", S. Abdennadher, C. Ringeissen (editors). , Electronic Notes in Theoretical Computer Science, vol. 124, n^{o} 1, Elsevier, June 2005, p. 133–147, http://www.sciencedirect.com/science/journal/15710661/.

[34] J.-P. Banâtre, P. Fradet, Y. Radenac. *Towards Grid Chemical Coordination*, in "Proceedings of Symposium on Applied Computing (SAC'06)", (to appear), 2005.

[35] J.-P. Banâtre, P. Fradet, Y. Radenac. *A Generalized Higher-Order Chemical Computation Model*, in "International Workshop on Developments in Computational Models", ENTCS, (to appear), Elsevier, 2006, http://www.cs.york.ac.uk/nature/workshop/papers/BanatreFradetRadenac.pdf.

[36] J.-P. Banâtre, J.-L. Giavitto, P. Fradet, O. Michel. *Challenging Questions for the Rationals of Non-Classical Programming Languages*, in "The Grand Challenge in Non-Classical Computation International Workshop", April 2005, http://www.cs.york.ac.uk/nature/workshop/papers/Michel.pdf.

[37] A. Beloued, J.-M. Gilliot, M.-T. Segarra, F. André. *Dynamic Data Replication and Consistency in Mobile Environment*, in "2nd International Doctoral Symposium on Middleware'05", ACM Press, November 2005.

[38] J. Buisson, F. André, J.-L. Pazat. *Enforcing consistency during the adaptation of a parallel component*, in "The 4th International Symposium on Parallel and Distributed Computing", July 2005, http://www.irisa.fr/paris/Biblio/Papers/Buisson/BuiAndPaz05ISPDC.pdf.

[39] J. BUISSON, F. ANDRÉ, J.-L. PAZAT. *A framework for dynamic adaptation of parallel components*, in "ParCo 2005", 2005, http://www.irisa.fr/paris/Biblio/Papers/Buisson/BuiAndPaz05PARCO.pdf.

[40] J. BUISSON, F. ANDRÉ, J.-L. PAZAT. *Dynamic adaptation for Grid computing*, in "Advances in Grid Computing - EGC 2005 (European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Revised Selected Papers), Amsterdam", P. SLOOT, A. HOEKSTRA, T. PRIOL, A. REINEFELD, M. BUBAK (editors). , LNCS, vol. 3470, Springer-Verlag, June 2005, p. 538–547, http://www.irisa.fr/paris/Biblio/Papers/Buisson/BuiAndPaz05EGC.pdf.

[41] J. BUISSON. *Un modèle pour l'adaptation dynamique des programmes parallèles*, in "Rencontres Francophones en Parallélisme, Architecture, Système et Composant : RenPar'16, 16ème édition des Rencontres Francophones du Parallélisme", session travail en démarrage, April 2005, p. 225–230, http://www.irisa.fr/paris/Biblio/Papers/Buisson/Bui05RenPar.pdf.

[42] Y. BUSNEL. *Prise en compte d'aspects sémantiques dans la contruction d'un réseau pair-à-pair.*, in "MAnifestation des JEunes Chercheur francophones dans les domaines des STIC (MajecSTIC'05), Rennes, France", November 2005, http://www.irisa.fr/paris/Biblio/Papers/Busnel/ybusnel-majecstic05.pdf.

[43] F. CAPPELLO, F. DESPREZ, M. DAYDE, E. JEANNOT, Y. JÉGOU, S. LANTÉRI, N. MELAB, R. NAMYST, P. PRIMET, O. RICHARD, E. CARON, J. LEDUC, G. MORNET. *Grid'5000: A Large Scale, Reconfigurable, Controlable and Monitorable Grid Platform*, in "Proc. Grid 2005 - 6th IEEE/ACM International Workshop on Grid Computing, Seattle, Washington, USA", held in conjunction with SC|05, November 2005.

[44] L. CUDENNEC, S. MONNET. *Extension du modèle de cohérence à l'entrée pour la visualisation dans les applications de couplage de code sur grilles*, in "Actes des Journées francophones sur la Cohérence des Données en Univers Réparti, Paris", À paraître, November 2005.

[45] J.-F. DEVERGE, S. MONNET. *Cohérence et volatilité dans un service de partage de données dans les grilles de calcul*, in "Actes des Rencontres francophones du parallélisme (RenPar 16), Le Croisic", April 2005, p. 47–55, http://www.irisa.fr/paris/Biblio/Papers/Monnet/DevMon05RENPAR.pdf.

[46] M. FERTRÉ, C. MORIN. *Extending a Cluster SSI OS for Transparently Checkpointing Message-Passing Parallel Applications*, in "International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN05), Las Vegas, Nevada, USA", October 2005, http://www.irisa.fr/paris/Biblio/Papers/Fertre/FerMor05ISPAN.pdf.

[47] M. FERTRÉ, C. MORIN. *Transparent Message-Passing Parallel Applications Checkpointing in Kerrighed*, in "High Availability and Performance Computing Workshop 2005 (HAPCW05), Santa Fe, New Mexico, USA", October 2005, http://www.irisa.fr/paris/Biblio/Papers/Fertre/FerMor05HAPCW.pdf.

[48] S. FRENZ, R. LOTTIAUX, M. SCHOETTNER, C. MORIN, R. GOECKELMANN, P. SCHULTHESS. *A Practical Comparison of Cluster Operating Systems Implementing Sequential and Transactional Consistency*, in "Proceeding of the 6th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Melbourne, Australia", LNCS, October 2005, http://www.irisa.fr/paris/Biblio/Papers/Lottiaux/FreLotSchMorGoeSchICA3PP.pdf.

[49] S. LACOUR, C. PÉREZ, T. PRIOL. *Generic Application Description Model: Toward Automatic*

Deployment of Applications on Computational Grids, in "6th IEEE/ACM International Workshop on Grid Computing (Grid2005), Seattle, WA, USA", Springer-Verlag, November 2005, http://www.irisa.fr/paris/Biblio/www/Year/2005.html.

[50] R. LOTTIAUX, B. BOISSINOT, P. GALLARD, G. VALLÉE, C. MORIN. *OpenMosix, OpenSSI and Kerrighed: A Comparative Study*, in "Proceeding of IEEE International Symposium on Cluster Computing and the Grid (CCGrid '05), Cardiff, UK", May 2005, http://www.irisa.fr/paris/Biblio/Papers/Lottiaux/LotBoiGalValMor05CCGrid.pdf.

[51] Z. NÉMETH, C. PÉREZ, T. PRIOL. *Workflow Enactment Based on a Chemical Metaphor*, in "The 3rd IEEE International Conference on Software Engineering and Formal Methods", IEEE Computer Society Press, September 2005.

[52] L. RILLING, C. MORIN. *A Practical Transparent Data Sharing Service for the Grid*, in "Proc. Fifth International Workshop on Distributed Shared Memory (DSM 2005), Cardiff, UK", Held in conjunction with CCGrid 2005, May 2005, http://www.irisa.fr/paris/Biblio/Papers/Rilling/RilMor05DSM.pdf.

[53] L. RILLING, C. MORIN. *Partage de données transparent et tolérant aux fautes pour la grille*, in "Actes de la 4ème Conférence Française sur les Systèmes d'Exploitation (CFSE 4), Le Croisic, France", April 2005, p. 135–146, http://www.irisa.fr/paris/Biblio/Papers/Rilling/RilMor05CFSE.pdf.

[54] G. VALLÉE, J.-Y. BERTHOU, H. PRISKER, D. LEPRINCE. *OSCAR on Debian: the EDF Experience*, in "The 3rd Annual OSCAR Symposium, University of Guelph, Guelph, Ontario, Canada", Held in conjunction with the 19th International Symposium on High Performance Computing Systems and Applications (HPCS 2005), May 2005, http://www.irisa.fr/paris/Biblio/Papers/Vallee/oscar05.pdf.

[55] G. VALLÉE, R. LOTTIAUX, D. MARGERY, C. MORIN, J.-Y. BERTHOU. *Ghost Process: a Sound Basis to Implement Process Duplication, Migration and Checkpoint/Restart in Linux Clusters*, in "The 4th International Symposium on Parallel and Distributed Computing, Lille, France", July 2005, p. 97–104, http://www.irisa.fr/paris/Biblio/Papers/Vallee/ispdc05.ps.

[56] G. VALLÉE, C. MORIN, S. L. SCOTT. *A Framework for High Availability Based on a Single System Image*, in "HAPCW'05: High Availability and Performance Computing Workshop", Held in conjunction with LACSI 2005, October 2005, http://www.irisa.fr/paris/Biblio/Papers/Vallee/hapcw05-gvallee.ps.

[57] G. VALLÉE, S. L. SCOTT, C. MORIN, J.-Y. BERTHOU, H. PRISKER. *SSI-OSCAR: a Cluster Distribution for High Performance Computing Using a Single System Image*, in "The 3rd Annual OSCAR Symposium, University of Guelph, Guelph, Ontario, Canada", Held in conjunction with the 19th International Symposium on High Performance Computing Systems and Applications (HPCS 2005), May 2005, http://www.irisa.fr/paris/Biblio/Papers/Vallee/oscar05-ssi.ps.

[58] G. VAYSSE, F. ANDRÉ, J. BUISSON. *Using Aspects for Integrating a Middleware for Dynamic Adaptation*, in "The First Workshop on Aspect-Oriented Middleware Development (AOMD'05)", ACM Press, November 2005, http://www.irisa.fr/paris/Biblio/Papers/Buisson/VayAndBui05AOMD.pdf.

[59] A. C. VIANA, M. D. AMORIM, Y. VINIOTIS, S. FDIDA, J. F. REZENDE. *Easily-managed and topological-independent location service for self-organizing networks*, in "ACM MobiHoc 2005 Conference, Urbana-Champaign, IL", May 2005, http://www.irisa.fr/paris/Biblio/Papers/Viana/ViAmoViFdiRez05Mobihoc.pdf.

[60] A. C. VIANA, M. D. AMORIM, Y. VINIOTIS, S. FDIDA, J. F. REZENDE. *Easily-Managed Location in SONs by exploiting Space-Filling Curves*, in "Infocom Student Workshop, Miami, FL", March 2005, http://www.irisa.fr/paris/Biblio/Papers/Viana/ViAmoViFdiRez05Infocom.pdf.

Internal Reports

[61] M. ALDINUCCI, S. CAMPA, M. COPPOLA, M. DANELUTTO, C. ZOCCOLO, F. ANDRÉ, J. BUISSON. *Parallel program/component adaptivity management*, Technical report, n° TR-0014, CoreGRID, September 2005, http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0014.pdf.

[62] G. ANTONIU, J.-F. DEVERGE, S. MONNET. *How to bring together fault tolerance and data consistency to enable grid data sharing*, To appear in Concurrency and Computation: Practice and Experience, Research Report, n° RR-5467, INRIA, IRISA, Rennes, France, January 2005, http://www.inria.fr/rrrt/rr-5467.html.

[63] G. ANTONIU, P. HATCHER, M. JAN, D. A. NOBLET. *Performance Evaluation of JXTA Communication Layers (extended version)*, Research Report, n° RR-5469, INRIA, IRISA, Rennes, France, January 2005, http://www.inria.fr/rrrt/rr-5469.html.

[64] G. ANTONIU, M. JAN, D. A. NOBLET. *A practical example of convergence of P2P and grid computing: an evaluation of JXTA's communication performance on grid networking infrastructures*, Submitted to a special issue of JOGC, Research Report, n° RR-5718, INRIA, IRISA, Rennes, France, November 2005, http://www.inria.fr/rrrt/rr-5718.html.

[65] G. ANTONIU, M. JAN, D. A. NOBLET. *Enabling JXTA for High Performance Grid Computing*, Submitted to HPCC 2005, Research Report, n° RR-5488, INRIA, IRISA, Rennes, France, February 2005, http://www.inria.fr/rrrt/rr-5488.html.

[66] J.-P. BANÂTRE, P. FRADET, Y. RADENAC. *Generalized Multisets for Chemical Programming*, Research Report, n° RR-5743, INRIA, Rennes, IRISA, Rennes, France, November 2005, http://www.inria.fr/rrrt/rr-5743.html.

[67] J. BUISSON, F. ANDRÉ, J.-L. PAZAT. *A negotiation-based approach of consistency for dynamic adaptation*, Technical report, n° PI-1700, IRISA, Campus universitaire de Beaulieu, avenue du Général Leclerc, 35042 Rennes CEDEX, March 2005, http://www.irisa.fr/paris/Biblio/Papers/Buisson/BuiAndPaz05PI1700.ps.

[68] Y. BUSNEL, A.-M. KERMARREC. *Integrating File Popularity and Peer Generosity in Proximity Measure for Semantic-based Overlays.*, Research Report, n° RR-5731, INRIA, IRISA, Rennes, France, October 2005, http://www.inria.fr/rrrt/rr-5731.html.

[69] R. GUERRAOUI, S. HANDURUKANDE, A.-M. KERMARREC, F. LE FESSANT, E. RIVIERE. *GosSkip: a Gossip-based Structured Overlay Network for Efficient Content-based Filtering*, Research Report, n° 020 200495, EPFL, Lausanne, Switzerland, 2005, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/GueHanKerFesRiv05rr.pdf.

[70] S. HANDURUKANDE, A.-M. KERMARREC, F. LE FESSANT, L. MASSOULIÉ, S. PATARIN. *Peer Sharing Behaviour in the eDonkey Network, and Implications for the Design of Server-less File Sharing Systems*, Research Report, n° 5506, Inria, February 2005, http://www.inria.fr/rrrt/rr-5506.html.

[71] S. Lacour, C. Pérez, T. Priol. *Description and Packaging of MPI Applications for Automatic Deployment on Computational Grids*, Research Report, n^o RR-5582, INRIA, IRISA, Rennes, France, May 2005, http://www.inria.fr/rrrt/rr-5582.html.

[72] S. Lacour, C. Pérez, T. Priol. *Generic Application Description Model: Toward Automatic Deployment of Applications on Computational Grids*, Research Report, n^o RR-5733, INRIA, IRISA, Rennes, France, October 2005, http://www.inria.fr/rrrt/rr-5733.html.

[73] D. Margery, R. Lottiaux, C. Morin. *Capabilities for per Process Tuning of Distributed Operating Systems*, Rapport de Recherche, n^o RR-5411, INRIA, IRISA, Rennes, France, December 2004, http://www.inria.fr/rrrt/rr-5411.html.

[74] L. Massoulié, A.-M. Kermarrec, A. Ganesh. *Efficient application-level multicast on a network-aware self-organizing overlay*, Research Report, n^o 5502, INRIA, IRISA, Rennes, France, February 2005, http://www.inria.fr/rrrt/rr-5502.html.

[75] 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^o CS-04-09, Williams College Department of Computer Science, Williamstown, MA, USA, 2005, http://www.cs.williams.edu/~terescoj/research/publications/pp04/pp04.pdf.

[76] G. Vallée, J.-Y. Berthou, H. Prisker, D. Leprince. *OSCAR on Debian: the EDF Experience*, Research Report, n^o 5537, INRIA, IRISA, Rennes, France, March 2005, http://www.inria.fr/rrrt/rr-5537.html.

[77] G. Vallée, R. Lottiaux, D. Margery, C. Morin, J.-Y. Berthou. *Ghost Process: a Sound Basis to Implement Process Duplication, Migration and Checkpoint/Restart in Linux Clusters*, Research Report, n^o 5476, INRIA, IRISA, Rennes, France, January 2005, http://www.inria.fr/rrrt/rr-5476.html.

[78] G. Vallée, S. L. Scott, C. Morin, J.-Y. Berthou, H. Prisker. *SSI-OSCAR: a Cluster Distribution for High Performance Computing Using a Single System Image*, Research Report, n^o 5538, INRIA, IRISA, Rennes, France, March 2005, http://www.inria.fr/rrrt/rr-5538.html.

[79] 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^o 5510, INRIA, IRISA, Rennes, France, March 2005, http://www.inria.fr/rrrt/rr-5510.html.

Miscellaneous

[80] Y. Busnel. *Prise en compte de la proximité sémantique dans la restructuration de réseau logique pair à pair*, Rapport de stage de Master Recherche, June 2005, http://www.irisa.fr/paris/Biblio/Papers/Busnel/ybusnel-m2ri-report.pdf.

[81] B. Daix. *Entrées/Sorties à haute performance dans le système à image unique Kerrighed*, Rapport de stage MRI, École Normale Supérieure de Lyon, June 2005, http://www.irisa.fr/paris/Biblio/Papers/Daix/daix05es_krg.pdf.

[82] M. Fertré. *Intégration d'un mécanisme de reprise d'applications parallèles dans un système d'exploitation pour grappe*, Rapport de stage de Master Recherche, Université de Rennes 1, June 2005, http://www.irisa.fr/paris/Biblio/Papers/Fertre/Fer05M2RI.pdf.

[83] P. Gallard, R. Lottiaux, C. Morin. *Kerrighed V2.0 – Etude de la problématique de haute disponibilité dans le système Kerrighed*, November 2005, Livrable PEA COCA.

[84] E. Jeanvoine. *Distributed Operating System for Resource Discovery and Allocation in Federated Clusters*, October 2005, http://www.irisa.fr/paris/Biblio/Papers/Jeanvoine/Jea05PosterSOSP.pdf, Poster presented at the 20th ACM Symposium on Operating Systems Principles (SOSP 2005: Brighton, UK).

[85] C. Morin. *Kerrighed : système d'exploitation à image unique pour le calcul à haute performance sur grappe*, July 2005, http://www.irisa.fr/orap/Publications/Bi-orap/biorap-44.pdf, BIORAP.

[86] C. Morin. *XtreemOS: Building and Promoting a Linux-based Operating Ssystem to Support Virtual Organizations for Next Generation Grids*, Integrated Project (IP) proposal submitted to FP6-2005-IST-5 European program, September 2005.

[87] X. Muller, T. Priol. *La programmation des grilles informatiques*, October 2005, http://interstices.info, Interstices.

[88] X. Muller, T. Priol. *Les réseaux grillent les limitations*, October 2005, http://interstices.info, Interstices.

[89] T. Priol. *La Grille, superordinateur mondial*, 2005, Sciences, 2005-3.

[90] G. Vallée. *Conclusion of the PostDoctoral Program on SSI-OSCAR*, September 2005, http://www.irisa.fr/paris/Biblio/Papers/Vallee/Val05postdoc_conclusion.pdf.

[91] G. Vallée. *RPM for Kerrighed, Installation Guide*, March 2005, http://www.irisa.fr/paris/Biblio/Papers/Vallee/Val05RPM_install

[92] G. Vallée. *SSI-OSCAR Quick Installation Guide*, May 2005, http://www.irisa.fr/paris/Biblio/Papers/Vallee/Val05quick-install-ssioscar3.pdf.

Bibliography in notes

[93] I. Foster, C. Kesselman (editors). *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, 1998.

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

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

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

[97] 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.

[98] 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^o 1, January 2002, p. 43–54.

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

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

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

[102] A. J. GANESH, A.-M. KERMARREC, L. MASSOULIÉ. *Peer-to-Peer membership management for gossip-based protocols*, in "IEEE Transactions on Computers", vol. 52, n^o 2, February 2003, http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/GanKerMas03IEEETOC.pdf.

[103] 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.

[104] 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.

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

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

[107] 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.

[108] 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.

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

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

[111] 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.

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

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

[114] 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.

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

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

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

[118] A. ROWSTRON, P. DRUSCHEL. *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*, in "IFIP/ACM International Conference on Distributed Systems Platforms (Middleware, Heidelberg, Germany", 2001, p. 329-350.

[119] 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.

[120] I. STOICA, R. MORRIS, D. KARGER, F. KAASHOEK, H. BALAKRISHNAN. *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*, in "ACM SIGCOMM, San Diego, CA", 2001, p. 149-160.

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