Project no. IST-033576

XtreemOS

Integrated Project
BUILDING AND PROMOTING A LINUX-BASED OPERATING SYSTEM TO SUPPORT VIRTUAL ORGANIZATIONS FOR NEXT GENERATION GRIDS

Final Publishable Report

Actual submission date: May 18^{th}, 2011

Start date of project: June 1^{st} 2006

Type: Deliverable
WP number: WP1.1

Responsible institution: INRIA
Editor & and editor's address: Christine Morin
INRIA
Campus universitaire de Beaulieu
35042 Rennes cedex
France

Version 1.1 / Last edited by Christine Morin / May 18, 2011

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
PU	Public	√
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision history:

Version	Date	Authors	Institution	Section affected, comments
0.1	01/03/11	Christine Morin	INRIA	Prepared the structure for the report.
0.2	08/03/11	Marjan Šterk	XLAB	Added the demonstrators section.
0.3	29/04/11	Christine Morin	INRIA	document polishing
1.0	05/05/11	Christine Morin	INRIA	final version of the report.
1.1	18/05/11	Christine Morin	INRIA	final version of the report taking into account comments received from the EC.

Executive Summary

This document comprises the final report on the IST Integrated Project XtreemOS - "Building and promoting a Linux-based operating systems to support virtual organizations for next generation Grids". The project started in June 2006 and ended in September 2010.

The XtreemOS operating system provides for Grids what a traditional operating system offers for a single computer: abstraction from the hardware and secure resource sharing between different users. It thus simplifies the work of users belonging to virtual organizations by giving them the illusion of using a traditional computer while removing the burden of complex resource management issues of a typical Grid environment.

We have developed a comprehensive set of cooperating system services. XtreemOS software components range from Linux kernel modules to application-support libraries. The XtreemOS operating system provides three major distributed services to users: application execution management (providing scalable resource discovery and job scheduling for distributed interactive applications), data management (accessing and storing data in XtreemFS, a POSIX-like file system spanning the Grid) and virtual organization management (building and operating dynamic virtual organizations).

Three flavours of the system have been implemented for individual PC, clusters and mobile devices (PDA, smartphone, notebook).

The XtreemOS software has been experimented and validated with a wide range of applications. Various demonstrators were implemented, shown at different events and published on the web.

The project results are available as open source software. The consortium member organizations plan to exploit some of the results in follow-up research projects and in future products.

Contents

1 The Challenge 5

2 Addressing the challenge: the XtreemOS Position 6
- 2.1 Fundamental Concepts . . . 6
  - 2.1.1 Transparency . . . 6
  - 2.1.2 Scalability . . . 7

3 Who can benefit from XtreemOS 9
- 3.1 Advantages for all users . . . 10
- 3.2 Advantages for end users . . . 10
- 3.3 Advantages for administrators . . . 11
- 3.4 Advantages for application programmers . . . 12

4 Highlights and Achievements 12
- 4.1 XtreemOS software . . . 12
- 4.2 Applications and Use Cases . . . 12
- 4.3 Evaluation . . . 13
- 4.4 XtreemOS Open Permanent Testbed . . . 14
- 4.5 XtreemOS Sustainability . . . 15

5 The Results 15
- 5.1 VO Management and Security . . . 15
- 5.2 Resource Discovery and Selection . . . 16
- 5.3 Application Execution Management (AEM service) . . . 16
- 5.4 XtreemFS Grid file system . . . 17
- 5.5 Volatile Data Management . . . 17
- 5.6 Distributed Servers & Virtual nodes . . . 17
- 5.7 XtreemOS API . . . 18
- 5.8 XtreemOS Cluster Flavour . . . 18
- 5.9 XtreemOS Mobile Device Flavour . . . 18

6 The Demonstrators 19

7 Availability of Results 20

8 Potential Impact of Results 21

9 Lessons Learned during the Project 22
- 9.1 High quality software development is challenging . . . 22
- 9.2 Attracting external users is another huge challenge . . . 22
- 9.3 Other lessons learned . . . 23

10 Partners 24

11 Contacts 24

1 The Challenge

For many businesses, the ability to dynamically adapt to changing environments becomes the key success component for the overall business. Therefore, many business companies need to significantly increase their agility and cost efficiency. This highly affects the underlying information and communication infrastructure. Hence, businesses are searching for new technologies that overcome current limitations and allow them to execute their businesses in an effective manner by additionally providing a high degree of adaptability. The mentioned limitations result from the different kinds of business applications as well as from security related issues. Many business applications consist of several executables that need to be started, managed and stopped in a coordinated fashion. Furthermore, the majority of business applications uses databases in the range of many Gigabytes for storing business related data. Additionally, a larger proportion of such applications is interactive and thus has higher requirements for the direct customer response times.

Many enterprises are operating in a distributed fashion. Thus, the whole company is divided into several administrative domains. In order to run the overall business effectively, the different locations must cooperate and dynamically adapt as a whole during changes. One of the main goals during this operation is the minimization of administration tasks. Thus, complex installations have to be automated as well as the dynamic adjustment of systems. Furthermore, it is essential for enterprise to be able to execute legacy software within these environments without the need to modify or recompile the various system components.

The term Grid computing was introduced at the end of 90s by Foster and Kesselman; it was envisioned as *"an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation"* [9]. Defining Grids has always been difficult but nowadays there is a general agreement that Grids are distributed systems enabling the creation of Virtual Organizations (VOs) [2] in which users can share, select, and aggregate a wide variety of geographically distributed resources, owned by different organizations, for solving large-scale computational and data intensive problems in science, engineering, and commerce. Those platforms may include any kind of computational resources like supercomputers, storage systems, data sources, sensors, and specialized devices.

More recently researchers belonging to the European Network of Excellence "CoreGrid"[1] reached an agreement on the following definition: a Grid is *"a fully distributed, dynamically reconfigurable, scalable and autonomous infrastructure to provide location independent, pervasive, reliable, secure and efficient access to a coordinated set of services encapsulating and virtualizing resources (computing power, storage, instruments, data, etc.) in order to generate knowledge"*. This is a more modern service-oriented vision of the Grid that stems from the conviction that in the mid-long term the great majority of complex software applications will be dynamically built by composing services, which will be available in an open market of services and resources. In this sense, the Grid will be conceived as a "world-wide cyber-utility" populated by cooperating services interacting as in a complex and gigantic software ecosystem.

The XtreemOS [17] project started in 2006 follows yet another approach to the management of large and very dynamic grid systems: users logged in an XtreemOS box will transparently exploit VO-managed resources through the standard POSIX interface.

[1] `<http://www.coregrid.net/>`

The implementation of this new Grid vision introduces new challenges: transparency for users and application developpers, scalability, manageability, security, trust.

2 Addressing the challenge: the XtreemOS Position

While much work has been done to build Grid middleware on top of existent operating systems, little has been done to extend the underlying operating systems for enabling and facilitating Grid computing. In this light, XtreemOS [4] aims to be a first European step towards the creation of true open source operating system for Grid platforms. The XtreemOS operating system is based on Linux traditional general-purpose OS, extended as needed to support VOs, and to provide appropriate interfaces to the Grid OS services. In contrast to middleware approaches, XtreemOS is an operating system able to execute any kind of application, including unmodified existing applications.

The XtreemOS operating system provides for Grids what a traditional operating system offers for a single computer: abstraction from the hardware and secure resource sharing between different users. It thus simplifies the work of users belonging to virtual organizations by giving them the illusion of using a traditional computer while removing the burden of complex resource management issues of a typical Grid environment.

2.1 Fundamental Concepts

The design of XtreemOS has been guided by two fundamental concepts: transparency and scalability that imply in particular three essential properties as explained in the remainder of this section:

- Interoperability: complying with all major standards such as POSIX and SAGA,
- Dependability: providing reliability and high availability through checkpointing and replication,
- Security: ensuring trust and integrity according to customizable policies.

2.1.1 Transparency

Transparency in XtreemOS can be seen from two different points of view: the user and the application. We consider what we mean by transparency in each case.

Transparency for the User

A traditional user will have the feeling that is still working on a Linux machine. For instance, traditional Linux commands will be used instead of new ones: jobs (including legacy and grid-unaware applications) will be submitted like regular processes (just by writing the program name); checking the status of jobs will be done by using the traditional `ps` command; and so on. This means that XtreemOS will bring the Grid to standard Linux users. It is important to mention that submitting a job to the Grid requires some Grid-parameters such as the resources needed. These parameters will be predefined by application developers, and/or learned automatically by XtreemOS. On the other hand, we will also allow Grid-aware users to define these Grid parameters themselves to offer maximum power to expert users.

Another way in which XtreemOS will make the Grid transparent to users is that there will be no limit in the kind of applications that will be supported. Unlike most Grid systems that only allow batch jobs, XtreemOS will allow interactive (both terminal and Xwindows) applications to be submitted to the Grid in the same way they are launched in a Linux box.

Regarding user sessions, XtreemOS will offer a mechanism that once the user has logged into the VO, all commands will become Grid commands and thus the user will not have to worry about any of the Grid aspects anymore. In other words, logging into XtreemOS will launch a Grid-aware shell that transparently takes care of all Grid-related issues.

Finally, XtreemOS will allow VO to be built to either share or isolate resources from the rest of the world. This will be defined by the administrator of the VO, and thus it will be transparent to users. For instance, in an environment where isolation is vital, it will be handled by the VO management and the user will run jobs in an isolated environment without having to worry further.

Application Transparency

XtreemOS will also make Grid executions transparent for applications, and application developers. First, the job semantics will be very similar to the process semantics. This means that XtreemOS will offer a hierarchy of jobs (much in the same way as the Linux process hierarchy) and the same system calls (same interface) used to manage processes will be able to handle jobs (i.e. wait for a job, send signals to a job, etc.). Furthermore, to simplify the task of programming, XtreemOS will treat process within a job in the same way Linux treats threads within a process (thus the programmer will not need to learn new relationships).

Another way in which XtreemOS will make the Grid transparent to applications is that files will be stored in XtreemFS, using a Grid file system, and applications will be able to access these files regardless of their physical location using the POSIX interface and semantics.

In addition, XtreemOS will offer transparent fault tolerance to applications. If a resource fails, the application will recover automatically and transparently from the user.

Finally, XtreemOS will also make clusters of computers transparent to applications because each cluster in the system will offer a *single-system image* to the applications (i.e. single login point, distributed shared memory, etc.)

2.1.2 Scalability

Scalability is a key property of the XtreemOS operating system. XtreemOS has potentially to deal with a large number of resources (millions of nodes) owned by different providers and located in different sites (possibly thousands of location/administrative domains). It will be used by a large number of users (thousands of users) executing altogether a very large number of applications of various kinds (data intensive versus compute intensive), some of them being large scale applications spanning multiple Grid nodes and requiring a large amount of resources (individual jobs may span thousands of nodes). Multiple virtual organizations may rely on a Grid infrastructure, each with its own members, administrator, resources and policies. Thus XtreemOS has to be able to scale to these large sizes. We consider some of these factors in more detail.

Heterogeneity XtreemOS has to deal with a large amount of heterogeneous Grid nodes (resources) interconnected by various wired or wireless networks (e.g. WAN, LAN, SAN). These networks are heterogeneous from the performance point of view (latency, bandwidth, jitter) and performance is variable depending on the load. Resources are heterogeneous from the hardware point of view. There are different kinds of Grid nodes from the point of view of computer architecture: individual PC, clusters, high-performance computers and mobile devices. Grid nodes are based on different processor types and have a different amount of local resources (memory, disk, number of processors and cores). Grid nodes are also heterogeneous from the software point of view. They may run a different version of an operating system and they may be configured in different ways, for instance using the same (or same version of) libraries. Finally, Grid nodes belonging to different sites are independently managed and there is no reason to assume that the administration policies would be the same in these sites.

XtreemOS must be capable of running on a wide variety of different platforms, ranging from powerful servers down to simple PDAs or mobile phones with only little computational power. Hence, there will be several flavors of XtreemOS, but all with the same consistent set of services and the same interfaces.

Dynamicity Grid nodes may join or leave the system at any time based on decisions of their administrator or user. This may happen with a prior announcement, via a sign on/off, or without, for example a crash. A given Grid node may be temporarily disconnected as networks failures may occur at any time. Some Grid nodes, such as laptops, PDA, or mobile phones may suddenly be disconnected and later be re-joined with different data.

VO Models and Dynamicity in VOs There is not a single VO model: VOs may be long-term static VOs or short-term dynamic VOs. While a static VO life-time may be several months or years (corresponding to the duration of a collaboration), the life-time of a dynamic VO may be limited to the life-time of an application. A VO may be statically created by a VO administrator (static VO) or dynamically created by an application (dynamic VO). VOs also differ in their policies. A VO is created, evolves and is finally dissolved. Several kinds of modifications may happen during the VO life cycle: the addition or removal of institutions participating in the VO; the addition or removal of a site; or the addition or removal of a resource or of a member. Policies may also change over time, and the resources within a VO may experience failures. So while still belonging to the VO, these resources may be temporarily not accessible, leading to another form of dynamicity within a VO.

XtreemOS will support different VO models [24], and scalability issues can be seen w.r.t. the performance of the system, and its ability to adapt to changes. As a resource node may provide access to thousands of grid users from multiple VOs, the local operating system must still provide strong isolation properties, such as the existence or failure of an application shall not affect security and performance of applications from different VOs running on the same node).

When VOs are dynamically created or changed, e.g. when some resources fail, maintaining consistency of static local configurations becomes a complex task and heavy administrative burden, even if done automatically. Thus, in order to support large numbers of users in such a dynamic environment, implementation solutions relying on local configuration files which statically contain user/resource information should be avoided. Resources and users belonging to various administrative domains, scalability means keeping the autonomous management of user accounts and resources by the domain system administrator.

Scalability of Services XtreemOS services should be designed to scale with the number of entities (e.g. resources, users, applications, VO, sites) and their geographical distribution. On one hand, they should be fully distributed, avoid any contention points and save network bandwidth for the sake of performance. On the other hand, they should be able to securely run over multiple administrative domains. They should adapt to the evolution of the system composition coping with the dynamicity of a Grid. As a result, it will not be possible to maintain a global view of the system. XtreemOS services will also have to be self-managed services automatically dealing with events such as a node joins, leaves, or crashes. To deal with the high churn of nodes characterizing a large scale distributed system such as a Grid, service migration should be made transparent to users. Finally, there should be no single point of failure in XtreemOS operating system. Thus, XtreemOS critical services will need to be highly available.

Scalability of application execution management To scale with the number of applications, the application execution management service will be distributed within the scope of a VO. As it is not possible to get an optimal scheduling in a large scale dynamic system where a very large number of users share resources for the execution of a potentially very large number of applications of different kinds, XtreemOS scheduling philosophy will be best effort with a job centric scheduler.

Scalability of the Grid composition service A node in XtreemOS should be able to communicate with any other node in the same VO, for example, to find resources for executing a job. As the number of nodes in a Grid may be very large, it is not conceivable that a node keeps information on all other nodes it may communicate with. For the sake of scalability, in XtreemOS, a node will only keep information on a few nodes. The intersection between information maintained by each node on other nodes belonging to the same Grid should be large enough to support simultaneous failures of multiple nodes. This means that XtreemOS will be a highly decentralized system in the same spirit as Peer-to-Peer systems that are able to cope with node dynamicity.

Scalability in data management XtreemOS data management service will be able to deal with large amounts of data stored in geographically distributed data storage units (in different administrative domains) and accessed from any Grid node. It will manage files (volumes) that are shared by Grid users from different VOs. XtreemOS will provide a Grid file system providing efficient data access and data high availability. The XtreemOS file system *XtreemFS* will perform file access control and will guarantee secure data management. To be scalable, the management of the Grid file system will be highly decentralized.

3 Who can benefit from XtreemOS

The main goal of a new approach to Grid support such as XtreemOS is to provide real advantages to end users over conventional Grid middleware. In this section, we consider the general advantages of

XtreemOS for all users and then the more specific advantages for three key classes of user: end users, systems administrators, and application programmers.

3.1 Advantages for all users

A key advantage of the system for all classes of users is XtreemOS's approach to handling heterogeneity in systems. This has three main aspects:

Heterogeneous applications. XtreemOS is designed to handle a wide range application types. At one end of the spectrum, large scale scientific collaborations tend to be widely geographically dispersed with a large number of institutions and last a long time, with a very general goal and relatively straightfoward security requirements. At the other end, commercial applications in business data centres typically involve a small number of partners on short time scales and tightly directed goals, often controlled by a workflow, and have key requirements for isolation and data security. By providing a sufficiently general and flexible infrastructure, XtreemOS aims to support this range of applications, which is demonstrated in the case studies within the project.

Heterogeneous platforms. XtreemOS aims to provide a single operating system, which will operate on workstations, clusters and mobile devices. Thus a collaboration can work across these devices transparently, integrating a range of different platforms together within a single management system, working to common protocols.

Heterogeneous Security Systems The different security mechanisms, which traditional systems use is a significant barrier to practical Grid computing. The mechanisms can be difficult or impossible to work together, and make it hard to establish trust between entities. The XtreemOS security model uses a common global security mechanism which can be translated to work with local security infrastructures. Global security decisions are made using the common system with their local enforcement. Thus a common trust and security basis can be established while not interfering with local security policy.

We break down the XtreemOS users into a number of different groups and consider the benefits which each group would accrue from XtreemOS.

3.2 Advantages for end users

Two kinds of end users can be distinguished: users launching applications (called here application users), and service administrators (users launching applications that are in fact services such as a web server or a database). Some of these users will be experts, others novices or non-computing specialists. An objective of XtreemOS is to make the Grid invisible for non-expert users. For expert users, it may not be desirable to make the Grid fully invisible. The expert user may be able to provide useful information to the system, for instance, to optimize the execution time of his/her application. XtreemOS thus provides a number of advantages to end users.

Ease of use. A number of XtreemOS features provide a user with an easy migration from a familiar local Linux environment. The use of a Single-Sign On mechanism to the Grid within a virtual organization allows the Grid user to access all the resources on the Grid with a single use of Grid user name and password, and then the underlying certificate mechanism will authenticate the user against resources appropriately. This provides a convenient mechanism without the need to manage multiple certificates explicitly. This seemless access without multiple authentication challenges also enhances the user's view of transparent access to resources, which can be called as if they appear on the local system, regardless of their actual location. Further as user commands in XtreemOS support Posix, as far as possible, one familiar interface can be used to call local and global resources.

Secure and reliable application execution XtreemOS provides a number of features which support secure and efficient execution of applications. XtreemOS provides a fine-grained control of access to resources on the available computing nodes, allowing specific data objects to be managed separately. This is supplemented by efficient application execution by identifying and utilising free compute nodes within the Virtual Organization, so maximising the use of the available computational power, and the execution is then monitored accurately by auditing services across the Virtual Organization.

Platform transparency XtreemOS provides a single interface, which can accomodate cluster machines and mobile devices as well as conventional workstations, so it supports a ubiquitous access to services, applications and data across devices within one system. Thus end users do not need to configure their application to work on a particular platform.

3.3 Advantages for administrators

Another key set of XtreemOS users are system administrators. System administrators can include resource administrators, responsible for a particular computing resource, or Virtual Organization administrators, responsible for supporting a collaboration within a community. Again, XtreemOS provides advantages for both groups, as follows.

Advantages for local resource administrators. XtreemOS supports the autonomous management of local resources. Security policies to access local resources, and accounts for local users, can be controlled locally by their own local administrators. Then, when the resource participates in virtual organisations, these local configurations can be propagated across the virtual organization and thus respected by other node. Further features, which support strong isolation, including node virtualisation and virtual firewalls are being added to support high integrity resources and applications, which also satisfy the requirements of local administration on the higher security on such resources.

Advantages for Virtual Organization administrators. XtreemOS provides a scalable virtual organization tool, which can be controlled via Virtual Organisation roles and policies, which respect the needs of local resources. XtreemOS also considers the whole lifecycle, supporting VO establishment, change and dissolution in a single controlled environment. XtreemOS will thus give VO administrators the tools to flexibly run their VOs while having the assurance that local requirements will be respected across the Grid.

3.4 Advantages for application programmers

XtreemOS is a Grid OS which can support a wide range of applications and services. It does not assume any particular architecture beyond the basic Grid and Virtual Organization services, many of which are transparent to the users, and is thus designed not to be limited to a particular programming paradigm. Current Linux applications can run with little or no modifications, so legacy applications to be executed in a Grid without modification, or recompilation. It is also intended that current Grid applications will also run with little or no modifications. XtreemOS supports the emerging standard OGF standard API 'SAGA', with currently both the C and Java languages supported. As other Grid systems will also support this interface, then a Grid application defined for one system should also run any other SAGA compliant system. This would potentially include applications running across a combination of XtreemOS and other Grid middleware.

4 Highlights and Achievements

4.1 XtreemOS software

At the end of the project, the main achievements in terms of software are the advanced versions of the three flavours of XtreemOS system for PCs, clusters and mobile devices (XtreemOS 2.1.2 (stable) and 3.0 (beta) releases). XtreemOS 3.0 is not only available for PDA but also for smart phones and notebooks. The XtreemOS 3.0 release has been packaged for Mandriva, RedFlag and Debian Linux distributions and are available as open source software under GPL/BSD license. Packaging for Open Suse and other popular RPM based Linux distributions such as CentOS is in progress. XtreemOS system comes with user (`http://www.xtreemos.eu/documentation-support/userguide.pdf`), administrator (`http://www.xtreemos.eu/documentation-support/AdminGuide.pdf`) manuals and a quick installation guide (`http://www.xtreemos.eu/project/publications/VMInstall.pdf`). Interfaces between the various system services are described in a dedicated document (`http://www.xtreemos.eu/documentation-support/Release2-1-interfaces.pdf`). We considerably simplified the installation, configuration and deployment process of an XtreemOS Grid. We developed a tool, xos-autoconfig, automating the configuration of Grid nodes.

4.2 Applications and Use Cases

Table 1 provides an overview of the set of applications which were used to evaluate XtreemOS development. Several applications have been added by various partners for experiments during the project extension phase. These applications are marked by (*). The total set of 22 applications covers a wide spectrum of fields with practical relevance. Reference applications fall in various areas: bio-informatics, enterprise solutions, tomography applications, computer aided engineering, computational fluid dynamics, 3D data analysis and visualization, electromagnetism, particle physics, numerical simulations, virtual presence, cloud image archive and economics. Hence, the perspectives of end-users from different industrial and academic domains could be respected during evaluation. Most applications with open source licenses have been packaged and bundled with XtreemOS releases to provide an easy hands-on experience to new XtreemOS users.

Table 1: Applications in WP4.2. Applications marked by (*) have been added for experimentation during the project extension period

Partner	Application Name	Short Name	Application Area
BSC	COMP Superscalar	COMPSS	Bio-informatics
BSC	SpecWeb	SPECWEB	Enterprise solutions
CNR	XOS-GATE Tomographic Applications (*)	XOSGATE	Numerical simulations
EADS	Amibe/JCAE	JCAE	Computer aided engineering
EADS	openFOAM (*)	OPENFOAM	Computational Fluid Dynamics
EADS	Paraview (*)	PARAVIEW	3D data analysis and visualization
EADS	Elfipole	ELFIPOLE	Electromagnetics
EDF	Moderato/Maestro	MODERATO	Particle physics
EDF	OpenTURNS	OPENTURNS	Sensitivity and reliability analysis
EDF	Zephyr	ZEPHYR	Fluid mechanics
EDF/INRIA	Salomé (*)	SALOME	Numerical simulations
INRIA	Environmental Application (*)	ENVIRON	Environmental simulations
SAP	SAP NetWeaver Search and Classification	TREX	Enterprise solutions
SAP	SAP MaxDB replayer	MaxDB	Enterprise solutions
SAP	Rule-based System Management	RBSM	Enterprise solutions
STFC	BioLinux Application (*)	BIOLINUX	Bio-Informatics
TID	TID Instant Messaging Application	IMA	Instant messaging
TID	Job Management Application	JOBMA	XtreemOS job management
UDUS	Wissenheim	WISS	Virtual Presence
VUA/ZIB	Cloud Computing	CLOUD	Image Archive
XLAB	Galeb	GALEB	Economics, optimization

4.3 Evaluation

The reference applications were used to perform experimental evaluations of selected XtreemOS components or of entire XtreemOS prototypes and packages.The evaluation focused on the installation of XtreemOS developments and on the execution of the reference applications using the functionalities provided by the components under test. Typically, the evaluation targeted aspects like fulfillment of requirements, stability, performance, scalability and usability. As opposed to the tests carried out by developers or at packaging level, the evaluation with reference applications allowed for testing the XtreemOS from the user perspective using input data with practical relevance. The evaluation has also been carried out regarding XtreemOS as foundation for cloud computing, comparison with other Grid solutions and usability evaluation of the XtreemOS MD flavour.

Results from the evaluation are reported in public deliverables. The interested reader is invited to consult the two most recent evaluation reports (http://www.xtreemos.eu/project/publications project-deliverables/d4-2-6.pdf and http://www.xtreemos.eu/project/publicat D4.2.7.pdf).

4.4 XtreemOS Open Permanent Testbed

Throughout the project lifetime, we devoted a lot of efforts towards the XtreemOS open source community to further disseminate the XtreemOS technology. One of the project main achievements is the XtreemOS open testbed. Anyone can request an account and execute applications on an XtreemOS Grid formed of PCs and clusters provided by XtreemOS partners. At the end of the XtreemOS project, the open testbed contains 14 nodes (see Fig. 1): 2 nodes run global core services and 13 resource nodes support user applications. The core nodes are located at STFC, UK and INRIA, France. The resource nodes are distributed as follows: 2 nodes at INRIA (France), 1 node at VU Amsterdam (The Netherlands), 1 node at the University of Duesseldorf (Germany), 3 nodes at University of Ulm (Germany), 6 nodes at CNR (Italy), 1 node at ZIB (Germany). In the future, any institution will be able to join XtreemOS Grid by providing resource nodes. We have also made available to the community ready to use virtual images of the XtreemOS system for KVM and Virtual Box. Moreover, we have produced XtreemOS images for the automatic deployment of an XtreemOS Grid on Grid 5000 large-scale experimentation platform.

Figure 1: XtreemOS open permanent testbed

4.5 XtreemOS Sustainability

XtreemOS technology will be sustained in open source. After the end of the project, two full-time engineers hired at INRIA have been providing support to the XtreemOS open source community and maintaining the PC and cluster flavours in collaboration with XtreemOS components developers. Furthermore, XtreemOS project results will be leveraged in the area of cloud computing in the framework of the FP7 Contrail IP project started in October 2010 and in which several partners from the XtreemOS consortium are involved.

5 The Results

XtreemOS system integrates a comprehensive set of cooperating system services [17, 4]. XtreemOS software components range from Linux kernel modules to application-support libraries. The overall layering of these components, grouped within software packages, is shown in Figure 2. XtreemOS provides a number of novel features described in the following.

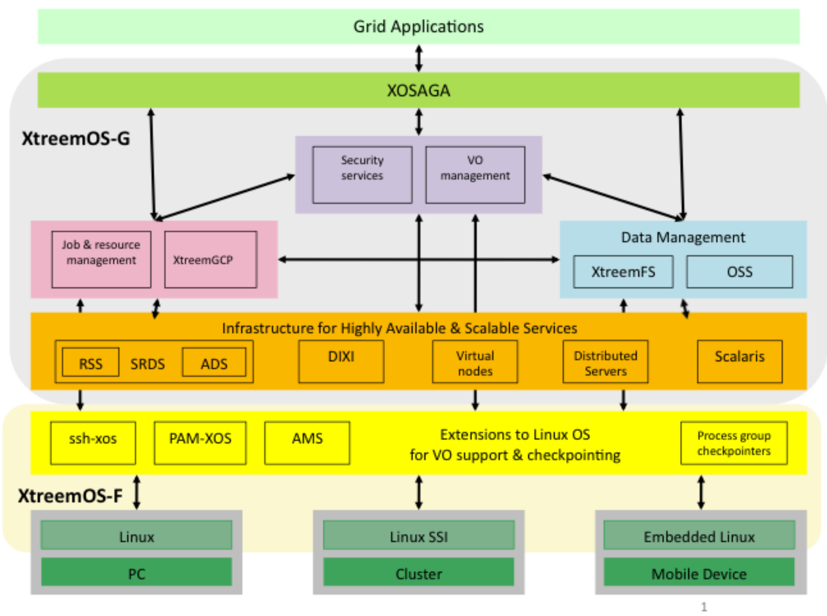

Figure 2: XtreemOS Architecture

5.1 VO Management and Security

XtreemOS supports various VO models, used in scientific as well as business scenarios and multi VO scenarios. User management and resource management are independent: there is no need to configure resources when new users are registered in VOs [3]. Several functionalities in the credential distribution authority (CDA) were automated, making it easier for administrators and users to manage certificates. CDA is now able to generate certificates for XtreemOS core services allowing the deployment of an XtreemOS Grid to be (optionally) automated. Moreover, for the sake of scalability, Grid users certificates are dynamically mapped onto Linux user accounts that are created on the fly for the lifetime of their jobs on their execution nodes. XtreemOS provides Single-Sign-On (SSO): when users perform a *login* within a VO, they receive credentials recognized by all resources of the VO without any need to re-authenticate. Resource access security in XtreemOS is policy-driven: access rights to a resource are evaluated from policies provided by users, VOs and resource providers; the latter remaining always in control [1]. In XtreemOS, the isolation degree (for QoS, performance and security) can be customized using native Linux mechanisms to execute applications: cgroups, containers, name spaces and virtual machines.

5.2 Resource Discovery and Selection

The resource discovery mechanism within XtreemOS is based on the SRDS (Service/Resource Discovery Service) distributed information service [2] comprising of the Resource Selection Service (RSS) and the Application Directory Service (ADS). RSS [5, 22] takes care of performing a preliminary selection of nodes to allocate to an application, according to range queries upon static attributes. It exploits a fully decentralized approach, based on an overlay network, which is built and maintained through epidemic protocols. As each node represents its own attributes in the overlay, failure management does not require any specific repair operation. ADS handles the second level of resource discovery, answering queries expressed as predicates over the dynamic attributes of the resources. ADS creates an application-specific "directory service" using the resources received by the RSS. Dynamic node attributes are periodically updated into the system. To provide scalability and reliability, distributed hash tables (DHT) techniques and their extensions to range and multi-attribute queries are used. ADS supports Scalaris transactional DHT and OverlayWeaver.

5.3 Application Execution Management (AEM service)

XtreemOS implements a job-oriented scheduling within the subset of resources obtained by the resource discovery mechanism. To ease the use of the Grid services, XtreemOS mimics the well-known Linux functionality as opposed to offering different abstractions and functionality, which are more oriented to the Grid. For instance, AEM [19] implements job control through signals. The reservation service implemented by AEM provides resource co-allocation allowing the execution of distributed/-parallel applications on multiple Grid nodes. Reservations can be dynamically modified. AEM also provides flexible and accurate job monitoring. It manages job dependencies, providing an interface to external workflow engines. AEM implements SchedFS, a novel scheduler based on the Vivaldi algorithm to allocate resources to a job close to the files it uses. Lastly, XtreemOS fully supports the execution of interactive applications. The XtreemGCP Grid checkpointing service [15] takes care of reliable execution of distributed applications that can take advantage of rollback recovery protocols. XtreemGCP can checkpoint/restart and migrate applications running on multiple heterogeneous Grid nodes. Various kernel checkpointers are supported through a library implementing a common checkpointer interface: BLCR, a checkpointer based on OpenVZ containers on PCs and LinuxSSI checkpointer on clusters.

5.4 XtreemFS Grid file system

XtreemFS is a distributed file system designed for deployment in wide-area environments spanning multiple locations in different administrative domains [11, 12]. It allows mounting an XtreemFS volume from any location, given the right permissions. XtreemFS is transparently integrated with the XtreemOS VO infrastructure in the form of dynamic user mappings and automatic mounting of home volumes. XtreemFS stores file data on several storage servers. It implements object-based file system architecture. The metadata of a file is stored separately from the file content on a metadata server. This metadata server organizes file system metadata as a set of volumes, each of which implements a separate file system namespace in form of a directory tree. XtreemFS is a full-featured file system that supports the full POSIX file interface, including extended attributes. In case of concurrent access by several distributed programs, it provides currently the NFS close-to-open consistency. XtreemFS provides efficient access to file data through file striping on multiple servers. It also manages data replication transparently to users. With XtreemFS users and administrators can take consistent snapshots of large-scale XtreemFS installations. This led us to implement versioning in XtreemFS internal services managing file metadata and data.

5.5 Volatile Data Management

The Object Sharing Service (OSS) eases the sharing of volatile data objects by transparently managing replicas and keeping them consistent [18, 16]. Grid applications can share objects through file system operations or by using customized functions. The latter includes support for speculative transactions, which alleviates network latency and avoids complicated lock management.

5.6 Distributed Servers & Virtual nodes

A distributed server is an abstraction that allows a group of server processes to appear as a single entity, with a single IP address, to its clients. Distributed Servers aim at allowing high-performance client-to-server communication, while being totally transparent to the clients. The only requirement is that the clients support the Mobile IPv6 protocol. The goal of virtual nodes is to provide fault-tolerant functionality to service-oriented applications with minimum effort for the service developers. Virtual nodes allow a programmer to choose a replication policy among several available ones, which include several flavours of passive and active replication [8, 7]. Virtual nodes and distributed servers are highly complementary to each other: virtual nodes provide fault-tolerant replication that is mostly transparent to the service developer, but lack an access method that makes fault-tolerance transparent to the clients. Distributed servers provide a solution for making the service replication transparent to the client. Although each service has its own utility when used in isolation, we see that merging both systems would in principle allow one to build fault-tolerant replicated services where the complexity of replication would be transparent to both the service developer and to the client-side application. To improve the reliability of XtreemOS system, we have applied the virtual node approach to replicate critical services in XtreemOS such as the AEM job and reservation managers and the CDA (Credential Distribution Authority) server.

5.7 XtreemOS API

The XtreemOS API has to serve three classes of applications: existing Linux applications, using POSIX interface, existing Grid applications, using the OGF SAGA interface [10], new applications, using functionality uniquely provided by XtreemOS. We have defined an API name space called XOSAGA (XtreemOS extensions to SAGA) that mirrors the SAGA API name space. XOSAGA contains only those packages, classes, and interfaces that require XtreemOS-specific extensions to SAGA. Together, SAGA and XOSAGA form the XtreemOS API, implemented in C/C++, Java, and Python [6].

5.8 XtreemOS Cluster Flavour

The XtreemOS cluster variant is based on LinuxSSI, which implements a full Single System Image (SSI) operating system for computing clusters. A full SSI operating system globally manages all cluster resources to give the illusion that a Linux cluster is a single Linux Symmetric-multiprocessing (SMP) node, allowing the execution of unmodified legacy sequential or parallel applications and system administration tools. LinuxSSI [25] leverages Kerrighed SSI technology. LinuxSSI nodes can be dynamically added or removed in a running cluster easing cluster maintenance. Moreover, checkpoint/restart mechanisms now cover system V IPC (Inter-process communication) objects, semaphore array, message queue and shared memory segments, which are checkpointed independently from the application checkpoint. Advanced scheduling strategies that can adapt the scheduling policy to the cluster load have been developed in the PlugProPol framework [23]. The CLUSTERIP mechanism has been ported to LinuxSSI for UDP and TCP over IPv4 making transparent service migration between cluster nodes. Additionally, LinuxSSI implements kDFS [13] a distributed / parallel file system exploiting the disks attached to compute nodes and providing support for handling persistent states during checkpoint/restart operations [21].

5.9 XtreemOS Mobile Device Flavour

XtreemOS also provides a mobile device flavour (XtreemOS-MD), which fully integrates most of XtreemOS functionalities, giving users on the move full access to the XtreemOS Grid [14, 20]. This kind of approach is much more scalable than gateway or Grid portal solutions for mobile access, as it eliminates the potential bottlenecks and single-points of failure of these gateways. Moreover, mobile Grid applications are able to run transparently with little or no modifications in mobile devices, due to the inclusion in XtreemOS-MD of OGF's standard SAGA API. Support for context-aware applications is integrated in the advanced version of XtreemOS-MD for smart phones. With the advanced version of the mobile device flavour, a mobile device is not only a client but also a resource provider thanks to the shared resource service we developed from scratch. We cover the sharing of input/output devices such as GPS in a generic approach implemented by means of a plug-in component for the AEM core service. This is a feature unique to XtreemOS. XtreemOS provides not only a full Grid operating system for mobile devices, but also a set of open source software modules that can be easily integrated into any modern mobile Linux distribution, by avoiding excessive reliance on any specific mobile platform.

6 The Demonstrators

In order to show the benefits of the fully integrated XtreemOS operating system, we developed multiple demonstrators with the main goal to motivate the potential users to download the system and try it out themselves. Since June 2009, the demonstrators have been shown at 13 public events such as scientific conferences and trade fairs. Most of the demonstrators are also available as screencasts on the project web sites.

The first group of demonstrators, the *featured use cases*, are available at `http://www.xtreemos.eu/documentation-support/screencasts/featured-xtreemos-use-cases`. These are intended for audience with no or little previous experience with grid or cloud computing and show various applications running smoothly on XtreemOS and benefiting from distributed resources available to them through XtreemOS. In other words, these demonstrations explain the "why" of XtreemOS. They include:

Geographical data processing: the computationally-intensive process of preparing geographical data for the Gaea+ client is run on XtreemOS. The advantages of XtreemOS are transparent use of multiple computational resources and ease of data management with the global XtreemFS file system. The graphical monitoring capabilities of XtreemOS are also shown.

Elastic Web Application Hosting: WS-Escape is an elastic system for web application hosting that runs on top of XtreemOS.

Wissenheim: Wissenheim is a distributed interactive 3D virtual world for edutainment and entertainment. The game is started by each participant by using the XtreemOS xconsole. The application loads the persistent game state information by using the XtreemOS grid file system (XtreemFS.) Subsequently, the scene graph is transparently distributed and replicated by the XtreemOS Object Sharing Service (OSS). Avatar interactions and scene graph changes are synchronized through speculative transactions provided by the OSS. In case of severe errors the application relies on the grid checkpointing service.

XtreemOS-MD Grid Media Player: GPlayer is an application developed by Telefonica for smartphones within the XtreemOS-MD context. This mobile application is a media player that uses the grid capabilities provided by XtreemOS to transcode any video to a more suitable format and/or size.

Sharing a smartphone's 3G connection: XtreemOS-MD provides the capability to share input-output resources of a smartphone, e.g. 3G access or GPS device, within the grid. We show how a N900 shares its 3G connection, using a p2p secure connection, with a user authorized in the XtreemOS grid.

XtreemOS-MD On-Demand File Sharing: XtreemOS-MD provides to mobile devices (like smartphones and PDAs) the capability of sharing the files stored locally, which will be uploaded to the grid file system (XtreemFS) just when requested by another user authorized in the XtreemOS grid. This feature is what we call *on-demand file sharing*.

This group is complemented by *other use cases*, which contain more applications but are not as polished as the first group, available at `http://www.xtreemos.eu/documentation-support/`

screencasts/other-use-cases/. Collectively, the use-case demonstrators have already been watched almost 3,000 times so far.

The final group of demonstrators is more technical and is available at http://www.xtreemos.eu/documentation-support/screencasts/technical-demonstrations/. In contrast to application demonstrations, this group explains the *how* of XtreemOS and includes:

Core demo scenario shows the most frequently used features of XtreemOS and demonstrates its ease of use and similarity to ordinary GNU/Linux.

Virtual Nodes demo shows how an e-mail server can be replicated with VirtualNodes and demonstrates fault tolerance by switching off individual replicas.

Checkpointing and restart demo saves a checkpoint of a simple job, kills and restarts it. Migration of a job to another node is shown next, followed by checkpoint and restart of a complex job consisting of two job units (processes) that communicate over sockets. The mechanism for checkpointing the state of the communication channels is explained as well.

Two XtreemFS videos demonstrate XtreemFS replication. First, replication and failover is shown with an unmodified MPlayer. A regular movie file stored on two XtreemFS OSDs (Object Storage Devices) is played. As one OSD is disconnected, the user only senses a short pause (depending on the size of MPlayer buffer) before the XtreemFS client transparently switches to another replica. The second demonstration shows how new replicas are created and how the on-demand replication strategy works.

The two XtreemFS videos have been watched over 17,000 times, the rest of the technical demos almost 4,000 times.

Finally, an XtreemOS grid can be set-up and configured in an interactive demonstrator called the "ad-hoc grid setup scenario". It shows the full process of adding client and resource nodes to an existing grid. Instructions and required software for executing this demonstrator are available on request from marjan.sterk@xlab.si.

7 Availability of Results

All XtreemOS technical deliverables are public and available on XtreemOS website (http://www.xtreemos.eu/project/publications), grouped by themes.

XtreemOS software is in open source development. The source code is hosted by the INRIA Gforge XtreemOS repository (https://gforge.inria.fr/scm/?group_id=411). All the flavours of XtreemOS system (for PC, cluster, mobile device) are available in open source and can be downloaded from the following webpage: http://www.xtreemos.eu/download. XtreemOS public website provides information and support to the XtreemOS open source community (http://www.xtreemos.eu/documentation-support). We have made available to the community ready to use virtual images of the XtreemOS system for KVM and Virtual Box. Moreover, we have produced XtreemOS images for the automatic deployment of an XtreemOS Grid on Grid 5000 large-scale experimentation platform.

We operate an open permanent XtreemOS testbed (`http://www.xtreemos.eu/open-testbed`). Anyone can request an account and execute applications on this testbed. In the future, any institution will be able to join XtreemOS Grid by providing resource nodes.

8 Potential Impact of Results

During the project over 150 conference papers and journal articles have been published, which are archived in the literature or available on line, to provide the academic and research communities with the conceptual basis and design details of the framework and implementations. Most of the 139 XtreemOS deliverables are public, providing the research community insight in XtreemOS technology design and development.

The dissemination of research concepts and design decisions, the contributions to technical standards (eg. SAGA), the support of open source components by user communities (eg. Kerrighed, Scalaris, XtreemFS), the commercial exploitation of individual components (eg. Kerrighed, Scalaris) and major functionality, together show a significant impact of the project as explained in the remainder of this section. Work done on the XtreemOS SAGA interface has been influential in the OGF SAGA working group led By Thilo Kielmann, a member of XtreemOS consortium. XtreemFS open source community has been created during the course of the XtreemOS project, while Kerrighed and Scalaris that were existing prior to the XtreemOS project start have significantly benefited from XtreemOS project. Kerrighed, Scalaris and XtreemFS open source communities have grown during the life time of XtreemOS project showing an increasing interest from the open source community in these components, XtreemOS contributed to make more robust and more usable in addition to extending their set of features. For instance, Kerrighed has been extensively used as part of the NuGO European Network of Excellence in bio-informatics involving several institutions from UK, Belgium and Italy. XtreemFS that can be used as a standalone component and whose client is available not only for Linux but also for MacOS and Windows machines has raised a lot of interest from industry (e.g. Google R&D) and from the research community. It is used in the framework of a number of on-going research projects such as the Scalus MCITN European project on storage and the Contrail FP7 integrated project on cloud computing. It is also used as a building block in MOSGRID and Compatible-One national projects in Germany and France. Kerrighed (backed by Kerlabs company), Scalaris (backed by On Scale Solutions company) and XtreemFS will be sustainably maintained after the end of XtreemOS project.

XtreemOS Grid distributed operating system has been experimented in various domains through collaborations with external partners. For instance, in the bio-informatics area, INRIA is collaborating with INRA (the French institute for agricultural research) on using XtreemOS system for the research on epidemiology. STFC is also collaborating with biologists in UK and is planning to work on the integration of XtreemOS in the Bio-Linux distribution. XtreemOS has also been experimented for the execution of tomography applications within a collaboration between two teams at CNR. In the context of the COOP project funded by the French research agency, we aim at executing the SALOME platform for numerical simulation on top of XtreemOS. Experiments with an application in the nuclear domain are planned.

XtreemOS is also exploited in the context of EIT ICT Labs. Fully documented testbeds based on XtreemOS and IaaS cloud technologies are being built and made available in open source. Two full-time engineers are funded by INRIA under the XtreemOS-Easy technological development action to support XtreemOS open source community after the end of XtreemOS project. They interact with XtreemOS developers to maintain the software and develop tools to further ease the use of XtreemOS system.

In the Computer Science research community, a number of Master and PhD students are using XtreemOS in their research work.

9 Lessons Learned during the Project

9.1 High quality software development is challenging

The system architecture should have had an even more prominent place in the project work, with more feedback to and from the other, software-producing work packages. This might have saved some software integration efforts.

It turned out that more efforts than expected were required in the integration phase. The integration process was longer than expected not only for the first release but also for the subsequent ones. We learnt that the release manager has a crucial role to play in coordinating the integration efforts.

It is also essential to produce documentations early enough in the process on component interfaces and also user and administrator manuals. It was difficult to keep these documents continuously synchronized with the evolving software. The organization of weekly telco involving developers and packagers helped a lot in the integration and packaging process.

It is important to have integrators (not involved in XtreemOS implementation) working full time on testing the software and reporting to the developers. It is essential to document a common testing process and that all developers test the software in the same environment.

A positive experience was the close cooperation with the developers. The evaluation reports and in particular the results of the evaluation were communicated to developers through various channels, including wiki pages, mailing lists, bug trackers, phone calls and finally a deliverable. Bugs and other issues reported through the evaluation were treated carefully to drive product improvements and guide the evolution of the XtreemOS product which became increasingly mature and complete.

The major lesson learned during the project is that experimenting the operating system layer at large scale is tricky. Users must have full access to the hardware. This implies major security issues. The security on production platforms is enforced by the operating systems. Operating system experimentations are possible only on configurations where security enforcement is based on external equipements: security in Grid'5000 is enforced using networking equipments.

Another lesson learned during the project is that Grid systems should provide means to react to misbehaving nodes or sites. The first version of the open testbed deployed on multiple partners site was administrated according to the grid philosophy: each partner was in charge of administrating his own nodes. Noone had administration rights on the whole system. This scheme did not work in XtreemOS.

9.2 Attracting external users is another huge challenge

A high-quality documentation and responsiveness from internal developers are key features to attract people from outside. A simple installation/configuration process is also crucial.

The real complexity of providing a Linux-based OS for mobile devices with basic Grid capabilities was underestimated (esp. when comparing with the PC and cluster flavors).

Another major lesson learned was that implementing demonstrators too late during the course of the project results in missed opportunities to show them and thus less impact than if the demonstrator had been available on time. Thus it would have been better to plan more application-based scenarios so that they could be implemented with earlier releases of XtreemOS and also to more actively encourage the developers to fix the specific bugs that are blocking the demonstrators. It would have been better to have these videos and demonstrators earlier to be used for communication activities. But developing an operating system (OS) takes a long time and demo applications cannot be tested before the OS is available.

9.3 Other lessons learned

The iterative approach to API definition was key to its success. Pre-planned API's would not have worked. Thanks to the considered useful XOSAGA API, XtreemOS has attracted quite some attention from grid application users.

The experience built during XtreemOS will be precious for conducting future research and E.U. projects.

The principle of adhering to guidelines, such as the OGF Grid Certificate Profile used in the CDA, gave the advantage of easing the interoperability work between XtreemOS and gLite.

Although current organization infrastructures support IPv6, the equipements are generally not cofigured to allows its exploitation. IPv6 is generally available inside partner's sites. But, in general, it cannot be used between sites mainly because of limitations in firewall configurations.

A professional communication manager (from an industrial partner) would have been useful, especially for developing a professional marketing-like communication strategy.

10 Partners

Organisation name	**Country**
Caisse des dépôts et consignations	FR
Institut National de Recherche en Informatique et Automatique	FR
Science and Technology Facilities Council	UK
Consiglio Nazionale delle Ricerche	IT
European Aeronautic Defence and Space Company	FR
Electricité de France	FR
Edge-IT *(until August 2010)*	FR
NEC Deutschland GmbH	DE
SAP	DE
Barcelona Supercomputing Center - Centro Nacional de Supercomputación	ES
Universitaet Ulm	DE
Vrije Universiteit Amsterdam	NL
XLAB	SI
Konrad-Zuse-Zentrum für Informationstechnik Berlin	DE
Institute of Computing Technology of Chinese Academy of Sciences	CN
Red Flag Software	CN
Telefónica I+D	ES
Universitaet Düsseldorf	DE
Kerlabs *(since June 2009)*	FR

11 Contacts

Administrative and Financial Coordinator

Jean-François Forté, Caisse des Dépôts et Consignations
15 Quai Anatole France
75007 Paris, France

Scientific Coordinator

Christine Morin, INRIA
Campus universitaire de Beaulieu
35042 Rennes cedex, France

Contact: contact@xtreemos.eu

Public web site: http://www.xtreemos.eu

References

[1] Benjamin Aziz, Alvaro Arenas, Ian Johnson, Artač Matej, Aleš Černivec, and Philip Robinson. Management of security policies in virtual organisations. In *International Conference on Security and Cryptography*, Athens, Greece, July 2010. SECRYPT 2010.

[2] E. Carlini, M. Coppola, P. Dazzi, D. Laforenza, S. Martinelli, and L. Ricci. Service and resource discovery supports over p2p overlays. In *Proc. of the International Conference on Ultra Modern Telecommunications & Workshops, 2009*, Hersonissos, Crete, Greece, October 2009.

[3] Massimo Coppola, Yvon Jégou, Brian Matthews, Christine Morin, Luis Pablo Prieto, Oscar David Sánchez, Erica Y. Yang, and Haiyan Yu. Virtual organization support within a grid-wide operating system. *IEEE Internet Computing*, 12(2):20–28, 2008.

[4] Toni Cortes, Carsten Franke, Yvon Jégou, Thilo Kielmann, Domenico Laforenza, Brian Matthews, Christine Morin, Luis Pablo Prieto, and Alexander Reinefeld. XtreemOS: a vision for a grid operating system. Technical report 4, XtreemOS European Integrated Projet, May 2008.

[5] Paolo Costa, Jeff Napper, Guillaume Pierre, and Maarten van Steen. Autonomous resource selection for decentralized utility computing. In *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems (ICDCS)*, Montreal, Canada, June 2009.

[6] Mathijs den Burger, Ceriel Jacobs, Thilo Kielmann, Andre Merzky, Ole Weidner, and Hartmut Kaiser. What is the price of simplicity? a cross-platform evaluation of the saga api. In *Euro-PAR 2010*, pages 392–404, Ischia, Italy, 2010. Springer Verlag. LNCS 6271.

[7] Jörg Domaschka, Thomas Bestfeisch, Franz J. Hauck, Hans P. Reiser, and Rüdiger Kapitza. Multithreading strategies for replicated objects. proc. of 9th international middleware conference, leuven, belgium, december 1-5, 2008. pages 104–123, 2008.

[8] Jörg Domaschka, Christian Spann, and Franz J. Hauck. Virtual nodes: a re-configurable replication framework for highly-available grid services. In *9th International Middleware Conference, Leuven, Belgium, December 1-5, 2008, Companion Proceedings*, pages 107–109, 2007.

[9] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan-Kaufmann, 1999.

[10] Tom Goodale, Shantenu Jha, Hartmut Kaiser, Thilo Kielmann, Pascal Kleijer, Andre Merzky, John Shalf, and Christopher Smith. A simple api for grid applications (saga). Recommendation GFD.90, Open Grid Forum, 2008.

[11] Felix Hupfeld, Toni Cortes, Björn Kolbeck, Jan Stender, Erich Focht, Matthias Hess, Jesus Malo, Jonathan Marti, and Eugenio Cesario. The xtreemfs architectureâĂŤa case for object-based file systems in grids. *Concurrency and Computation: Practice and Experience*, 20(17):2049–2060, 2008.

[12] Felix Hupfeld, Björn Kolbeck, Jan Stender, Mikael Högqvist, Toni Cortes, Jonathan Marti, and Jesús Malo. Fatlease: scalable fault-tolerant lease negotiation with paxos. In *HPDC '08: Proceedings of the 17th international symposium on High performance distributed computing*, pages 1–10, New York, NY, USA, 2008. ACM.

[13] Adrien Lèbre, Renaud Lottiaux, Erich Focht, and Christine Morin. Reducing kernel development complexity in distributed environments. In *Euro-Par 2008*, LNCS, pages 576–586, Las Palmas de Gran Canaria, Canary Island, Spain, August 2008. Springer.

[14] Alvaro Martínez, Santiago Prieto, Noé Gallego, Ramon Nou, Jacobo Giralt, and Toni Cortes. Xtreemos-md: Grid computing from mobile devices. In *MOBILWARE*, pages 45–58, 2010.

[15] John Mehnert Spahn, Thomas Ropars, Michael Schoettner, and Christine Morin. The Architecture of the XtreemOS Grid Checkpointing Service. In *15th International Euro-Par Conference*, Delft Pays-Bas, August 2009.

[16] Kim-Thomas Moeller, Marc-Florian Mueller, and Michael Schoettner. Adaptive conflict unit size for distributed optimistic synchronization. In *Proceedings of the International European Conference on Parallel and Distributed Computing (Euro-Par 2010)*, Ischia, Italy, August 2010.

[17] Christine Morin. XtreemOS: A grid operating system making your computer ready for participating in virtual organizations. In *ISORC '07: Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, pages 393–402, Santorini Island, Greece, May 2007. IEEE Computer Society.

[18] Marc-Florian Mueller, Kim-Thomas Moeller, and Michael Schoettner. Transactional data sharing in grids. In *Proc. of the International Conference on Parallel and Distributed Computing and Systems*, Orlando, USA, November 2008.

[19] Ramon Nou, Jacobo Giralt, Julita Corbalan, Enric Tejedor, J.Oriol Fito, J.Maria Perez, and Toni Cortes. Xtreemos application execution management: A scalable approach. In *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, Grid2010, pages 49–56, 2010.

[20] Santiago Prieto, Alvaro Martínez, Roberto Afonso, Noé Gallego, José María Peribáñez, Toni Cortes, Ramon Nou, and Jacobo Giralt. Grid-aware applications for mobile phones (poster). In *Proceedings of the 5th European conference on Computer systems*, EuroSys '10, New York, NY, USA, 2010. ACM.

[21] Pierre Riteau, Adrien Lèbre, and Christine Morin. Handling persistent states in process checkpoint/restart mechanisms for HPC systems. In *Proc of IEEE Symposium on Cluster Computing and the Grid (CC-GRID 2009)*, Shangai, China, May 2009.

[22] Jan Sacha, Jeff Napper, Corina Stratan, and Guillaume Pierre. Adam2: Reliable distribution estimation in decentralised environments. In *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS)*, June 2010.

[23] XtreemOS consortium. Design and implementation of a basic customizable scheduler. XtreemOS Deliverable D2.2.6, November 2007.

[24] XtreemOS consortium. Fourth specification, design and architecture of the security and vo management services. XtreemOS Deliverable D3.5.13, December 2009.

[25] XtreemOS consortium. Prototype of the first advanced version of LinuxSSI. XtreemOS Deliverable D2.2.11, May 2010.