INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Sardes

System Architecture for Reflective Distributed Computing Environments

Rhône-Alpes

THEME COM

Activity Report

2004

Table of contents

1. Team

Sardes is a project of Inria Rhône-Alpes and a research team of Imag-LSR (Software, Systems and Networks Laboratory), a joint research unit (UMR 5526) of Centre National de la Recherche Scientifique, Institut National Polytechnique de Grenoble and université Joseph Fourier.

Head of Project-team
Jean-Bernard Stefani [Inria Research Director, on leave from *corps des Télécommunications*]

Administrative Assistant
Valérie Gardès

Inria Personnel
Emmanuel Cecchet [Research Scientist]
Daniel Hagimont [Research Scientist]
Alan Schmitt [Research Scientist]

Faculty
Sara Bouchenak [associate professor, université Joseph Fourier]
Fabienne Boyer [associate professor, université Joseph Fourier]
Noël De Palma [associate professor, Institut National Polytechnique de Grenoble]
Sébastien Jean [associate professor, université Pierre Mendès France]
Gilles Kuntz [associate professor, université Joseph Fourier]
Sacha Krakowiak [professor, université Joseph Fourier]
Jacques Mossière [professor, Institut National Polytechnique de Grenoble]

Visiting Scientist
Slim Ben Atallah [École Nationale des Sciences de l'Informatique de Tunis, contract, till June 30, 2004]

Technical Staff
Matthieu Leclercq
Nicolas Modrzyk

Ph.D. Students
Takoua Abdellatif [industrial grant (Bull)]
Philippe Bidinger [Inria grant (contract)]
Jakub Kornaś [Inria grant (contract), from Oct. 1st, 2004]
Renaud Lachaize [government grant]
Oussama Layaida [Inria grant (contract)]
Ali Erdem Özcan [industrial grant (ST Microelectronics)]
Vivien Quéma [government grant]

2. Overall Objectives

2.1.1. Objectives.

The objective of the Sardes project is to develop concepts, methods, and tools to build distributed systems (including both infrastructures and applications) that are open, evolvable, and safe. The designers and developers of such systems are facing important challenges, among which scalability, administration, and quality of service. To respond to these challenges, the infrastructures and applications need to be *adaptable*. This may be achieved through reflexivity and component-based design.

The project has three main ambitions:

1. To identify architectural paradigms for the various aspects of adaptable distributed systems (naming, communications, resource management, fault tolerance, mobility); to elaborate the relevant design patterns and software frameworks, as well as programming and meta-programming models.

2. To develop software engineering techniques to build adaptable distributed systems, specially through the use of configurable components, reflexivity, and appropriate construction techniques (e.g. meta-programming and dynamic compilation).
3. To apply the concepts and techniques developed in the project to the prototyping of adaptable software infrastructures; to validate these concepts and techniques in various environments (clusters, large scale networks, embedded systems), and for various application domains (data management, multimedia, real time), with emphasis on scalability, administrability, and quality of service.

2.1.2. Organization and Collaborations.

The project is organized around two main activity domains: *models and tools for distributed components*; *autonomous distributed systems management*. The latter theme is taking an increasing importance. We have started a new project-wide activity whose aim is to design and implement an autonomic resource management system on a cluster, in order to ensure prescribed levels of availability and quality of service.

Sardes is involved in several industrial and international collaborations. It is active in the OBJECTWEB consortium (8.2) dedicated to open source middleware. It is a partner of several European projects and networks: Mikado, Ozone, CoreGrid, Gorda (IST program) and Osmose (ITEA program). It participates in two projects funded by the national research network on software technologies (RNTL): Arcad and Inside. It collaborates with several industrial partners: Bull, France Telecom, Microsoft, ST MicroElectronics, and has close links with Scalagent, a technology startup created by former members of the project.

3. Scientific Foundations

3.1. Introduction

In this section, we first present the main challenges that face the designers of large scale distributed systems. We then discuss recent advances and open problems in the two main areas covered by Sardes: component-based architectures for adaptable distributed systems, and distributed system management.

3.2. Challenges of Distributed Systems

The future[1] of information processing applications is envisioned as a range of environments in which processors will be everywhere and will be interconnected by an array of networks, from ad-hoc to the global Internet. Constructing the software base - the *middleware* - for such ubiquitous computing infrastructures poses a number of scientific and technical challenges, which arise from the wide variety of applications and services that need to be supported.

Software will run in a multitude of computing environments ranging from traditional desktops to multiple host, networked systems, to mobile systems, to embedded systems, to wearable computers. Software systems will need to be "aware" of their physical surroundings, taking input from real-world sensors and sending output intended to control the real-world environment. They will need to "understand" their proximity to other computing resources and use this information to guide their behavior. A number of application scenarios illustrating these new environments have been discussed in a recent report for the IST European program [42].

A fundamental requirement for such environments is for middleware architectures capable of supporting services that are *composable* and *adaptable*. We have singled out four major challenges: scalability, quality of service, manageability and programmability.

1. *Scalability* concerns the ability to scale in several dimensions; scaling in machine forms: from smart labels to server farms to meta-computing network overlays; scaling in numbers: objects, machines, users, locations; scaling in logical and organizational structures: from ad-hoc collaborations networks to federations of multi-domain enterprises.

[1] The text of 3.2 has been adapted from a Research Roadmap Report produced by the Midas IST project in which Sardes has participated.

2. *Quality of service (QoS)* concerns the ability to obtain correctness and service-level guarantees such as timeliness, availability, fault-tolerance, survivability for applications executing in large scale environments. The problem of meeting QoS requirements of applications is made harder in a ubiquitous computing environment where new services and customized services are expected to be added into (existing) applications at an alarming rate.
3. *Manageability* concerns the ability to monitor and to control the operation and evolution of large scale, long-lived distributed applications and services, avoiding manual intervention and centralization (i.e., assumption of one management domain). Distributed applications and services will need to be reconfigured dynamically, for example, to maintain user specified QoS guarantees despite changes in operating conditions (e.g., component failures). Mechanisms are needed to dynamically add, extend, remove or move component services in a dependable and predictable manner.
4. *Programmability* concerns the ability to compose new applications from existing applications and services, to deploy and to maintain them in highly dynamic and heterogeneous computing environments; further, applications are expected to be highly parallel, requiring high-level abstractions necessary for dealing with complexity, with fine-grained resource awareness built according to the end-to-end principle.

Middleware for a ubiquitous computing environment will by necessity be open and standardized, at least at the level of communication protocols, key application programming interfaces and language tools. Furthermore, such middleware will become an economic commodity, i.e. be available at very low or no cost in all manners of equipment and networks to serve as a viable basis for value-added services and applications. In this context, it makes both technical and economical sense to pursue the development of such middleware facilities using an open source model.

In the next two sections, we discuss specific aspects relevant to the current research interests of Sardes.

3.3. Architectures for Adaptable Distributed Systems

A first requirement for adaptable middleware is a modular architecture, in which the interfaces and dependencies between the parts should be clearly identified. The main motivations are to minimize the adaptation induced changes by defining components as adaptation localities, and to use structural modifications (e.g. reconfiguration) as a tool for adaptation. Thus research on adaptable middleware is essentially done in the context of component-based systems. This is the theme of Section 3.3.1.

Models, patterns and frameworks are the basic ingredients of an architectural approach to the design of complex software systems. These aspects are presented in Section 3.3.2.

3.3.1. Adaptable Component-based Middleware

Building an application out of composable parts is an old challenge of software engineering. Despite the apparent simplicity of its requirements, progress towards this goal has been slow. The notions related to components have only been elicited recently [61]. Although commercial infrastructures based on components are now available (e.g. COM+, CCM, EJB, .Net), the rigorous definition of a component model and of its architectural invariants is still an open issue.

Three main approaches are being used to achieve adaptability and separation of concerns in middleware systems: meta-object protocols, aspect-oriented programming, and pragmatic techniques.

Meta-object protocols [55] are the basic device of reflection. A reflective system is one that is able to answer questions about itself and to modify its own behavior, by providing a causally connected representation of itself to a meta-level. In a component infrastructure, the interface available at the meta-level allows the operations related to component lifecycle and composition to be dynamically adapted. A number of research prototypes (OpenORB [45], Flexinet, 2K/DynamicTAO) have been developed to investigate this approach.

The goal of aspect-oriented programming [56] (AOP) is to define methods and tools to better identify and isolate the code related to the various "aspects" present in an application, such as those related to extra-functional properties (security, fault tolerance, performance). This code is usually intertwined with that of the application proper, which is detrimental to easy change.

Pragmatic techniques usually rely on interception (interposing code at the interface between components, while preserving the interface). They are mainly used in commercial middleware.

The price of the increased flexibility brought by these adaptation techniques is paid in performance, due to the additional indirection or interpretation levels that these techniques introduce. Thus optimization techniques for reducing this performance penalty are an active subject of research. They include partial evaluation and program specialization [58], code injection (inlining), dynamic code generation and optimization [59].

3.3.2. Models, Patterns and Frameworks for Distributed Systems Architecture

We single out three aspects in the foundations of distributed systems architecture, in order of decreasing abstraction: defining formal or semi-formal models; identifying generic design patterns; developing the associated software frameworks.

Developing models for distributed systems is notoriously hard, compared to centralized systems. This is due to the increased complexity deriving from the asynchrony of the communication system, from fault tolerance requirements, and from mobility. Two main directions are relevant for our project: modeling software architecture, and specifically component-based systems (a survey of recent work is [57]); defining process calculi, specially for distributed and mobile processes (see survey in [46]). Our work aims at capturing those aspects of distributed systems that are relevant to dynamic adaptation, i.e. modularity, reconfigurability and mobility.

A *design pattern* [51] describes, in an articulated way, the structures, the representations and the techniques used to respond to specific requirements in a given context. A pattern captures in a generic, abstract form the experience acquired in the resolution of a class of related problems. A *software framework* [53] is a program skeleton that may be directly reused, or adapted according to well-defined rules, to solve a family of related problems. A framework usually embodies a pattern or a set of patterns.

Research on patterns for middleware has been very active since 1995, and a number of basic constructs have been identified [47][60]. Most middleware research prototypes are now developed as frameworks, in order to facilitate sharing and reuse.

Our own work is targeted towards global architectural patterns, such as naming and binding, components and reconfiguration, mobility and replication, transactions. These aspects have been less studied than the specific constructs used as building blocks. The results are transferred to the software platforms developed by OBJECTWEB (8.2).

3.4. Distributed System Administration

Administration is the function that aims at maintaining a system's ability to provide its specified services, with a prescribed quality of service. In general terms, administration may be viewed as a *control* activity, involving an event-reaction loop: the administration system detects events that may alter the ability of the administered system to perform its function, and reacts to these events by trying to restore this ability.

The operations performed under system and application administration include configuration and deployment, reconfiguration, resource management, observation and monitoring.

For about 15 years, the approach to system and network administration has been inspired by the manager-agent model, the base of widely used systems such as SNMP and CMIS/CMIP. The limitations of this model are now apparent, specially as regards its ability to scale up and to cope with the increased complexity and the dynamic evolution of the managed systems.

The next sections examine three specific aspects: the new requirements posed by event distribution and resource management, and the trend towards self-managed systems.

3.4.1. Event Channels

Many management functions (monitoring, resource management, fault tolerance) rely on *events*, propagated from the administered components of the system to the management components. In a large Internet-based system, these events may be generated at a very high rate. Thus efficient channels are needed to collect and to filter these events, and to propagate them to their recipients. This problem has emerged as a topic of active research. It is also related to the fast-developing area of sensor networks.

Event propagation usually follows the publish-subscribe pattern [50]. Scalable solutions, based on multicast groups, are known for publish-subscribe based on a fixed (or slowly changing) list of topics. However, designing an efficient publish-subscribe system based on the contents of the propagated messages is still an open problem. For Internet-scale systems, even filtering algorithms running in linear time with respect to the number of subscriptions are considered inefficient, and the goal is to design sub-linear algorithms. The Gryphon [43] and Siena [48] projects have made significant progress towards this goal.

The application of event channels to large scale observation and management is the theme of the Astrolabe [62] project.

3.4.2. Resource Management

The advent of resource-constrained applications, such as multimedia processing and real-time control, raised the need for delegating part of the resource management to the application, while leaving the operating system with the responsibility of fair overall resource sharing between its users. The rationale is that the application is in a better position to know its precise requirements and may dynamically adapt its demands to its needs, thus allowing global resource usage to be optimized, while guaranteeing a better service to each individual application.

The challenge is now to define relevant abstractions for resource management, both for the *resource principals*, i.e. the entities to which resources are allocated, and for the resources themselves. Thus, in the case of cluster computing, new abstractions such as "cluster reserves" and "virtual clusters", which group a dynamic set of cluster-wide resources, are being investigated. Another challenge is to define what part of resource management is delegated from the operating system to the application, and to specify the relevant interfaces.

Scheduler activations [44] is an example of early work in this respect in the area of multiprocessor scheduling. A more radical approach is illustrated by exo-kernels [49] in which the lower layer exports primitives allowing a kernel or application designer to define his own resource allocation policy. This is done at the expense of additional work, which may be alleviated by the use of appropriate frameworks, as done in THINK [7]. However, since exo-kernels give access to low level system entities, they are prone to security problems.

3.4.3. Autonomic Systems

Up to now, administration tasks have mainly been performed by persons. A great deal of the knowledge needed for administration tasks is not formalized and is part of the administrators' know-how and experience.

As the size and complexity of the systems and applications are increasing, the costs related to administration are taking up a major part of the total information processing budgets, and the difficulty of the administration tasks tends to approach the limits of the administrators' skills. As a consequence, there is a trend towards automating (at least in part) the functions related to administration. This is the goal of the so-called *autonomic computing* movement [54][52].

Autonomic computing aims at providing systems and applications with self-management capabilities, including self-configuration (automatic configuration according to a specified policy), self-optimization (continuous performance monitoring), self-healing (detecting defects and failures, and taking corrective actions), and self-protection (taking preventive measures and defending against malicious attacks).

Several research projects [41] are active in this area.

4. Application Domains

Keywords: *electronic commerce*, *embedded systems*, *multimedia*, *power supply*, *systems administration*, *telecommunications*.

Sardes develops generic tools for distributed applications, in the form of middleware, system kernels, and information servers. These tools are useful in application domains that have one or more of the following properties.

- Cooperation using shared distributed information;
- Mobility of users, information and services;
- Need for dynamic adaptation of infrastructures or applications;
- Use of high performance information servers.

The main actual application domains are:

1. telecommunications : administration of large scale networks, servers and caches for the Web, management of configurable added value services;
2. power supply : administration and monitoring of power supply networked equipment, e.g. uninterrupted power supply units.
3. embedded computing: development of custom made kernels for specific applications (robotics, real time), dynamically reconfigurable kernels;
4. multimedia applications: dynamic adaptation of a videoconferencing system for use by mobile clients;
5. electronic commerce: flexible access to remote services by mobile users, efficient transaction management.

5. Software

5.1. Introduction

Keywords: *J2EE*, *Java thread*, *benchmark*, *cluster*, *mobility*.

Software development is an important aspect of the activity of Sardes. This software serves as a testbed to apply, validate and evaluate the methods and tools developed in the project.

Sardes contributes to the development of the ObjectWeb open source code base (see 8.2), which is accessible at http://www.objectweb.org.

5.2. JavaThread : Support for mobility and persistence of Java threads

Contact : Sara Bouchenak

We have extended the Java Virtual Machine (JVM) with a non-intrusive mechanism for capturing and restoring the complete state of a Java thread. The state may be captured at any execution stage. The capture mechanism does not impose any run time performance penalty. It relies on dynamic type inference techniques for analyzing data on the execution stack, and on dynamic de-optimization of the JIT-compiled Java code. The mechanism has been implemented in the Sun Microsystems JVM.

The *JavaThread* software is distributed in binary form under an Inria license.

See http://sardes.inrialpes.fr/research/JavaThread/

5.3. C-Jdbc: Clustered JDBC-accessed Database

Contact : Emmanuel Cecchet.

JDBC™(short for Java DataBase Connectivity), is a widely used Java API for accessing virtually any kind of tabular data. C-JDBC (Clustered Java™DataBase Connectivity) is an open source database cluster middleware that allows any application to transparently access a cluster of databases through JDBC. The database is distributed and possibly replicated among several nodes and C-JDBC load balances the queries between these nodes. The related research activity is described in 6.2.3.

C-JDBC is an OBJECTWEB project distributed under an LGPL license. C-JDBC has achieved a wide visibility, and has won the Derby Code Contest at ApacheCon US in November 2004.

See http://c-jdbc.objectweb.org

5.4. Benchmarks for J2EE Systems

Contact : Emmanuel Cecchet.

RUBiS (Rice University Bidding System) is an auction site prototype modeled after `eBay.com`. RUBiS is currently used to evaluate design patterns, application servers and communication layers scalability. RUBiS has been implemented for PHP, Servlets and Enterprise Java Beans (EJB) in 7 different versions. Ongoing activities concern JDO (Java Data Objects) and Microsoft .Net versions of RUBiS. RUBBoS is a bulletin board benchmark modeled after an online news forum. Like RUBiS, RUBBoS was designed to evaluate design patterns, application servers and communication layers scalability.

Both RUBiS and RUBBoS are part of the OBJECTWEB JMOB (Java Middleware Open Benchmarking) project and are distributed under an LGPL license.

See http://jmob.objectweb.org

5.5. Jotm: Java Open Transactional Manager

Contact : Emmanuel Cecchet.

JOTM is an open source transaction manager implemented in Java. It was originally part of the JONAS Enterprise Java Beans platform. Since 2002, it has been extracted from the JONAS environment and developed as an autonomous project to provide support for any distributed platform that needs transactional facilities.

JOTM implements the Java Transaction API (JTA) specification with full support on both RMI/JRMP and RMI/IIOP. It also implements the Business Transaction Protocol (BTP) to support transactions for Web Services.

JOTM is an OBJECTWEB project distributed under an LGPL license.

See http://jotm.objectweb.org

6. New Results

6.1. Models and Tools for Distributed Components

The main goal of the component technology theme is to develop a provably sound and efficient reflective software component technology for the construction of highly reconfigurable distributed systems, and in particular distributed software infrastructures (operating systems and middleware). This gives rise to two main research activities: models and foundations, and support for component-based distributed programming. The Fractal component model, jointly developed by Sardes and France Telecom RD, plays a central part in both activities.

6.1.1. Models and foundations

(6.1.3). In this work, the aim is to develop novel models for component-based distributed programming, grounded on a rigourous semantical basis. The approach is to exploit and develop, where appropriate, techniques (e.g. behavioural equivalences, type systems) from process calculi to formally characterize the main primitives in our programming models and to serve as a basis for establishing the soundness of our supporting tools. The current work is aimed at characterizing program equivalence in the Kell calculus, at formalizing the FRACTAL component model, including its component sharing capability, and at defining provably correct abstract machines for the Kell calculus.

6.1.2. Support for component-based distributed programming

(6.1.4). In this work, the aim is to prototype basic support for our component models, building and extending on the FRACTAL component model, and its Java and C implementations. The current work is aimed at efficiently supporting fine-grain dynamic reconfiguration, at introducing activity and concurrency management concepts, and at devising a dynamic Architecture Description Language for FRACTAL.

6.1.3. Models for Distributed Computing

Participants: Jean-Bernard Stefani, Alan Schmitt, Philippe Bidinger.

The work on models has two main objectives: to elicit a model for distributed computing that includes constructs for mobility and dynamic reconfiguration; and to provide a basis for the formal specification of the systems developed in the project. This formal specification in turn let us derive a precise semantic for component-based programs and build verification tools to help the development of such programs.

In 2004, we have continued our work on the Kell calculus, a family of higher-order distributed calculi. More precisely, we have made progress in several directions.

- The design of an instance of the Kell calculus, codenamed FRAKTAL, used to faithfully model the FRACTAL specification [4]. This work participates greatly in the evaluation of the expressiveness of the calculus, and suggests what extensions should be explored.
- An extension of the Kell calculus to allow shared localities (localities with several parents), used to model shared resources. This feature, developed in collaboration with Daniel Hirschkoff's group at ENS Lyon, is needed to capture the notion of shared components from FRACTAL.
- An exploration of the equational theory of the Kell calculus. We have for instance proved that early strong context bisimulation characterizes strong barbed congruence. We are currently exploring late and normal versions of these bisimulations.

Work is in progress to apply the Kell calculus to the specification and proof of actual systems, e.g. the THINK framework for kernel development [7]. This is the subject of the ongoing Ph.D. thesis of Ph. Bidinger. Other work in progress include the implementation of the Kell calculus; and the design of type systems for the Kell calculus and for FRACTAL, inspired by our previous work on the M-calculus [11].

6.1.4. Reconfiguration of Component-based Applications

Run-time reconfiguration is a basic tool for dynamic software adaptation [23]. However, it is difficult to achieve safely and reliably with standard software infrastructures. FRACTAL provides built-in facilities such as dynamic rebinding, which can be used for run-time reconfiguration. A preliminary experience was to extend THINK (a framework for kernel development [7]) with a lightweight component framework derived from FRACTAL, thus allowing dynamic reconfiguration. This was the subject of the Ph. D. thesis of O. Charra [12].

More recent work was done in two directions.

6.1.4.1. 1) A Framework for the Reconfiguration of Fractal Components

Participants: Jean-Bernard Stefani, Vivien Quéma, Jakub Kornaś, Matthieu Leclercq.

The goal of this work is to extend the FRACTAL specification and implementation with dynamic reconfiguration capabilities. In 2004, we have made progress in two main directions.

- Developing, in collaboration with France Telecom R&D, an Architecture Description Language (ADL) for the FRACTAL component model. This extensible ADL currently allows for the deployment and basic reconfiguration of component-based applications. It is evolving towards a more dynamic language that should allow for powerful unanticipated reconfigurations.
- Developing a system support for the reconfiguration of component implementations [29]. The proposed support is based on the usage of the Java class loading mechanism. It has been integrated with FRACTAL ADL.

6.1.4.2. 2) A Framework for Building Self-adaptive Multimedia Applications

Participants: Daniel Hagimont, Oussama Layaida.

The goal of this work is to develop methods and tools for designing adaptive (or, better, self-adaptive) multimedia streaming applications. To that end, such applications must be enhanced with new features allowing them: to observe themselves and their environment, to detect significant changes, and to reconfigure their behavior accordingly. Our previous study on proxy-based multimedia adaptation has demonstrated the effectiveness of the component-oriented approach [31][32]. The next challenge consists in achieving these functions in a generic fashion so that different kinds of multimedia applications as well as their variable reconfiguration requirements can be addressed without additional development efforts.

Towards this goal we designed PLASMA [33], a component-oriented framework for building self-adaptable multimedia applications. PLASMA exploits the recursive composition of FRACTAL with a hierarchy of components (media components, QoS/Resource Probes and Actuators), based on which different kinds of multimedia services can be composed. Likewise, reconfiguration responsibilities are distributed across different levels with a well-established semantics. Combined with the use of a dynamic ADL, we believe that this approach provides an efficient means to attain our objective.

Several application scenarios have been implemented, among which a video streaming application with mobile equipments. The experimental evaluations in this context have demonstrated that reconfiguration can be performed with a very low overhead, while significantly improving the Quality of Service of multimedia applications as well as resource usage on mobile equipments. This work is done in association with Microsoft Research (7.3).

6.2. Distributed System Management

The main objective of our research on distributed system management is to develop methods and tools for building systems that are dependable and that can be managed with minimal human intervention. The intended result of this work is to define an integrated set of frameworks for building such autonomous systems. We believe that the component framework that we are developing (specially the support for ADL-assisted reconfiguration, cf 6.1.4) is a convenient base for such systems, since it provides capabilities for flexible composition, component-level management interfaces, and fault isolation.

The following efforts contribute to the above long term goal:

- developing a framework for distributed observation and monitoring (6.2.1)
- developing a framework for asynchronous event distribution (6.2.2)
- investigating techniques for dynamic resource management, in order to improve availability and performance; we are concentrating on data management on clusters, both at the database level (6.2.3) and at the storage service level (6.2.4),
- building an experimental platform for developing autonomic services on a cluster, integrating most of the above work (6.2.5).

6.2.1. Distributed Observation and Monitoring

Participants: Jean-Bernard Stefani, Emmanuel Cecchet, Renaud Lachaize, Vivien Quéma.

A resource monitoring application needs to have the following properties: distribution, as resources are spread all over the world; scalability, as thousands of devices are involved; flexibility, as new devices are joining and leaving the system dynamically; configurability, as the monitoring application must support a wide spectrum of devices; adaptability, as operating conditions may vary; interoperability, as it is necessary to export monitored data to third-party applications.

We have done a preliminary design of a monitoring application that aims at meeting these challenges [18]. The application is composed of three kinds of distributed cooperating tasks: *Collection* tasks are deployed on monitored devices; they produce events. *Processing* tasks are in charge of processing, i.e. filtering, forwarding or aggregating events sent by collection tasks. Finally, *Presentation* tasks make events sent by processing tasks available to the observers. These tasks are built as components executing asynchronously and communicating using a message-oriented middleware (MOM) built with the framework described in the next section.

This design is embodied in LEWYS, a monitoring infrastructure providing a set of hardware and software probes [26], monitoring filters and a repository to record monitoring information. Distributed observers can subscribe to monitoring events resulting in the dynamic deployment of probes and communication channels. The monitoring repository is a specific observer that records monitoring events for further asynchronous processing. LEWYS is available at http://lewys.objectweb.org

As a conclusion of previous work, the Ph.D. thesis of Ph. Laumay [13] has investigated several aspects of distributed observation (specially causality) and event processing.

6.2.2. Event-based Middleware

Participants: Jean-Bernard Stefani, Matthieu Leclercq, Vivien Quéma.

The results of our research on event-based middleware are embodied in DREAM (Dynamic Reflective Asynchronous Middleware) [34], a framework dedicated to the construction of communication middleware. DREAM provides a set of components (message queues, channels, routers, transformers, multiplexers, etc.) that can be dynamically assembled to build different paradigms of communication: message-passing, event-reaction, publish-subscribe, etc.

In 2004, we have made progress in two main directions.

- Implementing several communication middleware using the DREAM framework. This middleware ranges from JMS [35] to group communication protocols (probabilistic and uniform total order broadcast).
- Specifying and developing a meta-model for the management of activities of DREAM components. This meta-model allows fine-grain control over execution flows crossing components, which is a required feature for building scalable and reconfigurable middleware.

6.2.3. Clustered Databases

Participants: Emmanuel Cecchet, Nicolas Modrzyk.

Clusters of processors are a cost-effective alternative to high performance servers. They were initially introduced for CPU-intensive applications, but their use for data-intensive services is taking an increasing importance.

We have investigated the use of cluster servers to improve the availability and performance of distributed databases. To that effect, we have introduced the concept of RAIDb (Redundant Array of Inexpensive Databases), the counterpart of RAID (Redundant Array of Inexpensive Disks) for databases. RAIDb aims at low cost improvement of performance and fault tolerance over a single database, by combining multiple database instances into an array of databases. Based on this concept, we have implemented C-JDBC (Clustered Java™DataBase Connectivity) [27][39], an open source database cluster middleware (5.3) that allows any application to transparently access a cluster of databases through the JDBC standard interface. C-JDBC implements database partitioning, full mirroring and partial mirroring. Using standard benchmarks such as TPC-W, we have shown that partial replication can bring substantial scalability improvement (up to 25%) compared to full replication. However, static round robin policies are inefficient and partial replication needs dynamic load balancing algorithms to scale. We are also investigating the performance of SQL query caching at the middleware level.

6.2.4. System Support for Cluster File Systems

Participants: Jacques Mossière, Renaud Lachaize.

The aim of the Proboscis[2] project is to study the sharing of storage devices in clusters, where the storage devices are distributed across the nodes and accessed with efficient networking technologies. For this purpose, we are developing a modular and extensible software infrastructure for remote disk access construction and administration. This framework is intended as a set of building blocks, on which specific storage services such as cluster file systems can be built and deployed. A key feature of Proboscis lies in its use of explicit I/O data paths to facilitate disk access scheduling, reconfiguration at runtime and monitoring.

Recent work on Proboscis [30] has focused on dynamic reconfiguration of the data paths to increase the flexibility and availability of clustered data servers. The modularity of the framework is used to allow non-disruptive changes of the low level storage services without any impact on the upper layers of the system (e.g. logical volume managers, cluster file systems or databases). This is beneficial for fault tolerance (data redistribution in case of a server crash), for performance (load balancing and caching policies), and for occasional administration needs (e.g. code updates or transparent migration to new storage or networking hardware). We are currently leveraging the adaptable and autonomic features of the framework for realistic use cases such as distributed swapping and cooperative caching.

This work is the subject of the Ph.D. thesis of R. Lachaize.

The source code of Proboscis is available on request. See http://sardes.imag.fr/~rlachaiz/proboscis.shtml.

6.2.5. A Framework for Component-based Autonomic Computing

Participants: Jean-Bernard Stefani, Daniel Hagimont, Emmanuel Cecchet, Sara Bouchenak, Fabienne Boyer, Noël De Palma, Sacha Krakowiak.

In 2004, we have started the design of a platform called Jade, which we plan to use as a testbed for research on autonomic computing, integrating most of the work described in the previous sections. The infrastructure of Jade is a LAN-based cluster of Unix nodes. The application domain is interactive, data-driven computing based on dynamic web content, an area that is representative of an important segment of distributed computing, for which availability, performance, and ease of run-time administration are crucial issues. We have selected J2EE as the initial middleware environment, since we have acquired experience in the design, use and evaluation of various components of this environment, specially in the context of ObjectWeb (8.2).

The problem that we intend to address is *automated recovery management in the face of failures* (physical node failures in a first step; software failures in a later phase). The main thesis that we defend is that architecture-based management is an adequate base for autonomic computing. Our management system is based on the following design principles:

- Both the platform and the application proper are organized as a set of components, using the FRACTAL model, which provides explicit management interfaces allowing, among others, dynamic rebinding and reconfiguration.
- Management services rely on an explicit, causally connected description of the running system; these services are themselves supervised in the same way as the managed J2EE system. To this end, we exploit our work on architectural description of component-based systems (6.1.4). The internal representation is used by the control loops that detect and analyze abnormal conditions, and attempt to correct them by means of actuators performing recovery tasks.

[2] Proboscis is a joint research effort of Sardes and the DistLab group at DIKU, the Computer Science Laboratory of the University of Copenhagen.

At the end of 2004, a initial version of the prototype, which includes deployment and initial configuration, has been implemented. Based on this first experience, a redesign is underway. Preliminary ideas for the design of Jade may be found in [22].

7. Contracts and Grants with Industry

7.1. Collaboration with Bull

Participants: Jacques Mossière, Takoua Abdellatif, Simon Nieuviarts.

The theme of the collaboration with Bull is the development of system software for exploiting clusters operating under Linux for scientific and data management applications. This collaboration started in October 2000, and also involves the Inria project-team Apache (Jacques Briat, Yves Denneulin).

Our contribution is twofold:

- Using a clustered server for developing an efficient, fault-tolerant version of an *Enterprise Java Beans* (EJB) platform. This work has been completed and is the subject of the forthcoming Ph.D. thesis of S. Nieuviarts (Bull co-funding).
- Developing an administration support system for a cluster. This work started in September 2003 and is the subject of the Ph.D. thesis of T. Abdellatif (Bull co-funding).

7.2. Collaboration with France-Telecom R&D

Participants: Jean-Bernard Stefani, Vivien Quéma, Jakub Kornaś.

Sardes maintains an active collaboration with France-Telecom R&D (Norbert Segard Center, Distributed Systems Architecture group):

1. Collaboration on the following aspects: extensions to the FRACTAL component model [4] (J.-B. Stefani); framework for dynamic reconfiguration of FRACTAL based applications; framework for adaptable program composition ([14], Ph. D. thesis of R. Lenglet (FT R&D), supervised by D. Hagimont). Most of this work takes place within the OBJECTWEB consortium (8.2),
2. Collaboration on distributed process calculi including mobility and distribution (6.1.3), and on the associated virtual machines (J.-B. Stefani). This work is done within the European IST project Mikado (8.3.2).

7.3. Collaboration with Microsoft

Participants: Emmanuel Cecchet, Daniel Hagimont, Oussama Layaida, Jacques Mossière.

The goal of this contract is to develop techniques for dynamic adaptation of embedded multimedia applications on Microsoft Windows CE .Net. This work is jointly carried out by project-teams Sardes and WAM at Inria Rhône-Alpes (N. Layaida).

We are extending the adaptation techniques developed in Sardes (6.1.4) for dynamic proxy configuration in order to ensure QoS of multimedia applications using the Microsoft DirectShow middleware framework. We intend to demonstrate that our techniques can be applied in a resource-constrained environment.

7.4. Collaboration with ST-MicroElectronics

Participants: Jacques Mossière, Ali Erdem Özcan.

The goal of this project, started in November 2003, is to investigate the use of the THINK framework to develop operating systems for on-chip multiprocessors. This is the subject of the Ph.D. thesis of A. E. Özcan (ST-MicroElectronics co-funding).

7.5. RNTL Arcad Project

Participants: Jean-Bernard Stefani, Fabienne Boyer, Daniel Hagimont, Noël De Palma.

Arcad is an exploratory project funded by the French Ministry of Industry under the RNTL program. Its partners are: Inria (Sardes and Oasis project-teams), France Telecom R&D, École des Mines de Nantes and the I3S laboratory at Nice Sophia-Antipolis (Rainbow group, M. Riveill). The project aimed at designing and developing distributed extensible environments for deploying distributed component-based applications. Sardes contributed work on dynamic reconfiguration of applications, based on Fractal.

The project started in November 2000 and ended in June 2004.

See http://arcad.essi.fr/

7.6. RNTL Inside Project

Participants: Jean-Bernard Stefani, Matthieu Leclercq, Vivien Quéma.

Inside is a pre-competitive project funded by the French Ministry of Industry under the RNTL program. Its partners are Inria (Sardes project-team), and three companies: Schneider Electric, OpenSugar and ScalAgent. The goal of the project is to develop a software infrastructure to support Internet-based services (specially distributed equipment monitoring) in the area of power distribution. The infrastructure will be validated in a real-life environment. The contribution of Sardes is the work on event-based middleware, including the Dream prototype.

See http://www.telecom.gouv.fr/rntl/projet/Posters-PDF/RNTL-Poster-INSIDE.pdf

8. Other Grants and Activities

8.1. National Actions

8.1.1. ARP Network

Sardes is a member of the "Distributed Systems and Applications" group of the national research network on Systems Architecture, Networks and Parallelism (ARP: Architecture, Réseaux et Parallélisme).

See http://www.arp.cnrs.fr.

8.2. ObjectWeb Consortium

Participants: Jean-Bernard Stefani, Emmanuel Cecchet, Sacha Krakowiak.

OBJECTWEB is an open-source software community created at the end of 1999 by France Telecom R&D, Bull and Inria. Its goal is the development of open-source distributed middleware, in the form of flexible and adaptable components. These components range from specific software frameworks and protocols to integrated platforms. OBJECTWEB developments follow a systematic component-based approach.

In 2002, OBJECTWEB evolved into an international consortium hosted by Inria. The consortium is an independent non-profit organization open to companies, institutions and individuals.

Sardes contributes to OBJECTWEB through its technical involvement in the development of software components and frameworks (e.g. FRACTAL, C-JDBC, JOTM, DREAM, TRIBE) and through participation in the management structures of the consortium: board (J.-B. Stefani, past president), and college of architects (E. Cecchet, president, J.-B. Stefani and S. Krakowiak, members).

See http://www.objectweb.org

8.3. Projects Funded by the European Commission

8.3.1. IST Project Ozone

Participants: Daniel Hagimont, Slim Ben Atallah, Oussama Layaida.

The goal of the IST project Ozone is to develop a software environment for ubiquitous computing and ambient intelligence. The project's partners are: Philips (Netherlands), IMEC (Belgium), LEP (France), Epictoid (Netherlands), Eindhoven University (Netherlands), Thomson Multimédia and Inria (project-teams Arles, Moscova and Sardes).

One of the tasks of the project is to develop software frameworks for embedded real-time multiprocessor systems. Our contribution to this task is two-fold:

- we study the application of our results on the management of QoS for multimedia applications (6.1.4) in the context of the Ozone project.
- we study the application of our tools for component replication in the context of the Ozone project, more precisely in the context of applications based on Web Services.

See http://www.extra.research.philips.com/euprojects/ozone/

8.3.2. IST Project Mikado

Participants: Jean-Bernard Stefani, Philippe Bidinger.

The goal of the IST project Mikado is to develop a model for distributed mobile programming based on a process calculus. Investigation topics include type models, programming language and virtual machines technologies, and specification languages. The project's partners are: Inria (project-teams Mimosa and Sardes), France Telecom R&D (DTL/ASR Laboratory), University of Sussex, UK (M. Hennessy), Università di Firenze, Italy (R. de Nicola), University of Lisbon, Portugal (V. Vasconcelos). The main contribution of Sardes is in the areas of models for component-based systems and distributed process calculi.

See http://mikado.di.fc.ul.pt/

8.3.3. IST Project Gorda

Participants: Jean-Bernard Stefani, Emmanuel Cecchet.

GORDA (Open Replication of Databases) is an IST STREPS project. The GORDA project aims at (i) promoting the interoperability of databases and replication protocols by defining generic architecture and interfaces that can be standardized; (ii) providing general purpose and widely-applicable database systems; and (iii) providing uniform techniques and tools for managing secure and heterogeneous replicated database systems. The project partners are Universidade do Minho (R. Oliveira), Università della Svizzera Italiana (F. Pedone), Universidade de Lisboa (L. Rodrigues), INRIA (Sardes project), Emic Networks (Finland) and MySQL AB (Sweden). The main contribution of Sardes is the C-JDBC technology, and the development of frameworks and tools for its use.

See http://gorda.di.uminho.pt

8.3.4. ITEA Project Osmose

Participants: Emmanuel Cecchet, Jean-Bernard Stefani.

Osmose (Open Source Middleware for Open Systems in Europe) is a project funded by the ITEA program. The project is focused on the development, enhancement, and validation in defined test-beds of a comprehensive adaptable Open Source middleware to be hosted by the ObjectWeb consortium (8.2). The testbeds will be developed in three areas: telecom services, home gateway, and avionics.

The Consortium is built around three sets of partners: 6 large industrial companies (Bull, France Telecom, Philips, Telefonica, Telvent and Thales), 6 SMEs (Bantry Technologies, iTEL, Kelua, Lynx, VICORE and Whitestein Technologies), and 7 academic partners (Charles University, EPFL, Inria, INT, LIFL, Imag-LSR and Universidad Politécnica de Madrid). The contribution of Sardes is in the area of frameworks and tools for component-based middleware.

See http://www.itea-osmose.org/

8.4. International Networks and Working Groups

8.4.1. CoreGrid Network of Excellence (IST-004265)

Sardes is a member of CoreGrid, the European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies, launched in 2004 for four years. Its objective is to coordinate European efforts in the area of Grid and Peer-to-Peer technologies. It gathers forty-two institutions. Sardes is contributing in the areas of programming models and software architecture.

See http://www.coregrid.net/

8.4.2. Artist Network of Excellence (IST-2001-34820)

Sardes is a member of the Network of Excellence *Advanced Real Time Systems* (Artist), whose objective is to coordinate European efforts in advanced real time systems. Sardes is active in the *Adaptive Real-Time Systems For QoS Management* action.

See http://www.systemes-critiques.org/ARTIST/

8.5. International Bilateral Collaborations

8.5.1. Europe

Sardes maintains long term collaboration with several research groups in Europe:

- University of Newcastle upon Tyne, *Distributed Systems Group* (Prof. Santosh Shrivastava). Collaboration on middleware and distributed programming.
- École Polytechnique Fédérale de Lausanne: Distributed Systems Laboratory (Prof. André Schiper), Distributed Programming Laboratory (Prof. Rachid Guerraoui), LabOS (Prof. Willy Zwaenepoel). Collaboration on fault tolerance and performance management for clustered applications.
- University of Lancaster, *Distributed Media Systems* (Prof. Gordon Blair), on adaptable middleware for multimedia communication.
- Trinity College, Dublin, *Distributed Systems Group* (Prof. Vinny Cahill), on distributed programming and clusters. A. Senart, a former Sardes Ph.D. student, has been staying with TCD for the academic year 2003-2004.
- University of Copenhagen, DIKU Laboratory *Distributed Systems Group* (Prof. Eric Jul), on system support for clustered servers (visits and shared software).

8.5.2. North Africa

Sardes maintains contacts with École Nationale des Sciences de l'Informatique (ENSI), Université de la Manouba, Tunis, Tunisia. Slim Ben Atallah, associate professor at ENSI, has been staying with Sardes from July 2002 to June 2004 as an invited scientist. Sardes also welcomes ENSI students as interns for last term projects.

9. Dissemination

9.1. Community Service

J.-B. Stefani, E. Cecchet, D. Hagimont and N. De Palma are involved in the organization of the 6*th ACM/IFIP/Usenix International Middleware Conference*, which will be held at Grenoble in November 2005.

V. Quéma and S. Jean are members of the organization and program committees of DECOR'04, the first French Conference on Software Deployment and (Re)configuration, held at Grenoble in October 2004. DECOR'04 is organized by the Sardes and Adèle groups of the IMAG LSR Laboratory.

J.-B. Stefani is a member of the editorial board of the journal *Annales des Télécommunications*, and of the program committees of the *Middleware'04* Conference, the DOA 2004 Conference (*Distributed Objects and Applications*), the *OPODIS 04* Conference and of DECOR'04.

E. Cecchet has co-organized the *Workshop on Component And Middleware Performance* at the OOPSLA 2004 Conference (http://nenya.ms.mff.cuni.cz/projects/corba/oopsla-workshop-04/). He is a member of the program committees of COSET-1: *First International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters*, in conjunction with ICS'04, of the *Workshop on Dependable Distributed Data Management* in conjunction with SRDS 2004, and of the *2nd Workshop on Middleware for Grid Computing* in conjunction with Middleware 2004.

D. Hagimont is the chairman of the organization committee of the *Middleware 2005* Conference, and a member of the scientific committee of *Journées Composants 2004*, organized by the French Chapter of ACM SIGOPS.

A. Schmitt is a member of the program committee of DBPL05 (*10th International Symposium on Data Base Programming Languages*). He was invited speaker at FOOL04 (*Eleventh International Workshop on Foundations of Object-Oriented Languages*). He is also in charge of a joint Imag-Inria seminar on “Distributed Systems and Data Management”.

S. Krakowiak is a member of the steering committee of ERSADS (*European Research Seminar on Advances in Distributed Systems*), the school/workshop initiated in 1995 by the CaberNet Network of Excellence. He is also the chairman of the committee sponsored by *Specif* (the French Computer Science Researchers' and Teachers' Association) to select the best Ph.D. thesis in Computer Science defended each year in France.

9.2. University Teaching

S. Bouchenak, F. Boyer, N. De Palma, S. Krakowiak, J. Mossière and A. Schmitt have taught several operating systems and distributed systems courses at the M.S. and M.Eng. levels, both at Institut National Polytechnique de Grenoble and at université Joseph Fourier. Most of our Ph.D. students contributed to these courses as teaching assistants.

9.3. Participation in Seminars, Workshops, Conferences

Several members of Sardes attended various scientific conferences and workshops. See the relevant section of the Bibliography for details.

9.4. Miscellaneous

This section mentions publications involving members of Sardes, but not directly related to the project. See [24] and [28].

10. Bibliography

Major publications by the team in recent years

[1] S. KRAKOWIAK, S. SHRIVASTAVA (editors). *Recent Advances in Distributed Systems*, Lecture Notes in Computer Science 1752, Springer, 2000.

[2] G. BLAIR, J.-B. STEFANI. *Open Distributed Processing and Multimedia*, Addison-Wesley, 1997.

[3] S. BOUCHENAK. *Making Java Applications Mobile or Persistent*, in "6th USENIX Conference on Object-Oriented Technologies and Systems, San Antonio, Texas", January 2001.

[4] É. BRUNETON, T. COUPAYE, J.-B. STEFANI. *Recursive and Dynamic Software Composition with Sharing*, in "Proceedings of the 7th ECOOP International Workshop on Component-Oriented Programming (WCOP'02), Malaga (Spain)", June 10th-14th 2002.

[5] É. BRUNETON, M. RIVEILL. *An Architecture for Extensible Middleware Platforms*, in "Software – Practice and Experience", vol. 31, November 2001, p. 1237–1264.

[6] E. CECCHET, J. MARGUERITE, W. ZWAENEPOEL. *Performance and Scalability of EJB Applications*, in "Proceedings of OOPSLA'02, Seattle, WA, USA", November 2002.

[7] J.-PH. FASSINO, J.-B. STEFANI, J. LAWALL, G. MULLER. *THINK: A Software Framework for Component-based Operating System Kernels*, in "Proceedings of Usenix Annual Technical Conference, Monterey (USA)", June 10th-15th 2002.

[8] D. HAGIMONT, F. BOYER. *A Configurable RMI Mechanism for Sharing Distributed Java Objects*, in "IEEE Internet Computing", vol. 5, n° 1, 2001, p. 36–44.

[9] PH. LAUMAY, É. BRUNETON, N. DE PALMA, S. KRAKOWIAK. *Preserving Causality in a Scalable Message-Oriented Middleware*, in "Middleware 2001, IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg", nov 2001.

[10] É. NAJM, A. NIMOUR, J.-B. STEFANI. *Behavioural Typing for Objects and Process Calculi*, chap. 13, Cambridge University Press, 2001, p. 281–301.

[11] A. SCHMITT, J.-B. STEFANI. *The M-calculus: A Higher-Order Distributed Process Calculus*, in "In Proceedings of the 30th Annual ACM Symposium on Principles of Programming Languages (POPL'03), New Orleans, LA, USA", January 15-17 2003.

Doctoral dissertations and Habilitation theses

[12] O. CHARRA. *Conception de noyaux de systèmes embarqués reconfigurables*, Ph. D. Thesis, université Joseph Fourier, May 2004.

[13] PH. LAUMAY. *Configuration et déploiement d'intergiciel asynchrone sur système hétérogène à grande échelle*, Ph. D. Thesis, Institut National Polytechnique de Grenoble, March 2004.

[14] R. LENGLET. *Composition flexible et adaptable de transformations de programmes*, joint appointment with France Telecom R&D and Sardes, Ph. D. Thesis, Institut National Polytechnique de Grenoble, November 2004.

Articles in referred journals and book chapters

[15] S. BOUCHENAK, D. HAGIMONT, S. KRAKOWIAK, N. DE PALMA, F. BOYER. *Experiences Implementing Efficient Java Thread Serialization, Mobility and Persistence*, in "Software – Practice and Experience", vol. 34, n° 4, 2004, p. 355–394.

[16] O. LAYAÏDA, S. B. ATALLAH, D. HAGIMONT. *A Framework for Dynamically Configurable and Reconfig-*

urable Network-based Media Adaptations, in "In Journal Of Internet Technology, Special Issue on Real Time Adaptive Media Delivery over the Internet", vol. 5, n^o 4, 2004.

[17] V. QUÉMA, R. BALTER, L. BELLISSARD, D. FÉLIOT, A. FREYSSINET, S. LACOURTE. *ScalAgent : une plate-forme à composants pour applications asynchrones*, in "Technique et Science Informatiques", vol. 23, n^o 2, april 2004.

[18] V. QUÉMA, R. LACHAIZE, E. CECCHET. *An Asynchronous Middleware for Grid Resource Monitoring*, in "Concurrency and Computation: Practice and Experience", vol. 16, n^o 5, april 2004.

Publications in Conferences and Workshops

[19] T. ABDELLATIF, E. CECCHET, R. LACHAIZE. *Evaluation of a Group Communication Middleware for Clustered J2EE Application Servers*, in "International Symposium on Distributed Objects and Applications (DOA 2004), Larnaca, Cyprus", October 2004.

[20] T. ABDELLATIF, E. CECCHET, R. LACHAIZE, J. MOSSIÈRE. *Évaluation d'un intergiciel de communication de groupes pour des serveurs d'applications J2EE sur grappe*, in "Les Nouvelles Technologies de la Répartition (NOTERE 2004), Saida, Morocco", June 2004.

[21] S. BEN ATALLAH, D. HAGIMONT, S. JEAN, N. DE PALMA. *A first step toward autonomous clustered J2EE applications management*, in "1st International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters (COSET-1), Saint-Malo, France", June 2004.

[22] S. BOUCHENAK, F. BOYER, E. CECCHET, S. JEAN, A. SCHMITT, J.-B. STEFANI. *A component-based approach to distributed system management - A use case with self-manageable J2EE clusters*, in "11th ACM SIGOPS European Workshop, Leuven, Belgium", September 2004, http://www.cs.kuleuven.ac.be/conference/SIGOPSEW04/.

[23] E. BRUNETON, T. COUPAYE, M. LECLERCQ, V. QUÉMA, J.-B. STEFANI. *An Open Component Model and its Support in Java*, in "Proceedings of the International Symposium on Component-based Software Engineering (CBSE'2003), Edinburgh, Scotland", May 2004.

[24] C. BÉTOURNÉ, J. FERRIÉ, C. KAISER, S. KRAKOWIAK, J. MOSSIÈRE. *Ésope : une étape dans la recherche française en systèmes d'exploitation (1968–1972)*, in "Actes du 7ème Colloque sur l'Histoire de l'Informatique et des Télécommunications (CHIR'04), Rennes, France", November 2004, p. 173–198.

[25] E. CECCHET. *RAIDb: Redundant Array of Inexpensive Databases*, in "IEEE/ACM International Symposium on Parallel and Distributed Applications (ISPA'04), Hong Kong, China", December 2004.

[26] E. CECCHET, H. ELMELEEGY, O. LAYAÏDA, V. QUÉMA. *Implementing Probes for J2EE Cluster Monitoring*, in "Proceedings of OOPSLA 2004 Workshop on Component and Middleware Performance, Vancouver, British Columbia, Canada", October 2004.

[27] E. CECCHET, J. MARGUERITE, W. ZWAENEPOEL. *C-JDBC: Flexible Database Clustering Middleware*, in "Proc. USENIX Annual Technical Conference, Freenix Track, Boston, MA, USA", June 2004.

[28] J. N. FOSTER, M. B. GREENWALD, J. T. MOORE, B. C. PIERCE, A. SCHMITT. *Combinators for Bi-Directional Tree Transformations: A Linguistic Approach to the View Update Problem*, in "ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Long Beach, California", to appear.

[29] J. KORNAŚ, M. LECLERCQ, V. QUÉMA, J.-B. STEFANI. *Support pour la reconfiguration d'implantation dans les applications à composants Java*, in "Actes de la 1ère Conférence Française sur le Déploiement et la (Re)Configuration de Logiciels (DECOR'2004), Grenoble, France", October 2004.

[30] R. LACHAIZE, J. S. HANSEN. *Simplifying Administration Through Dynamic Reconfiguration in a Cooperative Cluster Storage System*, in "Proceedings of the 6th IEEE International Conference on Cluster Computing (Cluster 2004), San Diego, CA, USA", IEEE Computer Society, September 2004.

[31] O. LAYAÏDA, S. BEN ATALLAH, D. HAGIMONT. *Adaptive Media Streaming Using Self-reconfigurable Proxies*, in "Proc. 7th IEEE International Conference on High Speed Networks and Multimedia Comuunications (HSNMC'04), Toulouse, France", June 30–July 2, 2004.

[32] O. LAYAÏDA, S. BEN ATALLAH, D. HAGIMONT. *Reconfiguration-based QoS Management in Multimedia Streaming Applications*, in "Pro.30th IEEE/Euromicro Conference, Track on Multimedia and Communications: Challenges in Distributed Multimedia Systems, Rennes, France", August 31–September 3, 2004.

[33] O. LAYAÏDA, D. HAGIMONT. *PLASMA : A Component-based Framework for Building Self-Adaptive Applications*, in "Proc. SPIE/IS&T Symposium On Electronic Imaging, Conference on Embedded Multimedia Processing and Communications, San Jose, CA, USA", to appear, January .

[34] M. LECLERCQ, V. QUÉMA, J.-B. STEFANI. *DREAM: a Component Framework for the Construction of Resource-Aware, Reconfigurable MOMs*, in "Proceedings of the 3rd Workshop on Reflective and Adaptive Middleware (RM'2004), Toronto, Canada", October 2004.

[35] V. QUÉMA, R. BALTER, L. BELLISSARD, D. FÉLIOT, A. FREYSSINET, S. LACOURTE. *Asynchronous, Hierarchical and Scalable Deployment of Component-Based Applications*, in "Proceedings of the 2nd International Working Conference on Component Deployment (CD'2004), Edinburgh, Scotland", May 2004.

[36] A. E. ÖZCAN, S. JEAN. *Technologies de systèmes d'exploitation pour environnements répartis appliquées aux systèmes multiprocesseurs sur puce*, in "Actes du congrès MAJECSTIC'04 (Manifestation de Jeunes Chercheurs en STIC), Calais, France", 2004.

Internal Reports

[37] T. ABDELLATIF, E. CECCHET, R. LACHAIZE. *JGroups evaluation in J2EE cluster environments*, Technical report, n° RR-5336, Inria, August 2004, http://www.inria.fr/rrrt/rr-5336.html.

Miscellaneous

[38] B. BOUTALEB JOUTEI. *Cartographie automatique de systèmes distribués pour l'administration de grappes de machines*, Rapport de Master Recherche Informatique : Systèmes et Logiciel (Master's thesis), École Doctorale Mathématiques et Informatique, Grenoble, June 2004.

[39] E. CECCHET. *C-JDBC: A Middleware Framework for Database Clustering*, June 2004, IEEE Data Engineering Bulletin.

[40] J. REBOUL. *Surveillance d'applications sur grappes de machines*, Rapport de Master Recherche Informatique : Systèmes et Logiciel (Master's thesis), École Doctorale Mathématiques et Informatique, Grenoble, June 2004.

Bibliography in notes

[41] *Research Projects in Autonomic Computing*, 2003, http://www.research.ibm.com/autonomic/research/projects.html, IBM Research.

[42] *Report of the IST Advisory Group concerning Software Techologies, Embedded Systems, and Distributed Systems: A European Strategy towards an Ambient Intelligent environment*, June 2002, http://www.cordis.lu/ist/istag.htm, Office for Official Publications of the European Communities.

[43] M. K. AGUILERA, R. E. STROM, D. C. STURMAN, M. ASTLEY, T. D. CHANDRA. *Matching Events in a Content-Based Subscription System*, in "Symposium on Principles of Distributed Computing (PODC)", 1999, p. 53-61, http://citeseer.nj.nec.com/aguilera99matching.html.

[44] T. E. ANDERSON, B. N. BERSHAD, E. D. LAZOWSKA, H. M. LEVY. *Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism*, in "ACM Transactions on Computer Systems", vol. 10, n° 1, February 1992, p. 53–79.

[45] G. BLAIR, G. COULSON, A. ANDERSEN, L. BLAIR, M. CLARKE, F. COSTA, H. DURAN-LIMON, T. FITZPATRICK, L. JOHNSTON, R. MOREIRA, N. PARLAVANTZAS, K. SAIKOSKI. *The Design and Implementation of OpenORB v2*, in "IEEE Distributed Systems Online, vol. 2 no 6, Special Issue on Reflective Middleware", 2001.

[46] G. BOUDOL, F. GERMAIN, M. LACOSTE. *Analyse des modèles de programmation, plates-formes et langages – Tome I : Document d'analyse des langages et modèles de la mobilité*, 73 pages, Technical report, n° RR-3930, INRIA Sophia Antipolis, April 2000, http://www.inria.fr/rrrt/rr-3930.html.

[47] F. BUSCHMANN, R. MEUNIER, H. ROHNERT, P. SOMMERLAD, M. STAL. *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*, 467 pp., John Wiley and Sons, 1995.

[48] A. CARZANIGA, D. S. ROSENBLUM, A. L. WOLF. *Design and evaluation of a wide-area event notification service*, in "ACM Transactions on Computer Systems (TOCS)", vol. 19, n° 3, 2001, p. 332–383.

[49] D. ENGLER, M. KAASHOEK, J. O'TOOLE. *Exokernel: An Operating System Architecture for Application-Level Resource Management*, in "Proceedings of the 1995 ACM Symposium on Operating Systems Principles, Copper Mountain Resort, CO, USA", December 1995, p. 251–266.

[50] P. TH. EUGSTER, P. FELBER, R. GUERRAOUI, A.-M. KERMARREC. *The Many Faces of Publish/Subscribe*, in "ACM Computing Surveys", vol. 35, n° 2, June 2003, p. 114–131.

[51] E. GAMMA, R. HELM, R. JOHNSON, J. VLISSIDES. *Design Patterns: Elements of Reusable Object Oriented Software*, 416 pp., Addison-Wesley, 1994.

[52] A. G. GANEK, T. A. CORBI. *The dawning of the autonomic computing era*, in "IBM Systems Journal", Special issue on autonomic computing, vol. 42, n^{o} 1, 2003, p. 5–17.

[53] R. E. JOHNSON. *Frameworks=(Components+Patterns): How frameworks compare to other object-oriented reuse techniques*, in "Communications of the ACM", vol. 40, n^{o} 10, October 1997, p. 39–42.

[54] J. O. KEPHART, D. M. CHESS. *The Vision of Autonomic Computing*, in "Computer - IEEE Computer Magazine", vol. 36, n^{o} 1, 2003, p. 41–50.

[55] G. KICZALES, J. DES RIVIÈRES, D. BOBROW. *The Art of the Metaobject Protocol*, MIT Press, 1991.

[56] G. KICZALES. *Aspect-Oriented Programming*, in "ACM Computing Surveys", vol. 28, n^{o} 4, 1996, 154.

[57] G. LEAVENS, M. S. (EDS). *Foundations of Component-Based Systems*, Cambridge University Press, 2000.

[58] G. MULLER, R. MARLET, E. VOLANSCHI, C. CONSEL, C. PU, A. GOEL. *Fast, Optimized Sun RPC Using Automatic Program Specialization*, in "Proceedings of the 18th International Conference on Distributed Computing Systems, Amsterdam, The Netherlands", IEEE Computer Society Press, May 1998, p. 240–249.

[59] I. PIUMARTA, F. RICCARDI. *Optimizing direct threaded code by selective inlining*, in "1998 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'98), Montreal, Canada", 1998.

[60] D. C. SCHMIDT, M. STAL, H. ROHNERT, F. BUSCHMANN. *Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects*, 666 pp., John Wiley and Sons, 2000.

[61] C. SZYPERSKI. *Component Software - Beyond Object-Oriented Programming*, 2nd ed., 589 pp., Addison-Wesley, 2002.

[62] R. VAN RENESSE, K. P. BIRMAN, W. VOGELS. *Astrolabe: a Robust and Scalable Technology for Distributed Systems Monitoring, Management, and Data Mining* , in "ACM Transactions on Computer Systems", vol. 21, n^{o} 2, May 2003, p. 164–206.