Project no. IST-033576

XtreemOS

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

Evaluation Report and Revision of Application Requirements D4.2.5

Due date of deliverable: September 30th, 2008
Actual submission date: November 13th, 2008

Start date of project: June 1st 2006

Type: Deliverable
WP number: WP4.2
Task number: T4.2.2, T4.2.4 and T4.2.5

Responsible institution: SAP
Editor & and editor's address: Philip Robinson
SAP Research, CEC Belfast
TEIC Building - University of Ulster
BT37 0QB Newtownabbey, U.K.

Version 2.0 / Last edited by Philip Robinson / November 10th, 2008

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	01/08/08	Philip Robinson	SAP	Adopted template and provided outline
0.2	18/08/08	Various WP4.2 Partners	ALL	Added demo for MDs
0.3	17/09/08	Marjan Sterk	XLAB	Added key experiments
0.4	17/10/08	Luis Pablo Prieto	TID	Added demo for MDs
1.0	24/10/08	Philip Robinson	SAP	Added installation experiments and finalised draft
2.0	10/11/08	Philip Robinson	SAP	Acknowledged reviewer comments

Reviewers:
Zhiwei Xu (ICT), Bryan Matthews (STFC)

Tasks related to this deliverable:

Task No.	Task description	Partners involved$^{\circ}$
T4.2.2	Revision of requirements and use case scenarios	BSC, EADS, EDF, SAP*, TID, UDUS, XLAB
T4.2.4	Implementing and porting applications to XtreemOS	BSC, EADS, EDF, SAP*, TID, UDUS, XLAB
T4.2.5	XtreemOS experiments and evaluation	BSC, EADS, EDF, SAP*, TID, UDUS, XLAB

$^{\circ}$This task list may not be equivalent to the list of partners contributing as authors to the deliverable

*Task leader

Executive Summary

The work package *Applications, Experiments and Evaluations* (WP4.2) contributes a range of reference applications which are used in two ways. Firstly, the application references are the basis for the definition and revision of the XtreemOS requirements. Secondly, the applications in WP4.2 are used to perform experimental evaluation of the available XtreemOS releases and intermediate XtreemOS components thereby providing regular feedback to the developers. In this case the evaluation and experimentation has been targeted towards providing feedback concerning the first, packaged release of XtreemOS. There are therefore no updated or additional requirements, as we focussed evaluation on existing features. We therefore focus more on the evaluation of the internal release of the packaged XtreemOS and describe the application-based test-beds that are being developed for further evaluation and demonstration.

The deliverable at hand further provides the evaluation of the first internal releases of various XtreemOS components including node-level VO support (from WP2.1), checkpoint and restart (from WP2.1), LinuxSSI (from WP2.2), SAGA API (from WP3.1), XtreemFS (from WP3.4, and CDA (from WP3.5). The evaluation is fully documented, comprising test plans, procedures for each test case, test logs and test results, ensuring traceability and repeatability. Furthermore, the fulfillment status of application requirements is given according to the evaluation results.

Contents

List of test documents

Chapter 1

Introduction

WP4.2 is in charge of defining and revising the requirements which are derived from a range of reference applications from different sectors. Furthermore, WP4.2 continuously conducts experimental evaluations of the XtreemOS components by means of executing reference applications using these components, thereby providing feedback to the developers.

In *deliverable D4.2.1[2]*, we presented the catalogue of initial requirements for XtreemOS. Various requirements had been updated and further requirements were introduced, driven by the ongoing activities in SP2 and SP3 and by better understanding of application behaviors under different conditions. The latest versions and the evolution of requirements are constantly communicated to the consortium for reasons of transparency and acceptance. For example, *Deliverable D4.2.2[1]* reported on the early experiments with the Kerrighed operating system which is the foundation for the cluster flavor of XtreemOS referred to as Linux-SSI. The development of Linux-SSI commenced after the state-of-the-art analysis and specification of XtreemOS during the second half of the first project year. This is now available in the XtreemOS install CD. *Deliverables D4.2.3[3] and D4.2.4[5]* provided updated requirements for XtreemOS, which have been adopted by the developers of the various components.

In this deliverable, D4.2.5, we have limited the inclusion of new requirements to coincide with the first release of the operating system. The existing requirements are only updated based on experience with the internal release of XtreemOS. Secondly, having now had some initial experience with the XtreemOS release, we proceed to enhance the details of application scenarios that now appear to be practically feasible. This led to a more comprehensive evaluation and demonstration of the capabilities of XtreemOS. The structure of the deliverable starts by providing the current set of requirements and their status. It then describes the results from various experimentation surrounding the requirements.

Chapter 2

Revision of Requirements

In close cooperation with the development work packages, we discussed and negotiated the application requirements previously published in deliverable D4.2.3 [3]. The requirements have been modified where appropriate to align them to the ongoing activities in SP2 and SP3 and to reflect the deeper insights into applications' and market needs.

There are currently no new requirements that have arisen in this period of testing, as we have decided to support a feature freeze leading up to the release of XtreemOS. We therefore commit more time in this deliverable to explaining the further evaluation and demonstration activities that we now see possible for XtreemOS. We provide a list of new and updated requirements together with explanations motivating these changes. The entirety of all application requirements also including the extended and updated ones are presented in Appendix A. In the following sections, new requirements are listed with the whole text. For updated requirements, we only give a reference to the respective pages in the Appendix.

2.1 New and Updated Requirements

Updated R109: The installation of XtreemOS must be intuitive, familiar and not radically deviate from standard OS installation procedures – see page 121 for new version of full text.

Updated R110: All services that can be independently started must be easily started without having to worry about the necessary dependencies – see page 121 for new version of full text.

Updated R111: The standard Linux system management utilities must be made available on XtreemOS and work in a backwards-compatible manner. This includes utilities like `top` used for checking memory – see page 121 for new version of full text.

Updated R112: The installation of applications designed and compiled for a Linux operating system should not have to undergo recompilation and extensive changing of binaries for the purpose of installation on XtreemOS – see page 122 for new version of full text.

2.2 Consolidated Requirements

As a new addition to this deliverable, a consolidated view of the requirements is provided. These general requirements do not replace the aforementioned requirements, but serve to specify assessment parameters for the overall viability of XtreemOS, given the various application domains in the work-package. These consolidated requirements consider XtreemOS as a future IT product and its potential contributions to the profitability, productivity and sustainability of an organisation including it in its computational infrastructure. These requirements are therefore towards assessing the potential for XtreemOS to be accepted and used in practice, granted that there are still various bugs and usability issues at the time of writing this report.

2.2.1 Profitability

Profitability of a product is a measure of how it facilitates earnings that outweigh expenses. That is, the advantages (financial or otherwise) are greater than the expenses required to acquire and maintain the product. In most organisations that depend on IT, there are four concerns that contribute to the determination of profitability: total cost of ownership, return on investment, leanness and training.

Total Cost of Ownership

The total cost of ownership (TCO) is a measure of the expenses incurred when purchasing and maintaining a product throughout its lifetime. Organisations are constantly trying to find ways of reducing their TCO in order to have a more profitable operation. Evidence of this can be seen in the recent rise of cloud computing as an alternative for large-scale deployments. This entails outsourcing the installation and management of infrastructure to a provider. XtreemOS may be relevant for both consumers of resources and producers. XtreemOS must not increase the cost of maintaining existing infrastructure. Providers will have to charge more to remain competitive. For this reason XtreemOS must be careful not to introduce unnecessary loads and resource demands. These must be driven by the applications rather than the operating system.

Return on Investment

Corresponding to TCO, the return on investment (ROI) describes the increased gains in performance or capability as a result of a product. The profitability of a product is therefore generally calculated as a difference $Profit = ROI - TCO$. Therefore, the advantages of XtreemOS over other existing operating systems, middleware and infrastructure frameworks with similar objectives must be made clear. The novel capabilities must be clear and traceable for an organisation installing XtreemOS as its underlying operating system.

Leanness

Leanness is a property of a product to be void of excessive and unnecessary functionality and resource usage. This highlights one of the key motivations behind following a rigorous requirements gathering and specification process. With the inclusion of explicit requirements it is possible to trace the necessity of each feature of XtreemOS.

Training

One the major contributors to spending is on educating personnel how to use new technology effectively. XtreemOS should hence avoid major need for organisations to have to re-train experts in relevant technology areas such as Linux and Grid systems management. If there is major retraining to be done in these areas, this suggests that there have been some oversights in the design and implementation of XtreemOS components, protocols and interfaces. In any event, the documentation for installation and troubleshooting of XtreemOS in various environments needs to be sufficient for organisations to quickly grasp its technical features and to become productive.

2.2.2 Productivity

Productivity and profitability share a close relationship. The more productive an organisation can be the more profits it can generally create, although this is still dependent on the product and market. Productivity refers to the ability of personnel in an organisation to work effectively with limited disruption to their planned procedures and transactions. There are then four contributors to productivity to which XtreemOS can contribute.

Reliability

Reliability is the assurance that a product will consistently perform its functions to produce expected results given a set of inputs over a certain period of time. Reliability is therefore with respect to an expected operational time and given operational conditions. Reliability refers to the correctness of implementation and conformity to a set of requirements. As XtreemOS moves from several, standalone components developed by different partners to a packaged solution, the unit reliability guarantees may be lost. For this reason we see it more important to shift our testing towards the packaged release of XtreemOS as opposed to individual services and components. Each component has a specification of what inputs it should accept and the expected outputs that follow. Even if some aspect of the system fails, it should also fail gracefully and provide the fault notification expected and included in the system's specification.

Availability

Availability compliments reliability, in that it is a property of a system's functions to be reachable and executable for a given time period. If a system is not available it cannot be reliable and, moreover, it cannot support productivity. Availability is also a security property of a system. The security mechanisms need to control and validate access to functions of the system, given the roles of the requester. One type of attack that has not been investigated in depth is the denial of service attack. This type of attack is a means of making functionality unavailable by usually overloading the system beyond its intended usage and scale.

Scalability

All of the reference applications expected to run on top of XtreemOS will have different workloads, number of users and requests in their production environments. XtreemOS hence needs to be tunable for different scales at installation but also support runtime scaling with limited disruption.

Manageability

Another aspect of productivity is the ability to easily install and modify the configuration of the operating system. This may be for the purpose of scalability, availability and reliability, but also for the purpose of application-specific customization.

2.2.3 Sustainability

Sustainability of an IT product refers to the ability of a product to contribute to the uptime and life-time of an organisation's computational infrastructure. This is with respect to the stability, recoverability and viability of the infrastructure, given the inclusion of the IT product. A fourth factor considers the amount of independence that [legacy] applications installed in the infrastructure continue to have given the inclusion of the IT product in the infrastructure.

Stability

Stability is a weaker measure of reliability. It means that a system should not produce unexpected errors given very "normal" conditions. This is a basic expectation of any IT product, although there may be some tolerance given its stage of development.

Recoverability

Even if a system or product has a low stability, its reliability requirement can still be achieved. Recoverability is the property of a system that allows it to be reliable and available despite being unstable under certain conditions and within certain time periods. Recoverability is based on the contingency planning implemented within the software, as well as the manageability that it provides for quick response by its users.

Viability

Viability of an IT product and system is with respect to the cost of ownership (TCO - discussed above), but also with respect to the support for managing updates. Should the TCO and support effort become too expensive such that the computational infrastructure is often down and burdensome, the viability of the product and infrastructure decrease. XtreemOS needs to be designed and packaged in such a way that building a support framework is possible and evolvable.

Flexibility and Independence

XtreemOS should not force applications to be completely engineered and designed purely for its architecture and services. This is often a reason why IT products are rejected but also impacts on the lifetime of the overall computational infrastructure. If the organisation cannot have the freedom to port its applications into different execution environments, or to easily transfer legacy applications into new environments, then the uptime and lifetime of the computational infrastructure will be affected.

As XtreemOS becomes more mature in its packaging and demonstration, we will increasingly move away from evaluating the specific, detailed, technical requirements listed and discussed in this deliverable, towards the more consolidated view. This will aid in quicker evaluation cycles and a means of presenting XtreemOS from a value perspective. This will facilitate better cooperation between WP4.2 and the exploitation work-package WP5.1.

Chapter 3

Evaluation Report

In this chapter, we report on the setup, the procedure and the results of the experimental evaluation carried out by WP4.2. Section 3.1 positions the evaluations into the context of other test efforts within the consortium, an overview of the available XtreemOS components and their assignment to the various reference applications is given together with a general introduction into the test procedure and documentation. The actual evaluation of the available XtreemOS components is described in sections 3.2 to 3.7 including the test plans, test specifications, logs and test results. These results are summarized in Section 3.8 which also gives an overview of the fulfillment status of all requirements defined by WP4.2.

3.1 Overview

Testing of XtreemOS is carried out at the level of development, packaging and applications. This section provides an overview of the applications that we continue to use for evaluation purposes of XtreemOS. There are 6 stages involved in this form of application-driven evaluation:

1. **Requirements analysis:** this involves the specification, justification and classification of various technical requirements.

2. **Concept development and evaluation framework**: it is expected that XtreemOS enable new features for applications or enhance the way in which existing features are executed. This stage is therefore designing the new concept that is intended for the application to be enabled by XtreemOS.

3. **Technology trials**: this includes installing and learning different features of XtreemOS that are relevant to the particular application concept.

4. **Technology-specific concept development**: as most of the applications are already developed, this stage entails tailoring the application to installation and execution on top of XtreemOS. One of the key challenges we try to

avoid is significant changes in the application in order to be executional on top of XtreemOS. It is however not possible to avoid changes fully for more complex applications.

5. **Application execution testing**: this stage is now evaluating XtreemOS from the perspective of observing the execution of the application. This is a different form of testing as it does not involve explicit, individual execution of XtreemOS commands. It involves installing, running and performing operations on the application and its data to observe the way in which XtreemOS supports these operations.

6. **Collation of results**: once the different applications have been tested, the observations are then collected and collated in order to have a unified picture of XtreemOS's status.

The selected body of applications still remains unchanged and are listed in Table 3.1. There are 14 applications that are being provided and tested by 8 different partners and hence local XtreemOS testbeds. At the time of doing the experiments there was also an external testbed being developed, but most partners have selected to go the route of local installation for the purpose of limiting access to licensed software. A set of publicly accessible applications are still to be determined and included. Guidelines for building a local testbed have been internally published and are hence tested as part of the evaluation process.

Partner	**Application Name**	**Short Name**	**Application Area**
BSC	Specweb2005	SPECWEB	Enterprise solutions
BSC	GRID superscalar fastDNAml	GSDNA	Bio-informatics
EADS	Elfipole	ELFIPOLE	Electromagnetics
EADS	jCAE	JCAE	Computer aided engineering
EDF	Moderato	MODERATO	Particle physics
EDF	Simeon	SIMEON	Optimization
EDF	Zephyr	ZEPHYR	Fluid mechanics
EDF	Secured Remote Computing	SRC	Enterprise solutions
SAP	SAP Netweaver Application Server	WEBAS	Enterprise solutions
T6	DBE	DBE	Enterprise solutions
TID	TID Instant Messaging Application	IMA	Instant messaging
TID	Job Management Application	JOBMA	XtreemOS job management
UDUS	Wissenheim	WISS	Virtual Presence
XLAB	Galeb	GALEB	Economics, optimization

Table 3.1: Overview of the applications in WP4.2.

In addition to the internal installation documentation, a repository of software to be installed is being maintained. From this repository it is easier for non-developers to follow the progress made in developing, availability and packaging the operating system. Table 3.1 provides an overview of the XtreemOS components and their availability at the time of writing the deliverable, based on what is reported in the internal repository. Note that the 1st official release was not yet done at the point of writing, but there was sufficient internal activity to suggest that this would be completed by the time this document was reviewed. In the last deliverable D4.2.4[5], the components were identified at a higher-level granularity, given the status of packaging. In this deliverable, a finer-grained listing of components and their availabilities is provided, based on their status reported in our internal repository.

There are 21 features that have been identified as necessary to satisfy the requirements. Some of these features refer to one component, but some cover more than one and in some cases there are overlaps. In the previous deliverable they were referred to as components but features is actually a more technically-correct term. In the last deliverable D4.2.4[5], only 6 of the features were listed as being available, albeit in alpha state. At this point in time there are now X available and y partially available. For more details on each component and feature, refer to the rel;evant documentation and deliverables from the corresponding work-package (WP), as indicated in Table 3.2.

WP	Component Name	Readiness
	Packaged release of XtreemOS	yes
2.1	Node-level VO support	yes
2.1	Checkpoint and restart	yes
2.2	Linux-SSI [prev. Kerighed] (32 bit version)	yes
3.1	SAGA API (C++ engine)	yes
3.2	Publish/subscribe service	partial
3.2	Node management	yes
3.2	Directory service	yes
3.2	Virtual nodes	no
3.2	Distributed servers	no
3.3	Scheduler	no
3.3	Monitoring and accounting	no
3.3	Job submission	partial
3.3	Interaction with jobs	no
3.3	Checkpoint, restart, migration (Grid-level)	no
3.3	Job self-management	no
3.4	XtreemFS	yes
3.5	Credential Distribution Agency (CDA)	yes
3.5	Virtual Organization Policy Service (VOPS)	yes
3.5	Identity Service	no
3.5	Attribute Service	no
3.5	Virtual Organization Membership Service	yes

Table 3.2: Overview of features and their readiness for evaluations by WP4.2, based on internal reports

The availability of features is judged based on those included in the packaging of the 1st official release. Some of the highlights of the packaging are discussed in following; *Node-level VO support* was marked as available in the previous deliverable and is also included in the first release. Checkpointing and restart mechanisms are also included based on the Berkeley Lab Checkpoint/Restart (BLCR) architecture. The 32-bit version of Kerrighed is now modified and repackaged as Linux-SSI (Single System Image). Finally, the introduction of the VO-Life demonstration application also provides a means of observing multiple components interact for the purpose of more general VO (Virtual Organization) life-cycle management.

As in the previous deliverable, table 3.3 gives an overview of the tested XtreemOS components along with the applications used for evaluation.

WP	Component Name	SPECWEB	GSDNA	ELFIPOLE	JCAE	ZEPHYR	WEBAS/MaxDB[1]	DBE	WISS	GALEB	other
2.1	Node-level VO support				X					X	
2.1	Checkpoint and restart	X					X	X			
2.2	LinuxSSI (Kerrighed)			X						X	
3.1	SAGA API					X					
3.4	XtreemFS		X				X		X		
3.5	Credential Distribution Agency (CDA)										X
4.1	Credential Distribution Agency (CDA)										X

Table 3.3: Overview of available components and their assignment to applications in WP4.2

The following sections present the test documentation. As in the proceeding deliverable (D4.2.4), this test documentation is based on the "IEEE Standard for Software Test Documentation", IEEE 829-1998 [8]. Accordingly, the documents in these sections cover the phases test planning, test specification, and test reporting.

Among others the *test plan* describes the scope, approach, resources, the items and the features to be tested and the testing tasks to be performed. Per XtreemOS component, one test plan is provided covering the test setup for all WP4.2 applications evaluating this component.

The test specification consists of three document types:

- The *test design specification* gives more details on the test approach and presents the features covered by the design and its respective tests. Furthermore the associated test cases and test procedures are identified, and the feature pass/fail criteria are specified. Usually, for each application testing a certain XtreemOS component, a separate test design is provided.

- The *test case specification* provides the actual input values used as well as the anticipated outputs. Moreover, the test case describes the constraints on the test procedures.

- The *test procedure specification* explains all steps needed to operate the test system.

The test reporting consists of three document types :

- The *test log* records events, input and output during test execution. We use these documents to record events which do not require further investigation.

- The *test incident report* records all events during the test execution which need further examination.

- The *test summary report* summarizes the tests. For each XtreemOS component, we give a common summary for all applications testing it.

Note that we do not use the *test item transmittal report* proposed by the standard. Details on the releases tested along with the information on development groups and test groups are provided within the other test documents.

According to the suggestions of IEEE 829-1998, each test document has an identifier which allows for a unique reference to tests also across the subsequent deliverables of WP4.2. The following naming convention is applied for the identifiers:

- wpXX-CC-tsYY-NAME for the test plan and the test summary report

- wpXX-CC-tsYY-APPZZ-NAMENN for all other test documents

Explanation:

- XX = work package number

- CC = short XtreemOS component name (2 or 3 letters)

- YY = enumerator for test sequence. Test sequences describe the self-contained setup and activities for testing a certain component. A new test sequence may be created, e.g., in the case of new software releases or for new test setups.

- APP = short application name

- ZZ = test case number
- NN = running number for document
- NAME = type of document. Here we use the following abbreviations:
  - tp = test plan
  - tds = test design specification
  - tcs = test case specification
  - tps = test procedure specification
  - tl = test log
  - tir = test incident report
  - tsr = test summary report

Example: Node-level VO support from WP2.1 tested in test sequence 2 by application GALEB.

Document type	Identifier
test plan	wp21-vo-ts02-tp
test design specification 1	wp21-vo-ts02-galeb01-tds01
test case specification 1	wp21-vo-ts02-galeb01-tcs01
test procedure specification 1	wp21-vo-ts02-galeb01-tps01
test log 1	wp21-vo-ts02-galeb01-tl01
test incident report 1	wp21-vo-ts02-galeb01-tir01
test summary report	wp21-vo-ts02-tsr

3.2 Evaluation of Node-level VO Support

The node-level VO support component from WP 2.1 (short name: wp21-vo) provides a bridge between local accounts on nodes and identities of global grid users that are members of various VOs.

3.2.1 Test plan – VO

Test plan identifier: wp21-vo-ts02-tp

Introduction

This test plan covers the software for node-level VO support, which includes:

- verification of grid user's tokens for accessing a grid node,
- mapping global user identities to local user identities and group identities,

- translating VO-level access control policies into local OS-level access rights and capabilities,
- session management.

The purpose, architecture and use cases are described in the XtreemOS deliverable D2.1.2 [4].

As this software is not intended to be used directly by the users or applications, we test it through the application execution manager (AEM). For example, submitting a `whoami` job reveals what local account the global user identity is mapped to.

Although the same features were already tested for the XtreemOS deliverable D4.2.4, they were done using `xos-ssh` because the AEM was not yet available. Furthermore, the configuration interface of the account mapping service changed significantly. The test specifications, input, and procedures are thus significantly different from those described in D4.2.4.

Test Items

The tests will be done on the version of the software included in the XtreemOS release packaged on 2008-08-26. The contact person for the node-level VO support software is Haiyan Yu (`yuhaiyan@ict.ac.cn`). The contact person for the AEM, which is also used in the tests, is Matej Artač (`matej.artac@xlab.si`).

Features to be Tested

The tested feature will be the mapping of VO users to groups and accounts on the VO-aware node. Since the local account used implies access rights, capabilities, and isolation between users, these features are also implied by the mapping.

Features not to be Tested

At this stage, the following features will not be tested:

- VO roles and role mapping,
- quota specification and enforcement,
- performance,
- security.

These features will be tested on later versions of the software. In particular, security will be tested later because most XtreemOS components currently do not use SSL encryption. Furthermore, any invalid certificate (either expired or signed by a non-trusted CA) will be rejected by the AEM, before it even gets to the node's local account mapping service.

Approach

The purpose of these tests is to evaluate the current version of software, provide feedback to developers, and check which requirements are fulfilled, which are partially fulfilled, and which remain to be done. We will thus focus on evaluating the higher-level design, features and usability of each module rather than bugs in the implementation. Correspondingly, the test plan, item pass criteria etc are not given in too much detail. The test plan can also be adapted during testing to the current state of the modules and to their usage philosophy.

To ensure repeatability and comparability of tests among partners, all tests will be done on the packaged XtreemOS release from 2008-08-26, using stable versions of any required third-party libraries. The test documentation should include a description of the installation procedure and in particular detailed descriptions of any deviations that were made from the installation instructions supplied with the software and of steps that are not described adequately in the instructions. Such documentation will help the developers to improve installation instructions as well speed up the installation process for other partners.

Item Pass Criteria

Mapping of VO users will pass the test if users (identified by their XtreemOS certificates) are correctly mapped to either existing local accounts or accounts created on demand for each VO user, as specified in the configuration files of each test case. Users belonging to different VOs must must be mapped to different local groups. Users belonging to the same VO must be mapped to the same local group.

Testing Tasks

The test preparation requires the following tasks:

1. initial configuration of any other required software and/or third-party libraries,

2. preparing the testing environment, e.g., making any changes to the test bed configuration,

3. obtaining VO user certificates.

Each test then consists of adapting the configuration to the specific test, performing the test, and documenting the full procedure. The individual tests can be executed in any order, but the order of tasks during preparation and during execution of tests must be as given here.

Environmental Needs

The required resources include two computers. The installation and testing might interfere with normal usage of the nodes so it is recommended to use dedicated computers or virtual machines. There are no specific hardware requirements.

The machines must run the packaged XtreemOS release from 2008-08-26.

Responsibilities

This test plan and the included tests with the Galeb application are the responsibility of XLAB. The included tests with the jCAE applications are the responsibility of EADS.

Schedule

Being that the software is in development stage, the installation, setup and learning are complex processes, requiring one or two weeks if the staff is not familiar with the procedure. However, after the preparation is finished, the tests are simple and will be done in a few days.

All the planned tests should be finished in time for XtreemOS deliverable D4.2.5, which is due on 2008-09-30.

Risks and Contingencies

Apart from delays in testing caused by late delivery of new versions of the software and by unexpected installation problems, no specific risks are envisaged. Note that late delivery of new versions of the software can be a consequence of bugs discovered by these tests.

3.2.2 Test design specification

Test design specification identifier: wp21-vo-ts02-galeb01-tds01
Test plan reference: wp21-vo-ts02-tp

Features to be Tested

The subject of this test design specification is the mapping of VO users to groups and accounts on the VO-aware node (test cases wp21-vo-ts02-galeb01-tcs01 and **??**, testing requirements R89, R22, R24, R26, R28, R96). This mapping implies certain access rights and isolation between users belonging to different VOs and between those belonging to the same VO, thus also testing requirements R25, R85, R28, R86, R96).

To summarize, the fulfillment of the following requirements from D4.2.4 will be evaluated:

- full test: R25, R85, R89,
- partial test: R22, R24, R26, R28, R86, R96.

Approach Refinements

This test design covers very basic tests, which can be done either using the Galeb application, where the application spawns jobs to other nodes and accesses files on those nodes, but also using the command-line utilities like `whoami` and `id`.

Test Identification

The test case that is the subject of this test design is wp21-vo-ts02-galeb01-tcs01.

Feature Pass/Fail Criteria

The test will pass if the user is mapped according to her distinguished name and VO stored in the certificate. No password should be required. The feature fails if:

- an unauthorized user manages to log into the node,
- an authorized user manages to log into the node, but is mapped incorrectly,
- an authorized user is denied access.

3.2.3 Test case specification – VO mapping Galeb

Test case specification identifier: wp21-vo-ts02-galeb01-tcs01

Test Items

This test case tests mapping of VO users to accounts on the VO-aware node.

Input Specifications

The input of this test case consists of the rules to be used to map VO users to accounts on the node and by the user's distinguished name (DN) and VO name present in her user's certificate.

The following cases will be tested:

1. mapping to an existing local account,
2. mapping to a dynamically created account.

Each case requires one of the following command to input the mapping rule into the account mapping database:

```
xos-policy-admin-am -dn $USER_GUID -vo $VO_UNIQUE_KEY -locnam
    (cont.)$LOCAL_USER -drvname root -drvparam 2510
xos-policy-admin-am -dn \* -vo $VO_UNIQUE_KEY -locnam \* -drvname
    (cont.)root -drvparam 2510
```

The first rule says that the user with the globally unique ID $USER_GUID from VO $VO_UNIQUE_KEY should be mapped to the local account $LOCAL_USER. The second rule says that all users from VO $VO_UNIQUE_KEY should be mapped to dynamically created accounts.

The corresponding group mappings must be set up with one of the following commands, respectively:

```
xos−policy−admin−gm −grp \∗ −vo $VO_UNIQUE_KEY −locgrp
    (cont.)$LOCAL_GROUP
xos−policy−admin−gm −grp \∗ −vo $VO_UNIQUE_KEY −locgrp \∗
```

Each case must be tested with the VO given in the mapping rule (in which case the command must succeed) and with another VO (which should fail).

Output Specifications

The command submitted from a remote node through AEM must run under the account specified in the mapping rule. This is most conveniently checked by submitting the command `id`, which outputs the local username and uid as well as local group name and gid. An example output of the command `id` is:

```
[marjan@xostest1 ~]$ id
uid=502(marjan) gid=502(marjan) groups=10(wheel),107(fuse),502(
    (cont.)marjan)
```

3.2.4 Test procedure specification

Test procedure specification identifier: wp21-vo-ts02-galeb01-tps01
Test design specification reference: wp21-vo-ts02-galeb01-tds01
Test case specification reference: wp21-vo-ts02-galeb01-tcs01
Test log identifier: wp21-vo-ts02-galeb01-tl01

Purpose

This procedure executes the test case wp21-vo-ts02-galeb01-tcs01.

Procedure Steps

Configuration

1. XtreemOS must be installed and configured on the two test nodes so that job submission is possible.

2. Two distinct VOs must be created. One user should be a member of both while another user should only be a member of the first VO.

Set Up

1. Three user certificates must be created using the VOlife web interface (one for each VO membership, i.e. two for the first user and one for the second user).

2. For tests with the first user, the appropriate certificate must be selected (i.e. either copied to the default user certificate location `~/.xos/truststore/certs/cda.pem`, or the file `~/.xos/XATIConfig.conf` must be edited accordingly).

3. The mapping rule(s) must be prepared according to the test case specification. The `$USER_GUID` and `$VO_UNIQUE_KEY` to be used in the mapping rules can be extracted from the user certificate using the command `openssl x509 -in CERT_FILE -text`. `$USER_GUID` is given in the Subject field of the certificate and the `$VO_UNIQUE_KEY` in the extension field 1.34.5.0.14.1.

Start The commmand `id` is run by typing the command `xsub -f JSDL_FILE`, where the `JSDL_FILE` contains the following job description:

```
<?xml version="1.0" encoding="UTF-8"?>
 <JobDefinition xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/
    (cont.)jsdl">
    <JobDescription>
        <JobIdentification>
              <Description>Execution of whoami</Description>
              <JobProject>Testing of XtreemOS account mapping</
                 (cont.)JobProject>
          </JobIdentification>
          <Application>
              <POSIXApplication xmlns:ns1="http://schemas.ggf.org/
                 (cont.)jsdl/2005/11/jsdl-posix">
                   <Executable>/bin/id</Executable>
                   <Output>/tmp/id.out</Output>
                   <Error>/tmp/id.err</Error>
              </POSIXApplication>
          </Application>
      </JobDescription>
 </JobDefinition>
```

This job description tells the AEM to run `/bin/id` on the remote node, redirecting the standard output and standard error to `/tmp/id.out` and `/tmp/id.err`, respectively. Alternatively, files on a mounted XtreemFS volume can be used for output redirection.

Wrap Up Between successive tests with different rules, we must delete both the mapping database (which is a sort of 'cache' of mappings for the users that had already started at least on session and have thus been mapped) as well as the rules themselves (which are used when if the user is not found in the mapping database). Note that this 'cache' of existing mappings is necessary in order to assign the same UID for subsequent sessions belonging to the same grid user.

The following commands can be used:

```
xos-policy-admin-cleardb
/etc/init.d/xos-amsd stop
rm -f /etc/xos/nss_pam/mapdata/*
/etc/init.d/xos-amsd start
```

We decided to delete the contents of the `mapdata` directory each time because we did not find a command-line utility to delete all existing rules. We suggest to the developers to add a switch for deleting all rules to the `xos-policy-admin-del` command.

Contingencies The anomalies should be dealt with by resetting all the mappings as described in the Wrap Up paragraph.

3.2.5 Test log

Test log identifier: wp21-vo-ts02-galeb01-tl01

Description

The test procedure wp21-vo-ts02-galeb01-tps01 was executed by Marjan Šterk, XLAB. The nodes used were `xostest1` and `xostest2` from the XtreemOS consortium permanent testbed. Both machines initially had an older XtreemOS release installed but all packages were subsequently upgraded to the same versions as those on the 2008-08-26 release.

Installation and Setup

Execution Description The two test nodes were also used for the demonstration at the July's project review, thus they were installed and configured in July 2008. As mentioned above, they were subsequently upgraded to the 2008-08-26 release version.

Additionally, the VO-level policies were setup so that any job submission is granted on the VO level. The node-level VO service is thus the only access decition point.

The two VOs 'DemoVO' and 'secondVO' were then created with the VOlife web interface. Two grid users, 'user01' and 'user02' were created, corresponding to local user accounts of the same name on one of the test nodes. The first is a member of both VOs while the second one is only a member of DemoVO. The user certificates, with which the users proves the VO membership, were also issued with the VOlife web interface. For user02 the (only) certificate was stored as $HOME_DIR/.xos/truststore/certs/cda.pem. User01 has two certificates, one for each VO. She thus has to store the corresponding one to the above location before each test or, alternatively, change the corresponding setting in $HOME_DIR/.xos/XATIConfig.conf.

Procedure Results After successful setup job submission is possible.

Anomalous Events When the account mapping service's data is deleted and the service is reset, the rule database is not empty. Instead it contains a default set of rules:

```
[root@xostest1 ~]# xos-policy-admin-prt
-------------- rule database ------------
[key:1]: LlUspH7raQ5PlWkwY4VwjJvsieJIZRk
Rule Type:Mapping rule
        Subject:
                Type: <DN,VO,ROLE>
                Content_1: *
                Content_2: xtreemos
        Object:
                Type: local account
                Content:*
        Rule Content:
                Driver Name:
                Driver Params:
        Valid: 1
        TimeStamp: 1234567

[key:2]: MTbfhBWkawwXRMzEnmIANaDcq12EWuW
Rule Type:Mapping rule
        Subject:
                Type: <VO,ROLE,ATTRS>
                Content_1: xtreemos
                Content_2: *
        Object:
                Type: local groups
                Content:*
        Rule Content:
                Driver Name:
                Driver Params:
        Valid: 1
        TimeStamp: 1234567

[key:3]: hnsEEacG2PVnnYDx9msXzF7BxraBdsW
Rule Type:Resource usage
        Subject:
                Type: <DN,VO,ROLE>
                Content_1: -xtreemos-oscar
                Content_2: xtreemos
        Object:
                Type: quota object
                Content:NOFILE
        Rule Content:
                quota: [ABS|5.000000]
        Valid: 1
        TimeStamp: 1234567

[key:4]: q00dUbdA2pbFF8CEIDYhP9M4WYu76bX
Rule Type:Resource usage
        Subject:
                Type: <DN,VO,ROLE>
                Content_1: -xtreemos-shu
                Content_2: xtreemos
        Object:
                Type: quota object
                Content:NPROC
        Rule Content:
                quota: [ABS|4.000000]
        Valid: 1
        TimeStamp: 1234567

[key:5]: 349NKjiNMsWtczMIpSgEXyK1k0XWpEg
Rule Type:Resource usage
        Subject:
                Type: <DN,VO,ROLE>
                Content_1: -xtreemos-haiyan
                Content_2: xtreemos
        Object:
                Type: quota object
                Content:CPU
        Rule Content:
                quota: [ABS|4.000000]
        Valid: 1
        TimeStamp: 1234567

[key:6]: CeO8m3m0vCKxYZFmJrLHSHrh4suWfPY
Rule Type:Resource usage
        Subject:
                Type: <DN,VO,ROLE>
                Content_1: -xtreemos-yvon
                Content_2: xtreemos
        Object:
                Type: quota object
                Content:MEMORY
        Rule Content:
                quota: [ABS|6144.000000]
        Valid: 1
        TimeStamp: 1234567

[key:7]: u2yRQaN1NnCOUKZlDAkMGKDAwnl2as4
Rule Type:Access Control rule
        Subject:
                Type: <DN,VO,ROLE>
                Content_1: *
                Content_2: chemVO
        Object:
                Type: VO node object
                Content:127.0.0.1
        Rule Content:
                Permission: 0
        Valid: 1
        TimeStamp: 1234567

[key:-1]: Yql9N0hNHG5Od0TajvJNOLVBTD2AeN9
Rule Type:Mapping rule
        Subject:
                Type: <VO,ROLE,ATTRS>
                Content_1: 60500
                Content_2: 60000
        Object:
                Type: local groups
                Content:*
        Rule Content:
                Driver Name: 60500
                Driver Params: 60000
        Valid: 1
        TimeStamp: 1234567
```

Note that creating such default rule set is an appropriate way for the software to handle manual deletion of the rule set that we do before each test. This anomaly is merely an additional argument to introduce a command-line way of deleting all rules.

Running the Test Procedure

Execution Description The procedure was run on 2008-09-03 following the procedure specification wp21-vo-ts02-galeb01-tps01.

Procedure Results Table 3.4 summarizes the local accout to which the VO user is mapped. The local account to map to can be either an existing local account ('user01', 'root' etc) or a dynamically created account with the user name equal to the certificate subject (mapping rule *). The cases where mapping is incorrect are shown in **bold**.

Let us list the output created when executing the first entry of the table. The console from which the job is submitted (as expected):

```
[user01@xostest1 ~]$ xsub -f id.jsdl
Job submitted succesfully: d69d3e64-1305-4359-915d-54732cbf98c4
```

The console from which AMS daemon was started (too many warnings and errors reported):

```
xos_dbaux_grp.c:397: No records found for the token
xos_dbaux_pwd.c:528: No records found for the token
xos_policy_mgmt.c:129: subject type unmatch !
xos_policy_mgmt.c:129: subject type unmatch !
```

Table 3.4: Mapping to local accounts with incorrect mapping shown in **bold**.

mapping rule			contents of user certificate		actually mapped to	
VO ID	GUID (DN)	map to	VO ID	GUID (DN)	user	group
DemoVO	*	*	DemoVO	user01	60000(user01 ID)	60210(xosuser_g60210)
DemoVO	*	*	DemoVO	user02	60238(user02 ID)	60210(xosuser_g60210)
DemoVO	*	*	secondVO	user01	access denied	
DemoVO	user01	*	DemoVO	user01	60000(user01 ID)	60063(xosuser_g60063)
DemoVO	user01	*	DemoVO	user02	access denied	
DemoVO	user01	*	secondVO	user01	access denied	
DemoVO	user01	user01	DemoVO	user01	**0(root)**	60370(xosuser_g60370)
DemoVO	user01	user01	DemoVO	user02	access denied	
DemoVO	user01	user01	secondVO	user01	access denied	
no applicable rule			access denied (in all cases)			

```
xos_dbaux_pwd.c:528: No records found for the token
xos_dbaux_pwd.c:701: Not found the record
xos_dbaux_pwd.c:528: No records found for the token
xos_dbaux_grp.c:397: No records found for the token
xos_dbaux_grp.c:564: Not found the record
xos_dbaux_grp.c:397: No records found for the token
xos_dbaux_pwd.c:420: No records found for 'xosuser_u60000'
xos_policy_mgmt.c:129: subject type unmatch !
xos_policy_mgmt.c:129: subject type unmatch !
xos_policy_mgmt.c:147: Warning:unmatch '/CN=710ad39b-05e3-4c31-89
    (cont.)f8-52cc6922c52a'
xos_policy_mgmt.c:147: Warning:unmatch '/CN=710ad39b-05e3-4c31-89
    (cont.)f8-52cc6922c52a'
xos_policy_mgmt.c:147: Warning:unmatch '/CN=710ad39b-05e3-4c31-89
    (cont.)f8-52cc6922c52a'
xos_policy_mgmt.c:147: Warning:unmatch '/CN=710ad39b-05e3-4c31-89
    (cont.)f8-52cc6922c52a'
xos_policy_mgmt.c:164: unmatch 'abc15788-976a-4b8b-a849-3
    (cont.)e6621c3aff5'
xos_policy_mgmt.c:277: Warning:Can not deal with the rule
xos_policy_mgmt.c:285: Warning: Can not apply rule with handler '
    (cont.)ln9bJhttDX5vnSD0ukEfdH9Js4y7m3U'
xos_dbaux_pwd.c:522: bdb_getdbgtk_pwd: role is not same
xos_dbaux_grp.c:397: No records found for the token
```

The stdout file of the submitted job (as expected, username is the same as the DN in the certificate):

```
uid=60000(/CN=710ad39b-05e3-4c31-89f8-52cc6922c52a) gid=60210(
    (cont.)xosuser_g60210) groups=60210(xosuser_g60210)
```

And finally in the stderr file of the submitted job (too much output):

```
xos_protocol.c:286: AMS_Client:Sending message '8##60000###' ...
xos_protocol.c:292: AMS_Client:sent message.. wait for response...
xos_protocol.c:286: AMS_Client:Sending message '$quit$' ...
xos_protocol.c:292: AMS_Client:sent message.. wait for response...
xos_protocol.c:286: AMS_Client:Sending message '8####60210#' ...
xos_protocol.c:292: AMS_Client:sent message.. wait for response...
xos_protocol.c:286: AMS_Client:Sending message '$quit$' ...
xos_protocol.c:292: AMS_Client:sent message.. wait for response...
xos_protocol.c:286: AMS_Client:Sending message '8####60210#' ...
xos_protocol.c:292: AMS_Client:sent message.. wait for response...
xos_protocol.c:286: AMS_Client:Sending message '$quit$' ...
xos_protocol.c:292: AMS_Client:sent message.. wait for response...
```

Anomalous Events

As given in Table 3.4, there was one case when user01 from DemoVO should have been mapped to local account user01 but was instead mapped to root. Detailed analysis is given in the incident report wp21-vo-ts02-galeb01-tir01.

Another problem is that running two successive commands `xos-policy-admin-am` and `xos-policy-admin-gm`, which should produce an account mapping rule and a group mapping rule, sometimes produces just one of those. A workaround that works usually, but not always, is to wait a few seconds between the two commands.

3.2.6 Test incident report – VO mapping to root

Test incident report identifier: wp21-vo-ts02-galeb01-tir01
Test log reference: wp21-vo-ts02-galeb01-tl01

Summary

There was one case when user01 from DemoVO should have been mapped to local account user01 but was instead mapped to root.

Incident Description

During testing on 2008-09-04, the software behaved as follows. When access should be denied, it was. When the mapping rule was to map the user to an automatically generated account, the mapping was correct. However, if the rule was to map the user to an existing local account, she was instead mapped to root.

The output of the `id` command was:

```
uid=0(root) gid=60370(xosuser_g60370) groups=60370(xosuser_g60370)
```

The output in the console from which the AMS daemon was started was:

```
xos_policy_mgmt.c:164: unmatch '/VO=abc15788-976a-4b8b-a849-3
    (cont.)e6621c3aff5/ROLE=abc15788-976a-4b8b-a849-3e6621c3aff5'
xos_dbaux_grp.c:397: No records found for the token
xos_dbaux_pwd.c:528: No records found for the token
xos_policy_mgmt.c:129: subject type unmatch !
xos_policy_mgmt.c:129: subject type unmatch !
```

```
map_mech.c:256: account mapping: one−by−one for mapping to exist
   (cont.)user
map_mech.c:357: accountmapping:Mapping rule is wrong
xos_dbaux_pwd.c:528: No records found for the token
xos_dbaux_pwd.c:701: Not found the record
xos_dbaux_pwd.c:528: No records found for the token
xos_dbaux_grp.c:397: No records found for the token
xos_dbaux_pwd.c:522: bdb_getdbgtk_pwd: role is not same
xos_dbaux_grp.c:397: No records found for the token
```

Impact

The bug represents a fatal security hole in the component. The developers have been notified about the problem and are working on it. Other tests are not affected by this incident.

3.2.7 Test summary report – VO

Test summary report identifier: wp21-vo-ts02-tsr

Summary

FiXme: *TODO: update summary after JCAE tests*

The testing of the node-level VO support (as packaged in XtreemOS released on 2008-08-26) was evaluated using

- simple command-line tests in test case wp21-vo-ts02-galeb01-tcs01,
- the jCAE application in test case **??**.

The first test case focused on account mapping rules. The second used the functionality tested by the first two in a real application use-case.

Comprehensiveness Assessment

The testing process was in line with the approach given in Section wp21-vo-ts02-tp.

Summary of Results

While most tests executed successfully, there were 3 unresolved incidents, given in the incident reports wp21-vo-ts02-galeb01-tir01,, **??**, and **??**. Particularly the first one represents a fatal security hole and must be corrected as soon as possible.

Evaluation

Apart from the bugs that caused the incidents identified above, the software appears from the user perspective to be well thought-out. As expected, not all requirements are satisfied in this early version:

R22: fulfilled,

R24: fulfilled (though mostly by VOMS, the AMS has little to do with it), further testing needed (e.g. VO-level policies),

R25: partly fulfilled (will be fulfilled when the incidents identified above are resolved),

R26: not fulfilled (changing the VO can cause VO user's files to become inaccessible; could be fulfilled by other components),

R28: fulfilled,

R85: partly fulfilled (testing of group-level access and access to other objects required to assess complete fulfillment),

R86: fulfilled,

R89: fulfilled,

R96: fulfilled.

The requirements satisfaction is higher than in previous versions of the software.

It should be noted that the tested version of the software produces many debug messages, which are also visible to unauthorized users and could be exploited by the potential attacker.

3.3 Evaluation of Federation Management (LinuxSSI)

3.3.1 Test design specification

Test design specification identifier: wp22-fm-ts02-galeb01-tds01
Test plan reference: ??

Features to be Tested

The following features are the subject of this test design specification:

1. migration of individual processes (test case wp22-fm-ts02-galeb01-tcs01, testing requirements R6),

2. checkpointing of applications (test case wp22-fm-ts02-galeb01-tcs01, testing requirements R47, R32, R39, R33).

The fulfillment of the following requirements from D4.2.3 will thus be evaluated:

- full test: R6,
- partial test: R47, R32, R39,

Approach Refinements

Since the tests covered by **??** include multiple applications of various complexity, tests covered by this test design will only use the Galeb application. Galeb, as used in this test design, is a command-line application that is a text file containing the input data (a sampled function). The master process starts N slave processes, which then start the computation. Each slave process sends the results (an algebraic expression that approximates the input data, obtained by a genetic algorithm, plus a measure of error) to the master using SysV message queues. The master process simply selects the best one and outputs it.

Test Identification

Test case wp22-fm-ts02-galeb01-tcs01 (checkpointing and migration) will be performed.

Feature Pass/Fail Criteria

Feature 2 will pass if a process is successfully checkpointed (with the checkpoint stored to a file) and restarted, as well as being successfully migrated to another SSI node (without explicitly storing the checkpoint into a file). The migration must not break the on-going IPC.

3.3.2 Test case specification – FM checkpoint-migrate process

Test case specification identifier: wp22-fm-ts02-galeb01-tcs01

Test Items

This test case tests checkpointing, restart and migration of individual processes.

Reference to previously executed tests

This test case is the same as the corresponding test reported in XtreemOS deliverable D4.2.4. Only the test logs and potential incident reports are thus given here.

3.3.3 Test log

Test log identifier: wp22-fm-ts02-galeb01-tl01

Description

The test case wp22-fm-ts02-galeb01-tcs01 was executed by Marjan Šterk, XLAB. The tests were done on a two-node cluster of 32-bit single-core virtual machines running under VMware Server. Both nodes had XtreemOS release 2008-08-26 installed, with LinuxSSI extensions.

Installation and Setup

Execution Description The installation of both virtual machines was done using the default XtreemOS installer.

Galeb was installed and compiled on node 1, then copied to node 2. As no shared file system was used it is necessary to have the same executable files at the same locations on all nodes, otherwise migration will not work properly.

Procedure Results The installation of LinuxSSI was successful. After issuing the command `krg-adm cluster start` the two nodes are joined to logically form a single system.

Running Galeb without migration or checkpointing

Execution Description The tests were done on 2008-09-04. The application was started in the four ways enumerated in the corresponding test case description in D4.2.4.

Procedure Results Successful in all cases.

Migration scenarios

Execution Description The tests were done on 2008-09-04. The application was started in the four ways enumerated in the corresponding test case description in D4.2.4. The number of migrations was gradually increased to 10, unless the application failed consistently with a lower number of migrations.

Procedure Results If no process was migrated more than once, no problems were encountered. Furthermore, the master process could be migrated any number of times without problems, regardless of whether the migration was triggered while the slave processes were calculating or after they had finished and the master only had to collect their results.

However, migrating the calculating (slave) process itself turned out to be very problematic. If the calculation was run as a stand-alone application, it consistently crashed during the second migration. If the calculation was run in parallel, migrating slaves twice always caused at least one of the migrated slaves to fail. Some of the migrated slaves and all others finished successfully.

Anomalous events As described above, migrating a calculating process twice consistently crashed the process. Because the error occurs even when the process is run stand-alone, we suppose that it has nothing to do with parallel execution or IPC. Detailed analysis is given in the incident report wp22-fm-ts02-galeb01-tir01.

Checkpointing scenarios

Execution Description The tests were done on 2008-09-04. The application was started in the four ways enumerated in the corresponding test case description in D4.2.4.

Procedure Results Checkpointing and restarting a stand-alone slave process was always successful. The results of the restarted process were the same as of the original one. Multiple successive checkpoints and restarts during the same run were also successful.

The checkpointer does not save the Sys V message queues used by the application. Restarting a checkpointed parallel application thus worked correctly only if, after the checkpoint, the original application was killed. In this case the message queues remained in the system and the restarted application continued to use them. If, however, the message queues don't exist anymore, either because the original application closed them or because the cluster was rebooted after the checkpoint was made, the master process cannot read the slaves' results and the application fails.

Anomalous events As described above, a parallel application was not always restarted successfully. For detailed analysis please see the incident report wp22-fm-ts02-galeb01-tir02.

The combined migration/checkpointing/migration scenario

Execution Description The tests were done on 2008-09-04. A stand-alone application was started, migrated, and checkpointed, following the procedure described in D4.2.4. A restart from the checkpoint was then triggered.

Procedure Results We noted that after migration to node 2 the checkpoint can only be initiated on node 2, while trying to initiate it from node 1 resulted in the following non-intuitive error:

```
[root@ssi1 galeb]# checkpoint 102246
Checkpointing application in which process 102246 is involved...
checkpoint: No such file or directory
```

This also meant that the restart could only be initiated on the node where the checkpoint was made.

That said, restart of a migrated stand-alone process was successful. The restarted process could also be further migrated once.

Checkpointing of a parallel application that has had one slave process migrated to node 2 produced checkpointing files in the `/var/chkpt/$PID` directory on both nodes. After restart each process correctly resumed execution on the same node that it had been checkpointed on. However, the final communication between the migrated slave and the master failed.

Anomalous events As described above, restarting a parallel application that has had one slave process migrated was not successful. Detailed analysis is given in the incident report wp22-fm-ts02-galeb01-tir03.

3.3.4 Test incident report – FM Galeb second migration

Test incident report identifier: wp22-fm-ts02-galeb01-tir01

Summary

As notified in wp22-fm-ts02-galeb01-tl01, the calculating process of the Galeb application fails on second migration, regardless of whether it is run stand-alone or as a slave process.

The same incident happened during the first testing in October 2007. The bug has not been fixed despite being reported it in D4.2.4.

Incident Description

If the process is run stand-alone, the second migration causes either a segmentation fault or the error shown in Figure 3.1. The second migration of a slave of the parallel Galeb always causes the error shown in the figure.

Impact

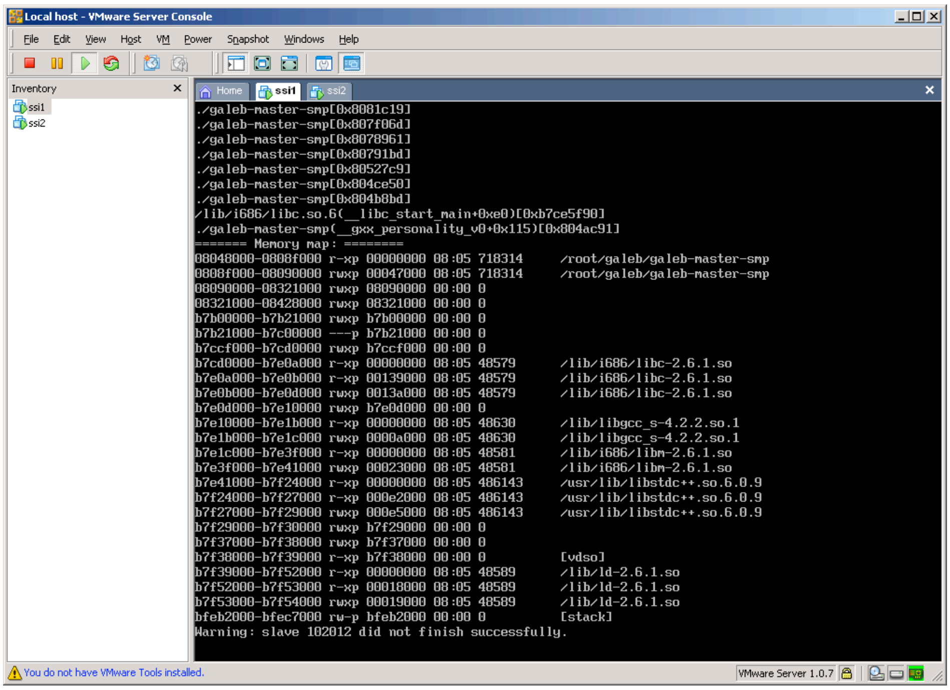
The described error is a critical bug in LinuxSSI and as such should be fixed quickly. It prevents testing more complicated migration and combined migration/checkpointing scenarios. It also prevents testing the scheduler because migration is a basic functionality required by schedulers.

3.3.5 Test incident report – FM Galeb parallel application restart

Test incident report identifier: wp22-fm-ts02-galeb01-tir02

Summary

As notified in wp22-fm-ts02-galeb01-tl01, restarting a checkpointed parallel application was not always successful.



```
./galeb-master-smp[0x8081c19]
./galeb-master-smp[0x807f06d]
./galeb-master-smp[0x8078961]
./galeb-master-smp[0x80791bd]
./galeb-master-smp[0x80527c9]
./galeb-master-smp[0x804ce50]
./galeb-master-smp[0x804b8bd]
/lib/i686/libc.so.6(__libc_start_main+0xe0)[0xb7ce5f90]
./galeb-master-smp(__gxx_personality_v0+0x115)[0x804ac91]
======= Memory map: ========
08048000-0808f000 r-xp 00000000 08:05 718314     /root/galeb/galeb-master-smp
0808f000-08090000 rwxp 00047000 08:05 718314     /root/galeb/galeb-master-smp
08090000-08321000 rwxp 08090000 00:00 0
08321000-08428000 rwxp 08321000 00:00 0
b7b00000-b7b21000 rwxp b7b00000 00:00 0
b7b21000-b7c00000 ---p b7b21000 00:00 0
b7ccf000-b7cd0000 rwxp b7ccf000 00:00 0
b7cd0000-b7e0a000 r-xp 00000000 08:05 48579      /lib/i686/libc-2.6.1.so
b7e0a000-b7e0b000 r-xp 00139000 08:05 48579      /lib/i686/libc-2.6.1.so
b7e0b000-b7e0d000 rwxp 0013a000 08:05 48579      /lib/i686/libc-2.6.1.so
b7e0d000-b7e10000 rwxp b7e0d000 00:00 0
b7e10000-b7e1b000 r-xp 00000000 08:05 48630      /lib/libgcc_s-4.2.2.so.1
b7e1b000-b7e1c000 rwxp 0000a000 08:05 48630      /lib/libgcc_s-4.2.2.so.1
b7e1c000-b7e3f000 r-xp 00000000 08:05 48581      /lib/i686/libm-2.6.1.so
b7e3f000-b7e41000 rwxp 00023000 08:05 48581      /lib/i686/libm-2.6.1.so
b7e41000-b7f24000 r-xp 00000000 08:05 486143     /usr/lib/libstdc++.so.6.0.9
b7f24000-b7f27000 r-xp 000e2000 08:05 486143     /usr/lib/libstdc++.so.6.0.9
b7f27000-b7f29000 rwxp 000e5000 08:05 486143     /usr/lib/libstdc++.so.6.0.9
b7f29000-b7f30000 rwxp b7f29000 00:00 0
b7f37000-b7f38000 rwxp b7f37000 00:00 0
b7f38000-b7f39000 r-xp b7f38000 00:00 0          [vdso]
b7f39000-b7f52000 r-xp 00000000 08:05 48589      /lib/ld-2.6.1.so
b7f52000-b7f53000 r-xp 00018000 08:05 48589      /lib/ld-2.6.1.so
b7f53000-b7f54000 rwxp 00019000 08:05 48589      /lib/ld-2.6.1.so
bfeb2000-bfec7000 rw-p bfeb2000 00:00 0          [stack]
Warning: slave 102012 did not finish successfully.
```

Figure 3.1: The error caused by the second migration of a Galeb process

Incident Description

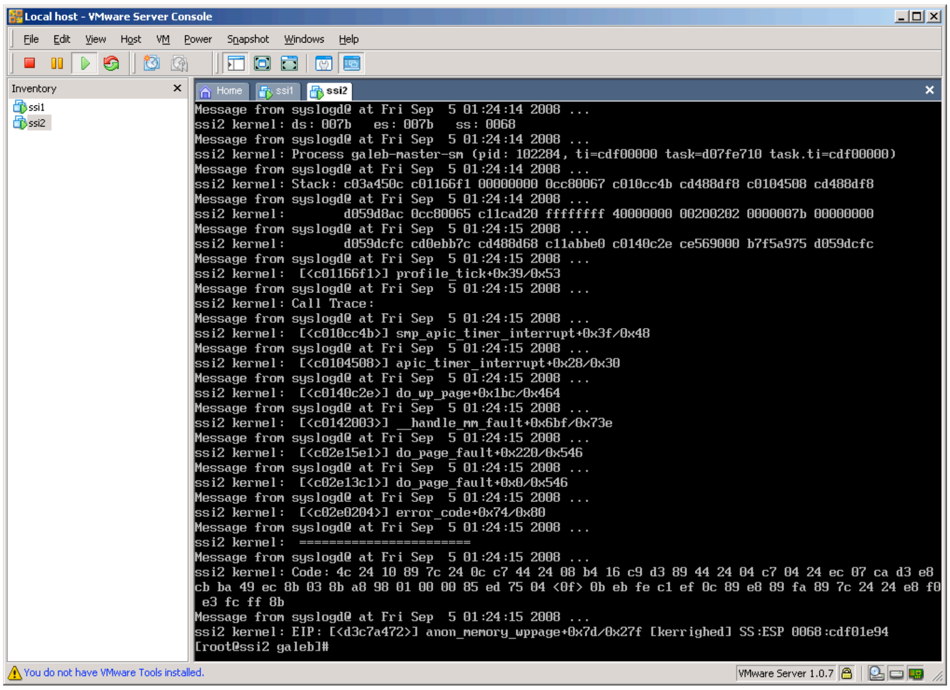
The incorrect behaviour reported in D4.2.4 was partly improved. The `waitpid()` system call made by the restarted master process now correctly waits for the restarted slaves.

However, the Sys V message queues are still not saved in the checkpoint. The restarted application will thus only work if the original one is killed immediately after the checkpoint and the cluster is not rebooted before restart, so that the message queues remain in the same state.

The behaviour is the same regardless of whether the checkpoint is made while the slave processes are still calculating or after they are done but before the master collects their results.

Impact

The described error represents an important shortcoming of the implemented checkpointer. The checkpointing and restarting mechanisms should be improved to successfully cope with parallel applications that use System V IPC. The error also prevents performing more complicated combined checkpointing/migration tests.



Local host - VMware Server Console
File Edit View Host VM Power Snapshot Windows Help
Inventory
ssi1
ssi2
Home
ssi1
ssi2
You do not have VMware Tools installed.
VMware Server 1.0.7

```
Message from syslogd@ at Fri Sep  5 01:24:14 2008 ...
ssi2 kernel: ds: 007b   es: 007b   ss: 0068
Message from syslogd@ at Fri Sep  5 01:24:14 2008 ...
ssi2 kernel: Process galeb-master-sm (pid: 102284, ti=cdf00000 task=d07fe710 task.ti=cdf00000)
Message from syslogd@ at Fri Sep  5 01:24:14 2008 ...
ssi2 kernel: Stack: c03a450c c01166f1 00000000 0cc80067 c010cc4b cd488df8 c0104508 cd488df8
Message from syslogd@ at Fri Sep  5 01:24:14 2008 ...
ssi2 kernel:        d059d8ac 0cc80065 c11cad20 ffffffff 40000000 00200202 0000007b 00000000
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:        d059dcfc cd0ebb7c cd488d68 c11abbe0 c0140c2e ce569000 b7f5a975 d059dcfc
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c01166f1>] profile_tick+0x39/0x53
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel: Call Trace:
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c010cc4b>] smp_apic_timer_interrupt+0x3f/0x48
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c0104508>] apic_timer_interrupt+0x28/0x30
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c0140c2e>] do_wp_page+0x1bc/0x464
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c0142003>] __handle_mm_fault+0x6bf/0x73e
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c02e15e1>] do_page_fault+0x220/0x546
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c02e13c1>] do_page_fault+0x0/0x546
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  [<c02e0204>] error_code+0x74/0x80
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel:  =======================
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel: Code: 4c 24 10 89 7c 24 0c c7 44 24 08 b4 16 c9 d3 89 44 24 04 c7 04 24 ec 07 ca d3 e8
cb ba 49 ec 8b 03 8b a8 98 01 00 00 85 ed 75 04 <0f> 0b eb fe c1 ef 0c 89 e8 89 fa 89 7c 24 24 e8 f0
 e3 fc ff 8b
Message from syslogd@ at Fri Sep  5 01:24:15 2008 ...
ssi2 kernel: EIP: [<d3c7a472>] anon_memory_wppage+0x7d/0x27f [kerrighed] SS:ESP 0068:cdf01e94
[root@ssi2 galeb]#
```

Figure 3.2: The error that occurs when the migrated-and-restarted slave process tries to send its results to the master process

3.3.6 Test incident report – FM Galeb checkpoint after migration

Test incident report identifier: wp22-fm-ts02-galeb01-tir03

Summary

As notified in wp22-fm-ts02-galeb01-tl01, restarting a parallel application that has had one slave process migrated was not successful.

Incident Description

The parallel version of Galeb with two slaves was started on node 1. One of the slaves was migrated to node 2, after which the checkpoint was made. All the processes were then killed (otherwise the restart would certainly fail, as described in wp22-fm-ts02-galeb01-tir02.

After restart the master as well as the slaves correctly resumed operation. The non-migrated slave also correctly sent the results to the master process. However, the migrated slave failed to communicate its results to the master as shown in the screenshot in Figure 3.2.

Impact

The described bug prevents correct restart of parallel applications, since those will always have some process migrated to other nodes to make use of their CPUs. However, the bug is irrelevant until the errors reported in wp22-fm-ts02-galeb01-tir01 and wp22-fm-ts02-galeb01-tir02 are fixed.

3.4 Installation of Business Application

This describes the evaluation of the install CD packaged in WP4.1. It therefore provides an assessment of the four new requirements introduced in the deliverable.

3.4.1 Test plan – Install

Test plan identifier: wp41-pkg-ts02-tp

Introduction

This test plan covers the installation CD for XtreemOS, hence covering the requirements of the following:

- Usability of the install CD and the provided documentation
- Flexibility of the startup with respect to selecting different configurations
- Compatibility with standard Linux utilities and commands including updates
- Installation of a more complex application on XtreemOS

The purpose of the install CD is to make the installation of the various XtreemOS components and services easy for a system administrator, taking care of all dependencies and providing documentation about configuration steps that need to be taken.

At the time of writing the deliverable, the install CD was retrieved from a repository hosted by the organisation responsible for the packaging.

These particular tests were not included in previous deliverables as the install CD was not available. This marks a transition from testing individual components to testing them within a single, packaged operating system bundle.

Test Items

The tests will be done on the version of the XtreemOS install CD release packaged on 2008-08-26. The contact person and organisation for the version of the install CD is Antoine Giniès from Mandriva (`aginies@mandriva.com`).

Features to be Tested

The tested feature will be the amount of effort required to boot up two physical machines with XtreemOS, create a cluster using Linux-SSI and install an application on top of the cluster. Secondly, the standard Linux commands for querying and managing networking, memory, filesystem and CPU.

Features not to be Tested

Security is currently not tested and it is assumed that the user is logged in a root to each machine. The limitations of this are however known. Especially for the application to be installed, it is important that the security features of XtreemOS are included.

Approach

The purpose of these tests are to evaluate the current version of the install CD software, provide feedback to developers and packagers, and to check which installation requirements are fulfilled, which are partially fulfilled, and which remain to be done. The approach is to create an execution platform for XtreemOS that is not trivial and demonstrates more than one machine being included in a Grid-like setting. For that reason a cluster is created and all features tested to see if they continue to behave stably as if a single machine. This is firstly done with two physical machines and repeated with four virtual machines, two on each physical host.

Item Pass Criteria

The pass criteria for the evaluations are based on the observation of expected behaviors. The tests are therefore based on knowledge of the underlying hardware (i.e. the processor, amount of memory, storage and networking interface), as well as the standard Linux commands as specified by the Linux Standard Base (LSB) Specification 3.2[7]. We did not perform a full LSB certification but apply similar principles based on the utilities available.

Testing Tasks

The test preparation requires the following tasks:

1. download of the latest install CD iso
2. mounting of the install CD
3. following the documented installation procedures
4. selection of appropriate kernel version
5. download and availability of application installer and all relevant shared libraries

Environmental Needs

The environment requirements are with respect to storage, CPU, memory and networking. There is no assumption made of simple mobile phones and limited memory devices in this basic installation test. All machines must be capable of installing and running the packaged XtreemOS release from 2008-08-26. They also need to be networked over Ethernet to allow the creation of a cluster.

Responsibilities

Similar experiments have been carried out by all partners. However, the results related in this deliverable for this particular test case have been conducted by SAP.

Schedule

The tests were trivial and could be performed in one day, as the install CD made setting up the various components rather straightforward.

Risks and Contingencies

There were two risks: (1) hardware failure and (2) software failure. Hardware failure can also arise from the hardware being incompatible with the XtreemOS kernel. This can only be done on a trial and error basis as XtreemOS will only be as portable as the Linux kernel on which it is built. Software failure is a risk but is also a result in this form of evaluation. By analysing the logs from different failures we can inform development about certain anomalies, but also learn the limitations that XtreemOS or comparable software architectures would have.

3.4.2 Test design specification

Test design specification identifier: wp41-pkg-ts02-sap-tds01
Test plan reference: ??

Features to be Tested

(see test plan: wp41-pkg-ts02-tp)

Approach Refinements

Along with the application installation facilities such as `rpm`, the CPU and memory usage were checked using the `top` command.

Test Identification

The test case that is the subject of this test design is **??**.

Feature Pass/Fail Criteria

(see test plan: wp41-pkg-ts02-tp)

3.4.3 Test case specification – Installation of XtreemOs and Installation of Application on XtreemOS

Test case specification identifier: wp41-pkg-ts02-sap01-tcs01

Test Items

This test case tests the viability of installing and running a business application on top of XtreemOS. The main requirements for business applications are reliability, availability, scalability, security and manageability. Before being able to investigate these requirements, it is first necessary that there are no technical conflicts between the features of the application and the operating system.

Input Specifications

There are three steps to installing the particular business application:

1. *Prepare system*: this involves the setting of environment variables that are necessary for the installation of the application. As a minimum

2. *Install software*: there are several scripts to be installed for the services, application server and database of the application

3. *Install license*: the application requires that a license be obtained and installed before it can be operational

Output Specifications

When the application is properly installed and made available, then a license successfully installed message is returned.

3.4.4 Test procedure specification

Test procedure specification identifier: wp41-pkg-ts02-sap01-tps01
Test design specification reference: ??
Test case specification reference: wp41-pkg-ts02-sap01-tcs01
Test log identifier: wp41-pkg-ts02-sap01-tl01

Purpose

This procedure executes the test case wp41-pkg-ts02-sap01-tcs01.

Procedure Steps

Configuration The XtreemOS Installation configuration was as follows:

- `ACPI = off` (limitation of ACPI on the servers used for this test, likely a HW issue)
- `Linux-SSI enabled`
- `VO demos not installed`
- `X environment not installed` (due to conflict with Linux-SSI)

After successful completion of the XtreemOS installation procedure, deployment of an SAP system was attempted. Due to the restriction to a 32-bit kernel, the last available 32-bit SAP system was chosen: SAP R/3 v4.7.

Set Up The XtreemOS install CD was installed on each node individually. The installation files for the application were store on one node.

Start The install script for the application was started on a single node in the cluster.

Wrap Up Observe if the application is correctly installed. If not, then retrieve the relevant OS and application logs in order to determine the cause of failure.

Contingencies As the application failed, the contingency was to learn and document as much as possible about the cause of the error, such that future troubleshooting would be made easier. Instead of testing a complex architecture, it was decided in the future to evaluate the installation of a simpler system with similar operational properties.

3.4.5 Test log

Test log identifier: wp41-pkg-ts02-sap01-tl01

Description

The test procedure wp41-pkg-ts02-sap01-tps01 was executed within the SAP network. Two Intel dual-core x86-64, 2.4 GHz x 2 physical nodes were used, each with 4GB RAM and 250 GB HDD.

The machines were never installed with XtreemOS before, but both had a version of SuSE Linux Enterprise installed, along with VMWARE as the VM technology.

Installation and Setup

Execution Description The installation of XtreemOS and the SAP application were done using two different settings. Firstly, XtreemOS was just installed on the two physical machines and a cluster created between these two machines. Secondly, XtreemOS was installed on top of four virtual machines, two per physical host, where the host operating system was SUSE Linux and the guest VM technology was VMWARE.

Procedure Results Installation of XtreemOS and creation of a cluster in each of the two test bed settings was successful.

Anomalous Events There was only one anamolous event that occurred whenever the `top` command was issued in the cluster of virtual machines - the machines ceased to respond. Even leaving them for sometime. Secondly, bringing down one machine in the cluster caused the entire cluster to be brought down.

Running the Test Procedure

Executing the tests were trivial. There are essentially 4 steps done over 2 iterations, where iteration 1 was on the 2 physical machines and iteration 2 was on the 4 virtual machines:

1. install OS using the linuxssi kernel version
2. start a cluster using the `krgadm` utility
3. retrieve the application installation files
4. run the application's install script

Anomalous Events

After a number of attempted executions of the application's install script, it was clear that a kernel version incompatibility was causing the installation to fail each time. The following is the output received from the strace utility:

```
execve("/bin/sh", ["sh", "install"], [/* 52 vars */]) = 0
brk(0)                                  = 0x8106000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=25969, ...}) = 0
mmap2(NULL, 25969, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f63000
close(3)                                = 0
open("/lib/libtermcap.so.2", O_RDONLY)  = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\360\n\0"..., 512) = 512
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7f62000
fstat64(3, {st_mode=S_IFREG|0755, st_size=11584, ...}) = 0
mmap2(NULL, 14504, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb7f5e000
mmap2(0xb7f61000, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2)
```

```
      = 0xb7f61000
close(3)                                = 0
open("/lib/libdl.so.2", O_RDONLY)       = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0p\n\0\000"..., 512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=9692, ...}) = 0
mmap2(NULL, 12412, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb7f5a000
mmap2(0xb7f5c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1)
      = 0xb7f5c000
close(3)                                = 0
open("/lib/i686/libc.so.6", O_RDONLY)   = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\360`\1"..., 512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1298800, ...}) = 0
mmap2(NULL, 1308112, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb7e1a000
mmap2(0xb7f54000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x139)
      = 0xb7f54000
mmap2(0xb7f57000, 9680, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)
      = 0xb7f57000
close(3)                                = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7e19000
      set_thread_area({entry_number:-1 -> 6, base_addr:0xb7e196c0, limit:1048575, seg_32bit:
      1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, useable:1}) = 0
mprotect(0xb7f54000, 4096, PROT_READ)   = 0
mprotect(0xb7f84000, 4096, PROT_READ)   = 0
munmap(0xb7f63000, 25969)               = 0
rt_sigprocmask(SIG_BLOCK, NULL, [], 8)  = 0
open("/dev/tty", O_RDWR|O_NONBLOCK|O_LARGEFILE) = 3
close(3)                                = 0
brk(0)                                  = 0x8106000
brk(0x8107000)                          = 0x8107000
open("/usr/share/locale/locale-archive", O_RDONLY|O_LARGEFILE)
   = -1 ENOENT (No such file or directory)
brk(0x8108000)                          = 0x8108000
open("/usr/share/locale/locale.alias", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=2586, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7f69000
read(3, "# Locale name alias data base.\n#"..., 4096) = 2586
brk(0x8109000)                          = 0x8109000
brk(0x810a000)                          = 0x810a000
read(3, "", 4096)                       = 0
close(3)                                = 0
munmap(0xb7f69000, 4096)                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_IDENTIFICATION", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=366, ...}) = 0
mmap2(NULL, 366, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f69000
close(3)                                = 0
open("/usr/lib/gconv/gconv-modules.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=26052, ...}) = 0
mmap2(NULL, 26052, PROT_READ, MAP_SHARED, 3, 0) = 0xb7e12000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_MEASUREMENT", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=23, ...}) = 0
mmap2(NULL, 23, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f68000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_TELEPHONE", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=56, ...}) = 0
mmap2(NULL, 56, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f67000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_ADDRESS", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=127, ...}) = 0
mmap2(NULL, 127, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f66000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_NAME", O_RDONLY) = 3
```

```
fstat64(3, {st_mode=S_IFREG|0644, st_size=77, ...}) = 0
mmap2(NULL, 77, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f65000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_PAPER", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=34, ...}) = 0
mmap2(NULL, 34, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f64000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_MESSAGES", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_MESSAGES/SYS_LC_MESSAGES", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=52, ...}) = 0
mmap2(NULL, 52, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f63000
close(3)                                = 0
brk(0x810b000)                          = 0x810b000
open("/usr/share/locale/en_GB.UTF-8/LC_MONETARY", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=290, ...}) = 0
mmap2(NULL, 290, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7e11000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_COLLATE", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=932330, ...}) = 0
mmap2(NULL, 932330, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7d2d000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_TIME", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=2378, ...}) = 0
mmap2(NULL, 2378, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7d2c000
close(3)                                = 0
brk(0x810c000)                          = 0x810c000
open("/usr/share/locale/en_GB.UTF-8/LC_NUMERIC", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=54, ...}) = 0
mmap2(NULL, 54, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7d2b000
close(3)                                = 0
open("/usr/share/locale/en_GB.UTF-8/LC_CTYPE", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=254076, ...}) = 0
mmap2(NULL, 254076, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7cec000
close(3)                                = 0
getuid32()                              = 0
getgid32()                              = 0
geteuid32()                             = 0
getegid32()                             = 0
rt_sigprocmask(SIG_BLOCK, NULL, [], 8)  = 0
brk(0x810d000)                          = 0x810d000
time(NULL)                              = 1219162812
brk(0x810e000)                          = 0x810e000
open("/proc/meminfo", O_RDONLY)         = 3
fstat64(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7ceb000
read(3, "MemTotal:      3624196 kB\nMemFre"..., 1024) = 728
close(3)                                = 0
munmap(0xb7ceb000, 4096)                = 0
brk(0x810f000)                          = 0x810f000
rt_sigaction(SIGCHLD, {SIG_DFL}, {SIG_DFL}, 8) = 0
rt_sigaction(SIGCHLD, {SIG_DFL}, {SIG_DFL}, 8) = 0
rt_sigaction(SIGINT, {SIG_DFL}, {SIG_DFL}, 8) = 0
rt_sigaction(SIGINT, {SIG_DFL}, {SIG_DFL}, 8) = 0
rt_sigaction(SIGQUIT, {SIG_DFL}, {SIG_DFL}, 8) = 0
rt_sigaction(SIGQUIT, {SIG_DFL}, {SIG_DFL}, 8) = 0
rt_sigprocmask(SIG_BLOCK, NULL, [], 8)  = 0
rt_sigaction(SIGQUIT, {SIG_IGN}, {SIG_DFL}, 8) = 0
uname({sys="Linux", node="localhost", ...}) = 0
brk(0x8110000)                          = 0x8110000
brk(0x8111000)                          = 0x8111000
brk(0x8112000)                          = 0x8112000
stat64("/home/xtreemuser/kerneltest", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat64(".", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
getpid()                                = 107963
brk(0x8113000)                          = 0x8113000
getppid()                               = 107962
stat64(".", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat64("/sbin/sh", 0xbfc1ecac)          = -1 ENOENT (No such file or directory)
stat64("/usr/sbin/sh", 0xbfc1ecac)      = -1 ENOENT (No such file or directory)
stat64("/bin/sh", {st_mode=S_IFREG|0755, st_size=755276, ...}) = 0
:
:
exit_group(154)                         = ?
```

The application logs were also available and showed that the expected progress was made up to the point of actually starting the database, where BEN is the name of a database instance in the application:

```
:
OK
*** Tue Aug 19 17:20:36 2008
param_addvolume 6 DATA /usr/sap/BEN/devspaces/DISKD0006 F 691200
---
OK
*** Tue Aug 19 17:20:36 2008
param_startsession
---
OK
*** Tue Aug 19 17:20:36 2008
param_put CAT_CACHE_SUPPLY 3000
---
OK
*** Tue Aug 19 17:20:36 2008
param_put CACHE_SIZE 118328
---
OK +++ Tue Aug 19 17:20:36 2008 dbmcli -t
/tmp/createdb.sh.109198.log -s -d BEN -u control,control +++ Tue Aug
19 17:20:36 2008 Connection established to node (local) for database
BEN
*** Tue Aug 19 17:20:36 2008
db_start
---
ERR -24994,ERR_RTE: runtime environment error 1,Check knldiag!
Kernel died before reaching ADMIN state
*** Tue Aug 19 17:20:36 2008
util_connect CONTROL,CONTROL
---
ERR -24988,ERR_SQL: sql error 1,database not running: no request
pipe
*** Tue Aug 19 17:20:36 2008
db_activate SUPERDBA,ADMIN
:
:
```

3.4.6 Test incident report – Installation of XteemOS

Test incident report identifier: wp41-pkg-ts02-sap01-tir01
Test log reference: wp41-pkg-ts02-sap01-tl01

Summary

The installation of XtreemOS in each of the cases explored was straightforward. By following the documentation step by step it was possible to have a working cluster of 2 machines in around 30 mins. The installation failure of the application on top of XtreemOS was a technical difficulty that we had hoped could have been resolved. However, this has to do with the application being compiled for an older kernel version than XtreemOS. There were therefore some functions that could not be executed. Access to the source of the application was not available, and the effort to change this would have been tremendous in any event. We also tried forcing XtreemOS to work with an older kernel, but this caused, naturally, other components to fail and the OS to act unstable.

There was only one problem while running the Linux-SSI cluster that arose when checking the memory and CPU availability of a cluster on top of virtual machines. This was subsequently identified as a problem with the network optimisation of the virtual machine software, and a solution was proposed by the developers of Linux-SSI.

Incident Description

Using the Linux tracing utilities and application logs, it was possible to follow and determine the reason for failure of the application. The ability to comprehensively troubleshoot and identify the sources of problems is an important feature of an operating system.

Having identified a kernel mismatch problem, the last available alternative was to use the `LD_ASSUME_KERNEL` environment variable to force compatibility. In doing so, however, compatibility with all other available programs was broken due to the C libraries looking for an incorrect kernel number, thereby rendering the entire system useless.

At this point, it was decided to wait for a version of XtreemOS with a 64-bit kernel, in order to better test a more recent SAP NetWeaver system.

Impact

Applications compiled for older kernel versions are to be excluded from any tests. During development there will be phases when applications and/or the OS need to be changed, but this is part of refining the requirements for the operating system. There are different levels of porting applications that need to be considered, where the aim is to have minimal change i.e. configuration scripts. If there is a need to change internal code or libraries, then this means poor "portability". Altering the source of applications must be motivated by an improvement in performance, reduction in memory required, improvement in manageability and scalability; the benefits must outweigh the effort required to change.

3.4.7 Test summary report – Install

Test summary report identifier: wp41-pkg-ts02-tsr

Summary

The testing of installing a 32-bit business application (SAP R/3) on XtreemOS (as packaged in XtreemOS released on 2008-08-26) was evaluated by following the normal installation steps for the application. It is expected that a software previously compiled for execution on a Linux kernel should still execute on top of XtreemOS, given that all shared libraries and environment variables were correctly set. Note that the particular 32-bit version of the application is no longer supported, such that a 64-bit version is required for more up-to-date testing.

Comprehensiveness Assessment

The evaluation was pursued until a dead-end was reached. This included the modification of scripts, environment variables and shared libraries. At some stage we also tried to label the kernel version, but this resulted in the kernel misbehaving and going into panic.

Summary of Results

Installation currently fails as a result of incompatibilities between the versions of shared libraries installed with XtreemOS and those required by the application when compiled. These sort of dependencies are difficult to resolve, but are more attributed to modifications made to the application as opposed to bad design of XtreemOS.

Evaluation

The attempt to install SAP R/3 on top of XtreemOS has progressed better than previous attempts to install on top of Kerrighed. Although there are still problems to get the application to actually execute and run, the recognition of why this is the case is easily detectable. As there is no value in further pursuing the installation of the 32-bit version of application on top of XtreemOS, this will be discontinued. We will rather use the results for better planning of what we hope to achieve with the 64-bit version of the operating system. Should this not be made available, we will investigate using smaller sub-modules of the large software as opposed to the entire distribution. This might be of more benefit for bringing XtreemOS up to industrial scale, as well as providing some insights for the application provider into the technical possibilities for investing into similar OS technologies.

3.5 Summary

The previous sections presented the test documentation and the evaluation results for [insert list of components]. In the following the main results are summarized.

3.5.1 Overview of Evaluation Results

In this section, the evaluation results of the selected XtreemOS components are summarized.

3.5.2 Assessment of the Requirements Fulfillment Status

The experiments allowed for testing the functionalities of various isolated XtreemOS components since an integrated version was not available for evaluation. The documentations of these tests also include an insight in how far certain requirements are fulfilled. In this section, we give an overview of these assessments which have been consolidated in the table below. A color coding is applied to visualize the fulfillment status:

🔴 red: requirement is not fulfilled.

🟡 yellow: requirement is "partially" fulfilled.

🟢 green: requirement is fulfilled.

⚫ grey: fulfillment of requirement could not be tested (e.g. respective functionality was not available for testing)

If multiple tested components contribute to a certain requirement indented rows are added containing the assessment for each such component (subsequently referred to as sub-assessment). In this case, the overall assessment of the fulfillment of the requirement can not be better than the worst sub-assessment of all contributing components. Regarding requirements which have been affected by the revision process for this deliverable, we added character "n" (new requirement) or character "u" (updated requirement) behind the requirement number in the first column of the table. Since these requirements have been changed or introduced only recently, the assessment of the fulfillment status has to be considered as tentative. Overall, this overview provides feedback to the developers and to project management using it to identify gaps, errors, white areas and to trace the progress of the project with respect to the fulfillment of requirements. Note, for reasons of providing a compact and consolidated overview of the requirements fulfillment, the table only gives a brief summary of the requirement descriptions. For cases where there is no description or summary, this indicates that there is currently no development to report on. If there is some development but no test performed, the description and comment are still provided. Given that the requirements R1 - R112 are actual links it is proposed to read the table in the electronic version of the document to make use of the clickable links to the requirements text given in the Appendix.

R.no.	Fulfillment	Brief desc.	Comments
R1	🟡 yellow	XtreemOS equally supports data-intensive and computation-intensive applications	scalability and performance are appropriate for some applications
R2	🟢 green	Support for heterogeneous hardware	Has been installed on various architectures including both physical and virtual machines
R3	🟡 yellow	Support for Grids with variable number of nodes	handles few nodes but not yet tested on very large scales
R4	⚫ grey	handle virtual (replicated) nodes in order to increase robustness in case of failures	the virtual nodes technology is available but not yet tested
R5	🔴 red	Nodes can dynamically be added to the Grid and removed from the Grid	The version of Linux-SSI tested did not support this requirement. Removing 1 node causes the entire cluster to be brought down.
R6	🟡 yellow	Migration of running applications	Migration worked only once in the test case. It is therefore not yet stable. Checkpointing was however always successful.
R7	🟡 yellow	Execution and Migration of Java Applications	Java programs can be executed. Migration of Java programs has not been tested.
R8	⚫ grey	Several Terabytes storage capacity	
R9	🟡 yellow	Several Gigabytes working memory capacity	Only tested with machines that have up to 4 GB memory
R10	⚫ grey	Support for software licensing mechanisms	This requirement needs to be reviewed
R11	🟡 yellow	Delivered as packaged release	The Mandriva distributions have been tested so far
R12	⚫ grey	fast and reliable communication	This is not yet evaluated but some of the components required to do this are now available
R13	⚫ grey	Compatible with highspeed interconnect standards	No tests with Infiniband performed
R14	⚫ grey	Support for Iv6	-
R15	⚫ grey	Support system 5 semaphores	No explicit tests have been performed
R16	⚫ grey	Support for applications on 64-bit architectures	this is still not available as a distribution
R17	🟡 yellow	SMP and multicore support	Runs on multicore architectures but `top` does not report the existence of multiple processors

R18	🟡 yellow	Support for virtual machines	Has been installed and tested on virtual machines but no significant testing of isolation has been done. No testing of VMs on XtreemOS has been done.
R19	⚫ grey	Support for multicast	Not yet tested
R20	🟡 yellow	Provide access to various Grid Services	Core services are currently included for job submission and Grid security
R21	⚫ grey	VOs manage a large amount of users	No test case defined
R22	🟡 yellow	Support both interactive and batch jobs	Batch jobs are supported but interactive not yet tested - developers however say they are not yet supported
R23	🟡 yellow	Provide role management	Support is included in the XOS certificates and policies. It is nevertheless complicated to set up role policies.
R24	🟢 green	Provide a means of managing VOs	VOlife web interface is excellent, the command-line interface is also good
R25	🟡 yellow	VO user accounts have to be independent from local user accounts	Will be green once the reported bug is fixed
R26	⚫ grey	Dynamically change the composition of VOs during application runtime	not tested; availability is uncertain
R27	⚫ grey	Lifetime of a VO must be guaranteed	not tested
R28	🟢 green	Allow multiple VOs on the same node within specified constraints	The domain admin must allow each node to be added to a VO, thus limiting their number. Other parts of R28 are fulfilled.
R29	🟢 green	A VO management interface has to be provided	These interfaces have been included in the VO-Life module of the OS see R24
R30	⚫ grey	VO management actions must be completed within a specified maximum amount of time.	The packaging is not yet sufficient to test this
R31	⚫ grey	XtreemOS has to support communication between VOs	not tested
R32	🟡 yellow	Automatic failure detection, checkpointing and restart	Checkpointing and migration are implemented and work for some applications. There is no automatic fault detection.
R33	⚫ grey	Checkpoint restart must mimic the original environment	Original application could not be tested on packaged solution
R34	⚫ grey	Notify checkpoint restart	not tested

R35	🟡 yellow	various ways of checkpoint initiation	partially fulfilled
R36	⚫ grey	Fast checkpoint restart performance	Performance tests have not yet been done
R37	🟢 green	Implementation of checkpointing as a kernel module	Already implemented as a kernel module
R38	⚫ grey	Customized checkpointing and restart	Not yet tested for this deliverable
R39	🟡 yellow	Save process state information to be restored on checkpoint/restart	There is some implementation
R40	🟢 green	Set the number of federation nodes	This is available in the install CD and can be set during boot up
R41	⚫ grey	Scheduler must have access to various node, application and grid information	Not yet tested for this deliverable
R42	⚫ grey	Specification of service qualities in federations	Not yet at a stage to test this
R43	🔴 red	Node properties constraints in federations	This has not been implemented and might have to be revisited - Linux-SSI assumes a homogeneous architecture
R44	🟢 green	Shared file system within a federation	KDFS is packaged and available with Linux-SSI
R45	🔴 red	Changing number of federation nodes	This cannot currently work. Will have to wait until a hot-plug mechanism is implemented
R46	⚫ grey	SSI scheduling of replicated processes	not yet tested
R47	🟡 yellow	Automated checkpoint and restart	Checkpointing and restart is implemented but not fully automated as there is no automated failure detection
R49	🟡 yellow	Other API Standards as basis for XtreemOS API	Support for SAGA exists
R50	🟡 yellow	Demand for POSIX compliance	Uncertain about all components
R51	🟡 yellow	XtreemOS must support several different programming languages	For C, C++ and Java there is support - these are the most important
R52	⚫ grey	It should be possible to use XtreemOS as a backend for GT4 WS-GRAM	Not tested this time. Uncertain of availability
R53	⚫ grey	Provide for a high availability of allocated nodes during the whole job execution	unavailable for testing
R54	⚫ grey	Bandwidth guarantees for applications	This depends on the networking infrastructure's protocols, topology and distribution

R55	● grey	Prioritisation of Grid services	unavailable for testing
R56	● grey	Supply quality of service measures	unavailable for testing
R57	● grey		
R58	● grey		
R59	● grey		
R60	● grey		
R61	● grey		
R62	● grey		
R63	● grey		
R64	● grey		
R65	● grey		
R66	● grey		
R67	● grey		
R68	● grey		
R69	● grey		
R70	● grey		
R71	● grey		
R72	● grey		
R73	● grey		
R74	🟡 yellow	Possible to concurrently read from files which are open for write access	Reported as possible
R75	🟢 green	The usual UNIX metadata must be accessible	Possible via
R76	🟢 green	Directory structures defined and usable	Support for directories works
R77	🟡 yellow	Access control to parts of Meta-Data	Not fully integrated
R78	● grey		
R79	🟡 yellow	File access time must be below 10 s to prevent time-out errors in applications	Has not been rigorously tested
R80	● grey		
R81	● grey		
R82	● grey		
R83	🟡 yellow	Support transactional consistency for object sharing	Transactions still not tested
R84	● grey		
R85	🟡 yellow	Access to data enforced by appropriate access rights	Not completely integrated
R86	🟡 yellow	Confidential communication	Appropriate cryptographic functions and certificates in place

R87	🟡 yellow	Integrity of data	Appropriate cryptographic functions and certificates in place
R88	⚫ grey	Integrity of communications	Appropriate cryptographic functions and certificates in place
R89	⚫ grey	Single sign-on	Not yet tested
R90	⚫ grey		
R91	🟡 yellow	VO membership validation	Not tested on more complex scale
R92	⚫ grey		
R93	⚫ grey		
R94	⚫ grey		
R95	⚫ grey		
R96	🟡 yellow	Standard security specifications and utilities	Followed for most available implementations
R97	⚫ grey		
R98	⚫ grey		
R99	⚫ grey		
R100	🟡 yellow	Support different CPU performance constraints	Some work done with the mobile flavour
R101	⚫ grey		
R102	⚫ grey	XtreemOS-MD must support ARM architecture for PDAs and mobile phones	Internal software release not available yet.
R103	⚫ grey	XtreemOS-MD must support Java	Internal software release not available yet.
R104	⚫ grey	XtreemOS-MD must support web services client	Internal software release not available yet.
R105	⚫ grey	XtreemOS-MD must allow VO management	Internal software release not available yet.
R106	⚫ grey	MDs should be considered special nodes	Internal software release not available yet.
R107	⚫ grey	XtreemOS-MD must provide communication and job management functions	Internal software release not available yet.
R108	⚫ grey	XtreemOS-MD should support lightweight security methods	Internal software release not available yet.
R109 n	🟡 yellow	Intuitive and familiar installation	Some deviations exist. Configuration is still reported as being too complex
R110 n	🟡 yellow	Start of individual services independently	Some services still failed when trying to start
R111 n	🟢 green	Standard Linux system management utilities	The basic Linux system utilities are integrated and work
R112 n	🟡 yellow	No recompilation of applications	Has not been tested widely but is true for the most part

Out of the 112 requirements (4 new requirements added) defined by WP4.2, 3 requirements (12 less than last report) have been rated as not fulfilled, 31 requirements (15 more than last report) are fulfilled partially, and 10 requirements (9 more than last report) are rated as fulfilled. There is therefore some improvement in development and the speed with which problems are being resolved. For the remaining 68 requirements (10 less than last time), it was not possible to provide an assessment due to either of 2 reasons: (1) exclusion of the respective components from the XtreemOS install CD or (2) limited interest in the particular component from an application perspective. However, it should be noted that internal testing has been carried out by the developers and there are ongoing plans for larger demonstrators that will cover more features of XtreemOS. These demonstration activities are discussed in the next section.

Chapter 4

Planning of Demonstration Activities

This chapter will outline the planned demonstration activities that will take place in the last year of the project within WP 4.4, which starts in M37. The reason for this chapter is that M37 may be too late to start planning the demonstrations from scratch. For each of the applications the set up will be explained, as well as the potential advantages expected for different applications from XtreemOS.

4.1 Scaling and Adaptation for Large Business Applications

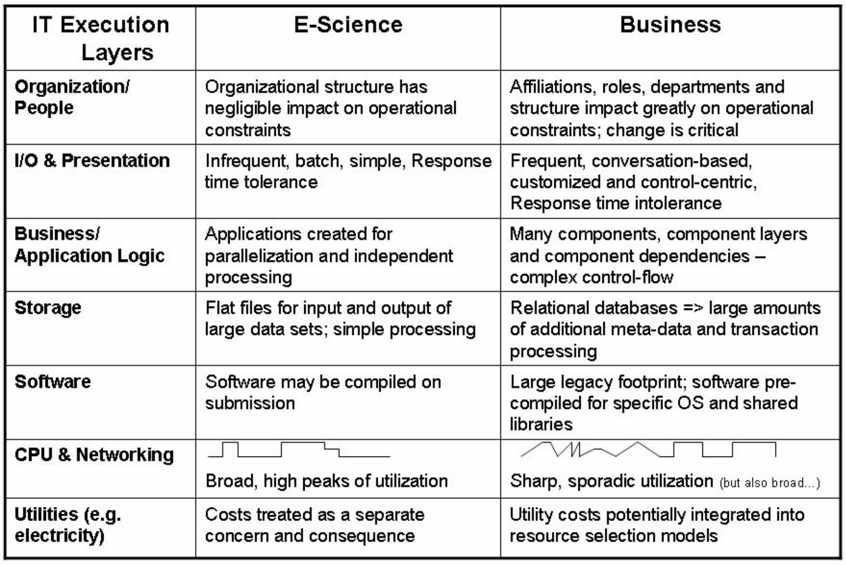
The majority of applications designed and tested on Grid infrastructures are either batch-jobs, non-interactive and can be parallel-enabled using message passing. In this section, the challenges for business applications are highlighted, as well as the requirements for XtreemOS they present. Furthermore, the ongoing evaluation of XtreemOS against these requirements are then described.

4.1.1 Overview of Business Applications

A business application is software that processes, provides and presents an organisationŠs data, executing aspects of its business logic and business processes in order to increase profitability, productivity and sustainability. Most large-scale business applications are composed of multiple layers (presentation, business logic, persistence), each having different storage, I/O and processing requirements. Secondly, quick (2s is the general acceptance limit) interactive response time is a core acceptance criteria for business applications, where users need to complete many transactions and retrieve information simultaneously. Especially for operational environments with 10,000s and more of users, scalability and efficiency of business applications becomes a serious requirement. In Grid computing, it is often the case that techniques such as parallelisation and distribution of jobs on multiple compute nodes are used in order to attain scalability and efficiency. In order to maintain the manageability of distribution, a SSI (Single System Image) technology is often introduced. For business applications, additional constraints need to be considered such as security and the architectural properties of the application with respect to interprocess communication. Much of the software written in the business applications we consider is based on shared memory, as this is more convenient to implement, given that the compiler is able to automatically optimise the software's binary. Secondly, although message-passing offers more portability of software, it comes with greater software overheads. However, shared memory software is not easily ported between different architectures and operating systems, limiting the way in which the scalability and efficiency advantages of a traditional grid infrastructure and OS can be recognised. Even if SSI is employed, there is usually an assumption that the software is based on message passing as opposed to shared memory. This is one limitation that we anticipate in XtreemOS's Linux-SSI, based on Kerrighed, but the compatibility and performance issues are still of concern.

In addition to scalability from a processing perspective, business applications are also highly data-driven. As the number of users, customisations, searches and transactions increases, the amount of data stored and queried also increases. For this reason, scalable, efficient reads and writes of data are also a critical requirement for business applications. We therefore look into the technology on which XtreemFS is built as an opportunity to develop highly scalable and fast storage management, such that tradeoffs between price and performance can be made. Business applications and data also have long operational lifetimes, and are intended for the purpose of reducing costs and making business processes more efficient. Therefore, the deployment environment for business applications is particularly sensitive to operational costs, otherwise known as the Total Cost of Ownership (TCO). TCO may include energy consumption, such that the operation and deployment of business applications needs to be dynamically adapted during runtime in order to maintain within the limitations of TCO. Figure 4.1 highlights important differences between business and e-science applications.

As an example business application, we consider SAP ERP solutions. SAP ERP is a core product from SAP that aids planning for businesses across the world. SAP ERP can be deployed and consumed on both Windows and Linux platforms. Currently, when a single Dialog Instance has hit maximum load on a system or group of systems, a new instance is created and the user is offloaded, thereby giving them the resources required to complete their task. However, this can lead to a loss of state, forcing the user to repeat some of their work. This is far from an ideal situation. Our intention and claim is not to deploy an entire SAP system on XtreemOS, as various technical issues immediately arise, but to evaluate XtreemOS's features against analytical and experimental results that we have obtained.



IT Execution Layers	E-Science	Business
Organization/ People	Organizational structure has negligible impact on operational constraints	Affiliations, roles, departments and structure impact greatly on operational constraints; change is critical
I/O & Presentation	Infrequent, batch, simple, Response time tolerance	Frequent, conversation-based, customized and control-centric, Response time intolerance
Business/ Application Logic	Applications created for parallelization and independent processing	Many components, component layers and component dependencies – complex control-flow
Storage	Flat files for input and output of large data sets; simple processing	Relational databases => large amounts of additional meta-data and transaction processing
Software	Software may be compiled on submission	Large legacy footprint; software pre-compiled for specific OS and shared libraries
CPU & Networking	Broad, high peaks of utilization	Sharp, sporadic utilization (but also broad...)
Utilities (e.g. electricity)	Costs treated as a separate concern and consequence	Utility costs potentially integrated into resource selection models

Figure 4.1: Difference between business and E-science applications

4.1.2 Testbed Setup

Storage Management Test-bed

Application Scalability and Adaptation Test-bed

We currently have 2 systems running an SSI setup with XtreemOS. The hardware specifications of each are as follows:

- HP Compaq dc7700 Convertible Minitower
- Intel Core 2 Duo 6600 @ 2.4 GHz with 4MB L2 Cache
- 4GB DDR2 RAM
- 250GB hard disk

The virtual machines have the following characteristics:

- ?
- 512MB RAM?
- 5GB hard disk?

To begin with, the setup will be tested on physical hardware, taking the following steps:

- Setup SAP ERP
- Simulate small load
- Increase load to 100% of current resource allocation
- Increase load further and observe balancing
- Decrease load and observe rebalancing

The next test will use the same steps, using 4 virtual machines on each computer and observing the distribution across them.

In doing this, we will be able to show that using SSI allows for simple and efficient scaling of resources according to demand. We may also evaluate the XtreemOS SSI against other SSI solutions such as Open Mosix and OpenSSI.

4.2 Wissenheim

This section describes demonstration activities surrounding Wissenheim.

4.2.1 Overview of Application

Wissenheim is a multi-user interactive 3D platform designed as a learning and leasure environment. Wissenheim is using shared objects in combination with optimistic transactions in a peer-to-peer setup.

4.2.2 Wissenheim on XtreemOS

Wissenheim is designed to run on any standard Linux distribution with X11 and OpenGL support. For distribution the OSS layer of XtreemOS must be available. The demonstration will allow the participants to take a stroll around the virtual worlds of Wissenheim and interact with the various objects and teaching contents available. The main XtreemOS features demonstrated with Wissenheim are the Object Sharing Service along with the XtreemFS file system.

4.2.3 Demonstration Setup

We are currently using two AMD X2/3800+ with 1GB memory and a 100MBit NIC to test Wissenheim on XtreemOS. In order to run the application no special libraries other than X11 and OpenGl are required. There should be no special configuration steps necessary to run Wissenheim within XtreemOS if the needed services are already preconfigured (OSS, XtreemFS). The demonstration will show how an interactive application can be easily distributed using the Object Sharing Service provided by XtreemOS.

4.3 jCAE and Elfipole

This section describes demonstration activities surrounding jCAE and Elfipole.

4.3.1 Overview of Application

Elfipole is physical simulation solver for electromagnetism. It solves Maxwell equations in three dimensions. Its applications are prediction of lightning effect, evaluation of antennas interference and enforcement of objects stealth. It has been chosen to test XtreemOS because its architecture and needed resources are representative of other solvers used in EADS. It is distributed using MPI and run on Linux or Unix clusters. To meet industrial IT requirements, Elfipole cannot rely on any libraries, so every dependencies are always included.

jCAE is a meshing tool which aims at creating input model for Elfipole and other solvers. Its main components are a GUI platform and a mesher. The GUI is used to display the 3D models which could be CAD or mesh. The mesher, named Amibe, is a command line application which can be executed locally or distributed. The Amibe command line allows to distribute the execution through any middleware (ex: ssh, Condor). jCAE is a free as in speech software. It's mainly java based but depends on native libraries.

4.3.2 jCAE and Elfipole on XtreemOS

Elfipole currently run on Linux cluster and most proprietary Unixes in both 32 and 64 bits. In practice, the performance bottlenecks are always on disk and network bandwidth. By running Elfipole on XtreemOS LinuxSSI we expect to have a more dynamic and flexible scheduling than on a common cluster.

jCAE is run on a cluster with PBS or LFS and Lustre or NFS. It's also running on workstation LAN with ssh and scp. Some tests have also been done with Condor. To do a distributed mesh with Amibe we need to distribute the CAD on each meshing node and gather the output piece of mesh on the user workstation. The meshing process it self is CPU intensive so we look for having a maximum of nodes. So this increase the time of data distribution. The data to distribute both include some large files and numerous small files. So jCAE is a good application test for XtreemFS. As jCAE does not include its own mechanism to start each elementary meshing process it will also benefits from the AEM services. We expect XtreemOS to improve the scalability and flexibility over the current solutions.

4.3.3 Demonstration Setup

We currently have XtreemOS installed on a four nodes vmware cluster set up on one physical server. The characteristic of the server are:

- 4 Xeon 5160 3Ghz (64 bit support)
- 8Gb of memory
- 300Gb of disk

The virtual machines have the following caracteristics:

- 512Mb of memory
- 5Gb of disk
- 1 Xeon 5160 3GHz

We will use the same cluster for both Elfipole and jCAE. For Elfipole the configuration steps are the following:

- Set up LinuxSSI on each nodes
- Unzip Elfipole
- Prepare an Elfipole project
- Start LinuxSSI

And for jCAE:

- Setup AEM services and XtreemFS on each nodes (be sure that LinuxSSI is stopped).
- Unzip jCAE on the XtreemFS file system
- Copy a CAD file on the XtreemFS file system
- Write JSDL job for Amibe
- Run the job with xsub command

With Elfipole we will validate that LinuxSSI support all system call required to run a typical HPC MPI application and that its scheduling is efficient. With jCAE we will validate that performances of XtreemFS are acceptable and that it support concurrent access on files. We will also partially test the AEM service (simple job submission, job monitoring, hardware discovery).

4.4 IMA and JobMA: Demonstrating the mobile flavour

One of the main distinguishing features of XtreemOS, and possibly the one with most appeal for non-expert users, is the fact that it supports multiple platforms, including **mobile devices** such as PDAs or smartphones. Two of the reference applications in WP4.2 (IMA – Instant Messaging Application, and JobMA – Job Management Application) were specially chosen in order to test this mobile flavour of XtreemOS, and its integration with graphical end user applications, including legacy applications.

In this section, a demonstration scenario that makes use of both applications and exploits many of the functionalities of XtreemOS-MD is presented.

4.4.1 Rationale and Goals

Many of today's grid applications are job-oriented, non-interactive distributed applications with little interaction on the part of end users. XtreemOS however, with its OS-like approach and its use of widespread application interfaces (POSIX, SAGA), opens the door to other kinds of applications, specially end user ones (e.g. graphical desktop applications). Moreover, with the inclusion of a mobile flavour of the grid-enabled operating system, a whole new range of possibilities in the mass market also become available.

Thus, two of the reference applications chosen in XtreemOS try to explore these possibilities. One of them is a slight modification of a popular instant messaging (IM) client, which will be used to evaluate the **ease of integration** of XtreemOS with existing applications, as well as its **transparency** and **usability**. The other one is a simple graphical frontend to the execution management system of XtreemOS, and will be used to evaluate the **ease of application development** for XtreemOS, as well as the overall **efficiency** and usability of the mobile flavour.

Apart from the reference applications themselves, this scenario should also make use of other XtreemOS-MD features that could set it apart from other grid (specially mobile grid) systems, like the **ease of installation** from scratch, or the **single sign-on** framework for usage of certificates by applications.

4.4.2 IMA and JobMA

IMA – Instant Messaging Application

The Instant Messaging Application (IMA) is a modification of the popular Pidgin IM client (see ⋆http://pidgin.im) to operate within a XtreemOS virtual organization, by using the user certificates provided by the VO, but also by using other XtreemOS services like XtreemFS to store application data (e.g. conversation history, preferences etc) in a secure and reliable manner, thus liberating the mobile device from this task.

The main objective in this case is to evaluate the amount of modifications to the code, and the development effort needed to achieve this integration into an already existing (*legacy*) application. Thus, the modifications to the code should be minimal (in fact, the current state of the application requires no changes to the source code, just to the packaging of the application).

JobMA – Job Management Application

The Job Management Application (JobMA) is a simple application developed from scratch for XtreemOS-MD, and which basically acts as a mobile graphical frontend to the Application Execution Management (AEM) system of XtreemOS. It allows mobile users to launch and manage grid jobs, including pausing jobs, cancelling jobs, restarting jobs etc. Naturally, this application also uses other XtreemOS features under the hood, such as the obtention and management of user certificates for the virtual organization, as well as accessing XtreemFS to obtain application data and preferences.

The objective of this application whas to evaluate the difficulty of developing new applications (specially graphical end user applications) that take advantage of XtreemOS services.

4.4.3 User Story for the Demo

> *Our hypothetical user, Bob, has just started working for an aerospace simulation SME (ASE from now on). ASE is part of a group of companies and universities which have decided to share their resources for doing simulations of various kinds (unsurprinsingly called "Simulation VO").*
>
> *When Bob joins ASE, he is registered by his boss as a user of that VO, and he receives a user name and passphrase to obtain certificates.*
>
> *On his first week at ASE, Bob has to visit some customers to present the results of the latest simulations and decide next steps*
>
> *Bob also receives from the company a brand new Internet Tablet and the URL of ASE's XtreemOS package repository.*
>
> *On his way to the airport, he decides to work out some details with his workmate Alice...*

After this introduction, the demo itself would begin (see next section).

4.4.4 The Demo Step-by-step

1. Bob uses the web browser to install XtreemOS-MD grid-enabling software from the package repository. Bob fires up Pidgin-IMA and talks with Alice, which is also away on a business trip (Bob will be asked for user/password to get the certificates from the CDA).

2. While talking, they discover that some needed simulations have not been run.

3. Bob fires up JobMA (no user/password is asked this time – single sign-on) and launches the jobs just as he is about to enter the flight.

4. When Bob arrives to the destination, the job has finished and Bob accesses the data (e.g. some large video files) which is stored in the grid (XtreemFS).

5. Bob can continue preparing the meeting...

4.4.5 Demonstration Setup

The hardware and software needed to demonstrate this scenario would ideally include a XtreemOS testbed operating at least one VO, and distributed across geographically distributed sites. However, logistics often preclude this kind of testbed to be accessed, and this is why we will describe a **minimal setup** (a barebones, standalone testbed) and an **optional setup** (ideal, distributed testbed).

Minimal Setup

The bare minimum setup for this demo would include:

- At least one mobile device (e.g. a Nokia N800 tablet) running Maemo and which, at the beginning of the demo, will have NO XtreemOS-related software whatsoever.

- A PC (e.g. a laptop) running XtreemOS and all the necessary core and resource node services to enact a VO:

  - VO and security services, specially a CDA, VOLifeCycle, RCA etc
  - XtreemFS services, including MRC, OSD and DIR
  - Application Execution (AEM) services, in the form of a XOSd with all needed services activated

  This PC would also need to have installed a web server (e.g. Apache) with a Maemo package repository containing the XtreemOS-MD software.

- Access to an instant messaging server (e.g. a XMPP/Jabber server) for all the participants in the demo. This server can be a publicly accesible one or it can also be hosted in the demo PC.

Optional Setup

If a geographically distributed testbed is available for demos, the setup should include:

- A mobile device with the same characteristics as the one described above.
- A fully operational VO distributed across the testbed, with all the core services and one or more resource nodes.
- A web server containing the Maemo packages, as described above.
- An instant messaging server similar to the one outlined above.

4.5 Summary of Planned Demonstrations

A set of more ambitious demonstrations have been defined. These demonstrations are subject to change according to the feedback received from XtreemOS and the prioritisation of features. The goals of these demonstrators are to further evaluate features of XtreemOS but to also have a means of demonstrating the profitability, productivity and sustainability properties of XtreemOS. It is expected that these demonstrators will be critical for making the case for XtreemOS clear. Although XtreemOS is currently in its prototypical stages (and likely to remain in that stage given the timelines for IT productisation) its benefits must be made clear.

Chapter 5

Conclusion

The deliverable provides the updated and extended set of application requirements which have been discussed in cooperation with the development work packages. The revision was driven by the ongoing design and development activities, the evolution of the project vision and also by the deeper insights into the interplay between XtreemOS and the application references. The developers of XtreemOS need to be aware of the entire set of requirements as many requirements are interdependent thereby affecting multiple development work packages. The requirements serve as a source of inspiration and as a guideline. However, the distribution of responsibilities and the coordination of work regarding the implementation of the required functionalities belong to the responsibilities of the project and technical management as well as the work package leaders in SP2/SP3. During the upcoming phases of the project, the requirements will continuously be revised and extended if necessary, driven, e.g., by evaluation results and by the activities in SP2 and SP3.

The second main contribution of this deliverable is the evaluation of the first official XtreemOS release as well as individual components comprising [include list of components]. The deliverable provides the entire test documentation based on the IEEE standard 829-1998 including test plans, specification, logs and results, which allows to easily access relevant information and to retrace test procedures. Test results were summarized to give a first overview of the fulfillment status of the application requirements.

The evaluation reports and in particular the results of the evaluation are being communicated through various channels, including wiki pages, mailing lists, bug trackers, phone calls and finally this deliverable. All intermediate and final test documents are made available on the internal SVN server (`htps://scm.gforge.inria.fr/svn/xtreemos-deliv/WP4.2/experiments`). We hope that the evaluation provides useful feedback to the development work packages and we are looking forward to a successful continuation of the cooperation established.

The description of demonstrations describes where we see potential for XtreemOS being further applied and used in practice, given the state of affairs. These demonstrations are subject to change over time, but represent what we see as feasible in the near future given our evaluation of XtreemOS. This therefore gives an indicator of how the development of technologies is progressing internally and we hope that there can be continued collaboration towards making these demonstrators a reality.

Chapter 6

Acknowledgments

We would like to thank the developers in SP2 and SP3 and in particular the respective work package leaders for their cooperation regarding the identification of interesting test scenarios and for their support during the installation XtreemOS components. Furthermore, they provided useful feedback inspiring the revision of requirements.

Bibliography

[1] XtreemOS consortium. Early experiments and evaluation. XtreemOS deliverable D4.2.2, 2006.

[2] XtreemOS consortium. Requirements capture and use case scenarios. XtreemOS deliverable D4.2.1, URL: http://www.xtreemos.eu/publications/project-deliverables/d4-2-1_main.pdf, 2006.

[3] XtreemOS consortium. Application references, requirements, use cases and experiments. XtreemOS deliverable D4.2.3, 2007.

[4] XtreemOS consortium. Design and implementation of node-level vo support. XtreemOS deliverable D2.1.2, 2007.

[5] XtreemOS consortium. Application references, requirements, use cases and experiments. XtreemOS deliverable D4.2.4, 2008.

[6] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150, 2001.

[7] Linux Foundation. Linux standard base homepage. www.linuxfoundation.org/en/LSB, 2008.

[8] IEEE. Ieee standard for software test documentation, ieee 829-1998. IEEE Computer Society, 1998.

Appendix A

Requirements

In this appendix, we present the updated and extended set of requirements. The changes and extensions of requirements were explained in greater detail in Chapter 2. Due the number of extensions modifications performed on the requirements, it was decided that deliverable D4.2.5 should provide more than a list of changes such that the partners in the consortium have access to the entire updated requirements catalogue as a whole with no need to cross-reference deliverables D4.2.1, D4.2.3 or D4.2.4.

As before, the requirements are maintained according to the following structure:

- General requirements
- Virtual organization support in XtreemOS
- Checkpointing and restart
- Federation management
- XtreemOS interfaces
- Highly available and scalable Grid services
- Data management
- Security in virtual organizations
- Support for mobiles devices

Although this structure suggests a close relation to the organization of work packages, the requirements are strongly inter-related and inter-dependent. Therefore, the members of SP2 and SP3 are asked to read all the requirements to asses the relevance for their own work. The requirements are presented by their requirement number (Rx), a requirement title and a corresponding detailed description. Furthermore, the following fields provide additional information:

Previous number (D4.2.4): In order to maintain a continuous numbering within this document, some requirement numbers are changed due to the insertion of one additional requirement. Therefore we provide a reference to the requirement number used in deliverable D4.2.4.

Importance: "Obligatory" indicates that the respective requirement must be implemented to ensure that the applications can be executed, whereas "Optional" means that the specific requirement would increase the usability of XtreemOS. However, in the "Optional" case, the applications can still be executed without the requested additional functionality.

Updated: This field signifies whether the title, description or importance of the requirement has been changed in comparison to Deliverable D4.2.4.

Importance for MDs: This field, if given, indicates whether the importance or relevance of the respective requirement is different for the mobile devices (MDs) flavor of XtreemOS.

Implementation order: On a scale between 1 (this functionality has to be provided very early) and 10 (may be implemented during later phases of the project) the implementation order (average over all responses received) proposes a guideline in which phases of the project lifetime the requested functionality should be provided.

A.1 General Requirements

R1: XtreemOS equally supports data-intensive and computation-intensive applications

Most of the applications considered are either data-intensive or computation-intensive. Some of them (SPECWEB, SIMEON) are both whereas SRC, IMA and JOBMA are neither data nor computation-intensive. Regarding data-intensive applications it is essential that XtreemOS can efficiently manage access to central and distributed databases and can also handle file-based applications (cf. requirements for data management).

Previous number (D4.2.3): R1
Updated: no
Importance: obligatory
Importance for MDs: optional
Implementation order: 2.9

R2: XtreemOS supports heterogeneous hardware

The XtreemOS system shall run on heterogeneous nodes with different architectures and different memory & storage capacities as visualized in Figure A.1. Some applications are programmed to be platform-independent and do therefore support heterogeneous architectures. Certain applications like ELFIPOLE or ZEPHYR benefit from homogeneous architectures to facilitate runtime prediction. Within a single federation, heterogeneity that must be supported includes different CPUs and different amount of memory, processor clockspeed etc.

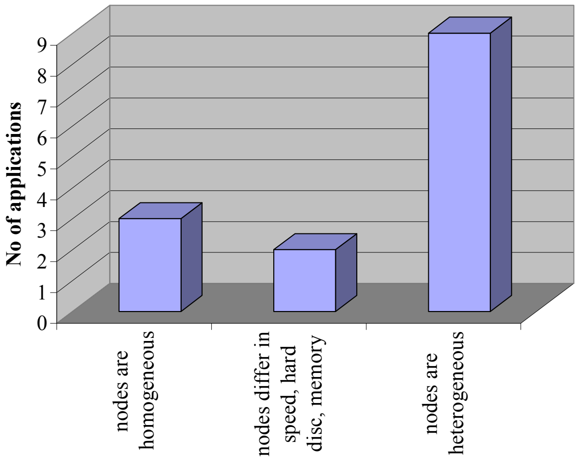


Figure A.1: Hardware diversity

Previous number (D4.2.3): R2
Updated: no
Importance: obligatory
Importance for MDs: Only ARM architecture
Implementation order: 10.0

R3: XtreemOS must support Grids with variable number of nodes

The XtreemOS system must be able to handle from a few up to more than thousand of computing resources.

Figure A.2 demonstrates that 5 applications require relatively "small" Grids whereas the majority of applications may also demand Grids with up to thousands of nodes.

Previous number (D4.2.3): R3
Updated: no
Importance: obligatory
Implementation order: 3.5

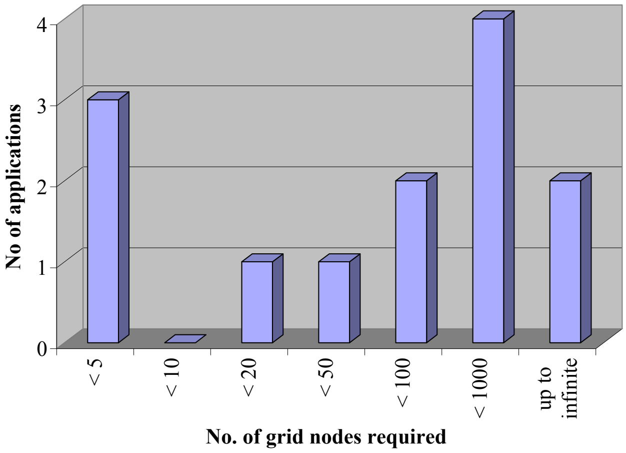


Figure A.2: Number of Grid nodes required by the applications

R4: XtreemOS needs to handle virtual (replicated) nodes in order to increase robustness in case of failures

XtreemOS should provide two ways to increase robustness: a checkpoint/restart mechanism, as defined in deliverables D2.1.1 and D2.2.1, and a replication mechanism termed virtual nodes, as defined in D3.2.1.

Using virtual nodes, a service, application or their individual parts whose robustness and availability is critical is in fact being run replicated on multiple nodes. This is seen by the application as a single "virtual" node. On node or link failure, another copy can take its place immediately, which makes this mechanism suitable for interactive applications and for services whose availability is critical. Resource usage is multiplied by the number of replicas.

Four applications require and benefit from virtual nodes. Typically, the number of required replicas is low (1 or 2) whereas DBE can benefit from as many replicas as possible.

See also R46 for more details on how to extend this mechanism to the federation level.

Mechanisms/suggestions: the application specifies which processes are critical and therefore require replication. The number of replicas can be specified by the application as well. Alternatively, a mechanism could allow specifying the desired robustness, after which the system would choose a suitable number of replicas, taking into account the estimated robustness of each node.

Previous number (D4.2.3): R4
Updated: no

Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 5.7

R5: XtreemOS must support that nodes can dynamically be added to the Grid and removed from the Grid

During runtime XtreemOS shall be able to dynamically adapt to the available amount of resources. It must be possible to add new nodes to the Grid. Furthermore, the scheduling system must consider these additional resources. If nodes are removed from the Grid (for various reasons) XtreemOS must handle the migration of running applications (see R6). This capability to dynamically react to a changing number of Grid nodes is required by about half of the applications.

Previous number (D4.2.3): R5
Updated: no
Importance: obligatory
Implementation order: 3.7

R6: XtreemOS must support migration of running applications

XtreemOS must support the migration of running applications. The migration functionality must be customizable allowing e.g. to specify the maximum time to complete this migration or to explicitly allow/deny migrations.

Previous number (D4.2.3): R6
Updated: no
Importance: obligatory
Importance for MDs: optional
Implementation order: 3.7 (corresponds to R5)

R7: Execution and Migration of Java Applications

It must be possible to at least migrate an individual JVM and the Java application running in it as a whole.

Previous number (D4.2.3): R7
Updated: no
Importance: obligatory
Importance for MDs: not applicable
Implementation order: to be determined

R8: XtreemOS must support Grid nodes with up to several Terabytes storage capacity

The storage requirements range from only a few MBs up to more than 500 GB. However the actual amount of storage required depends on the size of input/output data and the number of nodes. Furthermore WEBAS requires that Grid nodes containing the database can store up to 5 TB of data.

Previous number (D4.2.3): R8
Updated: no
Importance: obligatory
Implementation order: 8.1

R9: XtreemOS must support Grid nodes with up to several Gigabytes working memory capacity

The working memory requirements vary between a few MBs and several GBs. However the actual amount of memory required depends on the size of input/output data and the number of nodes.

Previous number (D4.2.3): R9
Updated: no
Importance: obligatory
Implementation order: 4.6

R10: XtreemOS needs to provide for software licensing mechanisms

Several applications require local license files or license servers for their execution. Accordingly, XtreemOS must provide for the distribution of license files to the respective Grid nodes (respectively connectivity to license servers) taking into account that license files may be bound to specific MAC/IP-addresses or dongles. It must also be ensured that licensing software is installed in the Grid nodes prior to starting the applications. Overall, this requirement demands that the scheduling mechanism must incorporate the connectivity between nodes.

Previous number (D4.2.3): R10
Updated: no
Importance: obligatory
Implementation order: 6.0

R11: Packaging of XtreemOS releases

In order to increase the acceptance of XtreemOS in the users and developers community it is required that all intermediate and the final releases of XtreemOS are distributed as self-installing software packages including but not restricted to the rpm and deb format. These packages must be compatible with at least the following Linux distributions: Mandriva, Red Flag, and Debian.

Previous number (D4.2.3): R11
Updated: no
Importance: obligatory
Implementation order: to be determined

R12: XtreemOS must provide for fast and reliable communication

The performance of applications relies very much on fast and reliable network connections. Therefore, XtreemOS has to ensure that the available network resources are managed and used efficiently. Especially data-intensive applications require high-speed connection to transfer large data sets or to process frequent database transactions. Several applications demand a permanent connectivity between nodes. Reliable connectivity is also a crucial requirement by various applications, which may be tremendously affected by network failures (see Figure A.3).

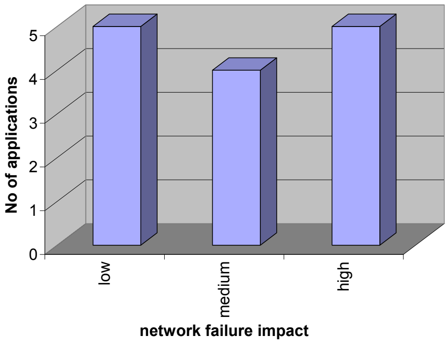


Figure A.3: Expected impact of network failures

When mapping applications onto Grid resources XtreemOS also must take into account the respective network characteristics. It should therefore be possible that applications can define certain parameters like maximum network delay, minimum bandwidth and reliability. These parameters need to be considered by XtreemOS when allocating and re-allocating applications.

Previous number (D4.2.3): R12
Updated: no
Importance: obligatory
Implementation order: 3.4

R13: XtreemOS must be compatible with highspeed interconnect standards

XtreemOS must support highspeed communication including Infiniband, Myrinet and SCI. Regarding the SSI flavour of XtreemOS, support for Infiniband is obligatory.

Previous number (D4.2.3): R13
Updated: no
Importance: obligatory

Implementation order: to be determined

R14: XtreemOS must support IPv6

The XtreemOS must support IPv4 and IPv6.

Previous number (D4.2.3): R14
Updated: no
Importance: obligatory
Implementation order: 7.5

R15: Semaphores

XtreemOS must support semaphores, in particular System 5 semaphores are required.

Previous number (D4.2.3): R15
Updated: no
Importance: obligatory
Implementation order: To be determined

R16: XtreemOS must support 64 bit architectures

Since almost no pure 32 bit systems are sold anymore in today's HPC market. Furthermore several upcoming releases of applications will only be executable on 64-bit processors. Therefore it is essential that XtreemOS supports x86_64 and Power PC architectures.

Previous number (D4.2.3): R16
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable (No MD currently uses 64 bit architecture)
Implementation order: 2.4

R17: SMP and multicore support

XtreemOS must support SMP and multicore architectures.

Previous number (D4.2.3): R17
Updated: no
Importance: obligatory
Implementation order: to be determined

R18: XtreemOS must support virtual machines (e.g. XEN, VMWare, OpenVZ/ Virtuozzo)

This requirement is actually twofold.

1. XtreemOS should be able to run inside a virtual machine.
2. XtreemOS should be able to run virtual machines (i.e. act as the host operating system).

The second point is important for business applications with strict isolation requirements between different VOs. This is especially true for the case of multiple VOs sharing the same physical hardware. For such scenarios with strong isolation requirements it must therefore be possible to execute applications in VMs or virtual containers/compartments, and a VO is created across a group of selected VMs/containers.

Remark: This requirement is closely related to the other security requirements in this document.

Previous number (D4.2.3): R18
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 6.0

R19: XtreemOS shall support multicast

It is required that XtreemOS supports multicast within a cluster.

Previous number (D4.2.3): R19
Updated: no
Importance: optional
Implementation order: 8.0

R20: XtreemOS needs to provide access to various Grid services

The main Grid services required by the applications are file transfer, resource discovery and job submission, as visualized in Figure A.4.

Further services required are related to scheduling and monitoring jobs as well as viewing and modifying user information, in particular user location, presence information etc. Various applications also offer Grid/web services, e.g., resource scheduling or resource and file management, instant messaging or job monitoring. Therefore it must ensured that these services can be accessed and executed. Services and resources are discovered mainly by Uniform Resource Identifiers (URI), full description or by an approximate/semantic description (cf. Figure A.5).

Previous number (D4.2.3): R20
Updated: no

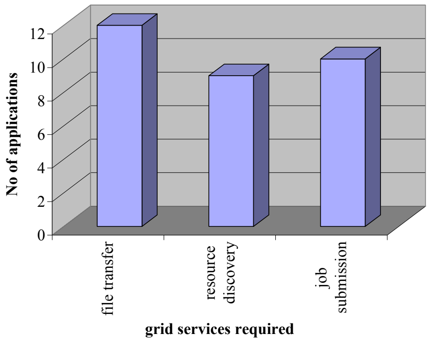


Figure A.4: Main Grid services required

Importance: obligatory
Implementation order: 2.8

R21: XtreemOS VOs shall manage a large number of users

The number of users within a VO is very different for the applications considered. About half of the applications can be executed with at most 4 different users whereas the other half demands to manage several thousands of users, which may also concurrently work with the application.

Previous number (D4.2.3): R21
Updated: no
Importance: obligatory
Implementation order: 4.7

R22: XtreemOS must support the execution of interactive and batch jobs

The majority of applications in WP4.2 is executed interactively as shown in Figure A.6. Some of them are purely interactive programs whereas others are combinations of interactive and automated steps. Several applications may be started in either interactive or batch mode. Five applications are typically run in batch mode. Consequently, XtreemOS must be able to facilitate the distribution and execution of both, interactive applications and batch jobs.

Previous number (D4.2.3): R22
Updated: no
Importance: obligatory
Importance for MDs: optional
Implementation order: 2.2

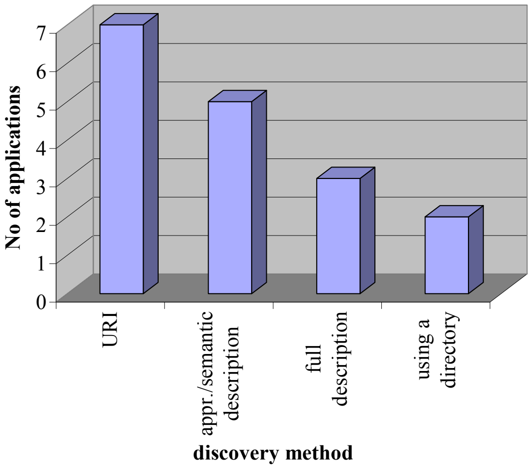


Figure A.5: Methods for resource and service discovery

R23: XtreemOS VOs must provide role management

XtreemOS must be able to manage users through roles – each role having its own rights. Users must be able to read, write, change and delete files and to execute applications. Administrators need the permission for installation, maintenance and to manage accounts (add/remove accounts, change of permissions). Furthermore, it must be possible to execute different parts of the applications, e.g. the database in a multi-tier business application stack, in the context of different users and different groups.

Remark: Middleware systems like WEBAS also require a complex role management at the application layer.

Previous number (D4.2.3): R23
Updated: no
Importance: obligatory
Implementation order: 7.2

A.2 Virtual Organization Support in XtreemOS

R24: XtreemOS has to provide the means to manage VOs

VOs must be manageable. Management actions include the creation, change and destruction of a VO. Three different roles will be involved in managing VOs: domain administrators, VO administrators and VO users. Domain administrators

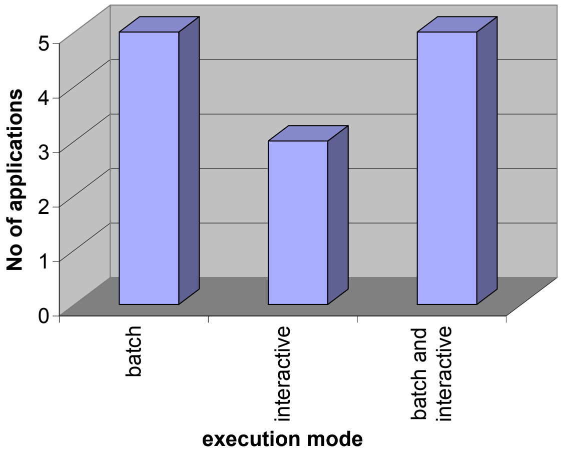


Figure A.6: Application execution modes

maintain a pool of resources that are allowed to be integrated into a VO. VO administrators are allowed to compose VOs from the resources provided by various domains. They are also responsible for creating VO user accounts and for configuring and modifying the permissions that VO users have within a VO. Participating institutions agree on one or more persons to be VO administrators. Therefore it must also be possible that each participating institution has at least one person that acts as VO administrator. VO users also require the right to manage a VO. The respective (restricted) permissions are granted by the VO administrator. This allows for instance that a VO is created when an application is started and the VO is destroyed after the termination of the application.

Previous number (D4.2.3): R24
Updated: no
Importance: obligatory
Implementation order: 2.5

R25: VO user accounts have to be independent from local user accounts

A person can have a user account in her/his local domain A and a VO user account in a VO comprising domains B, C and D. To obtain a VO user account it is not necessary to have a local user account in one of the domains belonging to the VO. The VO user need not have a local account anywhere in the Grid at all. However, it must be possible to transfer data, files and directories between local and VO accounts (e.g., by copy or mount).

Previous number (D4.2.3): R25

Updated: no
Importance: obligatory
Implementation order: 5.2

R26: XtreemOS allows to dynamically change the composition of VOs during application runtime

It must be possible to change the composition of resources within VOs while applications are executed. This dynamic adaptability is needed e.g. if certain computing or communication resources fail. In this case, the unavailable resources need to be automatically substituted by alternative resources (if available). This alternative resource is chosen from the pool of available resources as agreed (by contract) among the participating institutions.

Previous number (D4.2.3): R26
Updated: no
Importance: obligatory
Implementation order: 7.4

R27: The lifetime of a VO must be guaranteed

The lifetime of a VO (as required by the applications) may range from a couple of days up to infinite (i.e. not known in advance). Therefore it must be possible to a) guarantee the lifetime of a VO for a specified amount of time and to b) guarantee the lifetime of a VO until a notification that the VO is not required anymore.

Previous number (D4.2.3): R27
Updated: no
Importance: obligatory
Implementation order: 5.2

R28: XtreemOS must allow to establish multiple VOs on the same node within specified constraints

VOs comprise nodes from different domains. An overlapping of VOs is allowed. Therefore, it must be possible that a node is contained in multiple VOs. The domain administrator, however, is allowed to restrict the maximum number of VOs to which a certain node can belong to. One special - but also very important - case is that a VO comprises only a single node. Therefore it must also be possible that XtreemOS is executed on a single node with multiple concurrent VOs established. Consider isolation mechanisms within and between VOs (cf. WP3.5)

Previous number (D4.2.3): R28
Updated: no
Importance: obligatory
Implementation order: 7.0

R29: A VO management interface has to be provided

The management of VOs must be possible by an API (Application Programming Interface), a CLI (Command Line Interface) and a GUI (Graphical User Interface). The management interface must also provide means for monitoring the VO, e.g. information on the effectiveness of the changes made on VOs. The VO management service must be transaction aware to allow a consistent resume e.g after node failure.

Previous number (D4.2.3): R29
Updated: no
Importance: obligatory
Importance for MDs: optional
Implementation order: 4.6

R30: VO management actions must be completed within a specified maximum amount of time

Various applications require that a VO can be created, changed and destroyed within a certain maximum amount of time. In some cases (SIMEON, WISS) this response time needs to be a couple of seconds or at most 10 minutes. It is therefore necessary that VO users can in advance define how fast management actions need to be performed with respect to each application (to be agreed with the VO administrator).

Previous number (D4.2.3): R30
Updated: no
Importance: obligatory
Importance for MDs: optional
Implementation order: 7.3

R31: XtreemOS has to support communication between VOs

The applications require exchange of information between VOs by means of messages (also instant messages), shared memory and data transfer.

Previous number (D4.2.3): R31
Updated: no
Importance: obligatory
Implementation order: 7.1

A.3 Checkpointing and Restart

XtreemOS should provide two ways to increase robustness: a checkpoint/restart mechanism, as defined in deliverables D2.1.1 and D2.2.1, and a replication mechanism termed virtual nodes, as defined in D3.2.1.

A checkpoint made previously is used to restart an application that failed (either by its own fault or for other reasons). This mechanism does not need any additional resources, apart from the disk or memory space required for the checkpoint. It is appropriate for applications and services that either run isolated from the environment, such as off-line computations, or can adapt to changes in the environment. The failed application is available again after a certain amount of time.

R32: XtreemOS must support automatic failure detection, checkpointing and restart

XtreemOS must provide automatic failure detection (e.g. computer went down or network broke), checkpointing and restart at the system level preferably with no need to modify the application. *Mechanisms/suggestions* Restarts should be done from the last checkpoint, unless the user specifically requests the use of an older checkpoint. Alternatively, if the application fails a few times in a row from the last checkpoint, an older checkpoint can be tried automatically.

Previous number (D4.2.3): R32
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 4.7

R33: Restart must mimic the original environment

XtreemOS must provide the illusion of the original environment considering in particular IP addresses, hostnames and PID numbers. Accordingly, a mechanism for handling virtual IP addresses, hostnames and PIDs must be provided. If the restart takes place on a different VO resource, it must be ensured that the necessary security objectives are achieved during communication.

Previous number (D4.2.3): R33
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 5.7

R34: XtreemOS must be able to notify the application of checkpointing and restart

XtreemOS has to notify the running application prior to checkpointing and interruption such that the application has the opportunity to react appropriately (e.g. store data, close open files, send messages ...). Furthermore, XtreemOS must notify a restarted application of the changed execution environment, including - but not limited to - IP addresses, hostnames and PIDs.

Previous number (D4.2.3): R34
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 4.6

R35: XtreemOS has to support various ways of checkpoint initiation

It is required that XtreemOS provides the mechanisms for creating and storing sequences of checkpoints. The checkpointing mechanism needs to be configurable to support:

- Checkpointing initiated by the application independent from the OS
- Checkpointing initiated by the application to stable storage provided by the OS
- Checkpointing initiated by the OS at application's request
- Automatic checkpointing initiated by the OS (with and without notification)

A respective API has to be provided.

Previous number (D4.2.3): R35
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 7.4

R36: Checkpointing/restart performance

Checkpointing must be fast enough that the required checkpointing frequency will not represent a significant load to the system. Restart from a saved checkpoint must take less than the time between successive checkpoints, i.e. for one checkpoint each 20 minutes, at most 20 minutes should pass from failure to the moment when the application is up and running again. *Quantification:* One application has extremely high demands: at least one checkpoint per 30 seconds. The estimated amount of working memory per node for this application is 512 MB, but the size of the data requiring checkpointing is much smaller. The checkpointing frequency must be high for online application to allow restarting quickly after failure and without loosing much data; otherwise the user experience will suffer.

Other applications demand at least one checkpoint per hour. Their memory requirements are specified as up to a few GB.

Previous number (D4.2.3): R36
Updated: no
Importance: obligatory

Importance for MDs: Not Applicable
Implementation order: 7.0

R37: Checkpointing/restart must be implemented in the kernel space

Checkpointing and restart must be implemented on OS level in the kernel space.

Previous number (D4.2.3): R37
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 6.4

R38: XtreemOS allows to customize checkpointing and restart

It is required that XtreemOS allows the applications to specify on which nodes a checkpointed application will be restarted. This is important e.g. for applications that require user interaction. Restart from last and older checkpoints must be customizable to support:

- automatic restart
- restart with user-interaction

XtreemOS must allow the user to specify how often (or when) checkpoints are created. XtreemOS must allow to activate/deactivate checkpointing and automatic restart during application runtime with no need to reboot nodes.

Previous number (D4.2.3): R38
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 7.1

R39: Information on the process state that must be saved/restored during checkpointing/restart

Checkpointing and restart must save/restore the following information on the process state (customizable by the user/application):

- Threads
- IPC
- Network communication (in particular open MPI communication)
- Open files (contents must be saved)
- Registers

- Caches

Optionally, linked libraries should also be saved.

Mechanisms/suggestions: The automatic restart with the same threads etc may use the mechanisms already implemented in Kerrighed. The involved restart can perhaps use similar mechanisms to those that will be used for normal application startup, with a flag signalizing that this is a restart. Saving contents of large open files is not realistic. If the files are not open for very long, the checkpoint can be delayed automatically for a few seconds, hoping that there will be a moment when no large file will be open. Otherwise, the applications and the OS will have to cooperate on this issue.

Handling of IPC communication at restart: Checkpoint/restart mechanisms in the kernel should be able to save and restore the IPC state, but the decision may be left to the higher level mechanism that is able to coordinate a checkpoint between all participating processes (e.g. at the mpirun level for MPI). It is only at that level that it is possible to know if the current process restart is part of a coordinated restart or is a standalone restart.

Previous number (D4.2.3): R39
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 6.5

A.4 Federation Management

R40: Number of federation nodes used

It must be possible to specify the number of federation nodes to use because the number of nodes that can be used effectively is application-specific and thus cannot be determined by the system. None of the applications uses a fixed number of nodes (apart from IMA and JOBMA, which use one node only). Figure A.7 gives an overview over the maximum number of nodes required within a federation though it must be considered that various applications could use as many nodes as they can get.

Previous number (D4.2.3): R40
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 2.3

R41: Information available to SSI scheduler

The LinuxSSI scheduler must have access (at least) to:

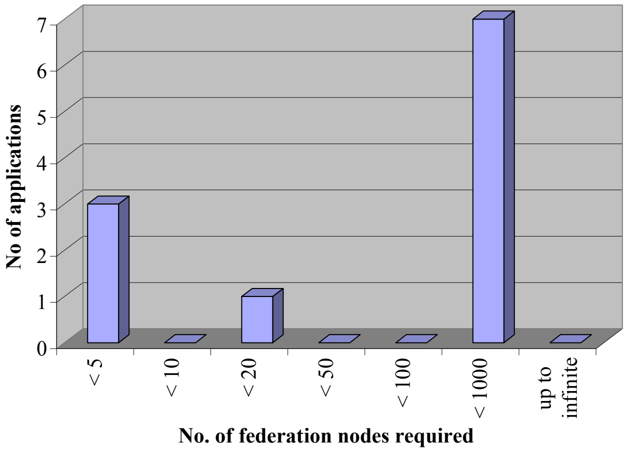


Figure A.7: Maximum number of federation nodes required by the applications

- node-level and cluster-level variables, for example per-node CPU and memory usage,
- application-specified (or process-specified) variables, for example application's requirement to be scheduled only to nodes with certain properties,
- grid-level variables, for example a requirement that two processes must never be scheduled to the same node within the cluster. Such conditions must be coordinated with the grid-level scheduler.

The above will allow the LinuxSSI scheduler to satisfy requirements R42, R43, and R46. It must also be possible to define new variables and new, custom scheduling policies that take them into account.

Previous number (D4.2.3): R41
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: To be determined

R42: Specification of service qualities in federations

It must be possible to specify service qualities (e.g. maximum network delay, availability of resources, throughput) for a certain application. The scheduler must not allocate resources that do not fulfill the defined service qualities to the application.

XtreemOS also allows applications to specify a topology of the resources which provide the best performance.

Previous number (D4.2.3): R42
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 5.3

R43: Node properties constraints in federations

It must be possible to specify some required properties of federation nodes:

1. node architecture (homogeneous, heterogeneous),
2. installed libraries,
3. installed web services and other software.

Most applications depend on some libraries and other software. Furthermore, some applications would require additional effort to be able to distribute parallel execution among heterogeneous nodes. Although XtreemOS could optionally offer facilities to overcome some or all of the above difficulties automatically (which is not a subject of this requirement), such facilities would incur a performance penalty.

The XtreemOS API must provide means to specify the constraints when starting the application. The scheduler must not allocate nodes without the required properties to the application.

Previous number (D4.2.3): R43
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 5.5

R44: Shared file system within a federation

XtreemOS must offer shared file system within federations. Also related to Section A.8.

Previous number (D4.2.3): R44
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 3.9

R45: Changing number of federation nodes

It must be possible to change the number of nodes that the application uses during runtime. If the number of available federation nodes changes, XtreemOS must notify the running applications. The application then decides whether it can adapt to the change.

Mechanisms/suggestions: if the application can adapt to the change, it is its responsibility to rearrange any variables and computations going on. If an application cannot adapt on-the-fly to fewer nodes being available, perhaps it can be checkpointed and restarted on fewer nodes. The notification mechanism can be decided on later.

It must also be possible that the running application requests a change of the number of federation nodes. A running application can request additional nodes to start processes. These additional resources have to be provided by XtreemOS (if nodes are available). Furthermore, a running application may release certain resources after terminating calculations on these nodes. These nodes are then available for the execution of other applications. XtreemOS must be able to dynamically consider these released nodes in resource management and to provide them to other applications.

Previous number (D4.2.3): R45
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 4.4

R46: SSI scheduling of replicated processes

For processes replicated by the virtual nodes mechanism provided by WP3.2 (see also R4, it must be possible to run multiple replicas on the same federation with the restrictions that they are never scheduled onto the same SSI node. The replicated processes are not required to be aware of being executed on a federation, and they do not need to use SSI mechanisms.

Previous number (D4.2.3): R46
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 8.3

R47: Checkpointing and restart

Automatic failure detection, checkpointing and restart must also be supported on federation nodes. The respective requirements are equivalent to those in Section A.3.

Previous number (D4.2.3): R47
Updated: no
Importance: obligatory
Importance for MDs: Not Applicable
Implementation order: 5.3

A.5 XtreemOS Interfaces

R48: Other API Standards as basis for XtreemOS API

XtreemOS API must consider the following standard as a basis: SAGA (especially the subsets DRMAA, GAT). Furthermore, any other standard allowing applications to access user and/or job information is welcome. WP3.1 must ensure that applications can adapt to XtreemOS in a way that complex start and stop procedures can be specified that are used to start/stop the various parts of the overall application. To this end, WP3.1 must ensure that the interfaces are sufficient enough and compatible with WP3.3

Additionally, one application requires the following functionalities that are actually provided by Globus/Globus-related components: GridFTP, Apache Axis, and the GSI public key infrastructure. Here, an equivalent XtreemOS functionality is needed.

Previous number (D4.2.3): R48
Updated: no
Importance: obligatory
Implementation order: 5

R49: XtreemOS must support Linux software with no need for modifications

It must be possible to install and execute Linux applications, in particular legacy applications, with no need for modifications neither to installers nor to application code. The Linux extensions and modifications produced by the XtreemOS project must be exploitable by such applications as far as possible. The fulfillment of this requirement largely leverages the acceptance of XtreemOS in the users (in particular industrial users) and developers community.

Previous number (D4.2.3): R49
Updated: no
Importance: obligatory
Implementation order: to be determined

R50: Demand for POSIX compliance

XtreemOS must provide access to the distributed resources in the Grid from any node using the standard Posix interface. Mandatory access control (ACL) as defined e.g. in POSIX .1e, IEEE 1003.1e/2c (which was withdrawn). It is yet to be clarified which calls have to be provided. Furthermore, it would prove useful if functions for management of processes on remote machines would be part of the XtreemOS API.

Previous number (D4.2.3): R50
Updated: no
Importance: obligatory
Implementation order: 5.3

R51: XtreemOS API language support

XtreemOS must support several different programming languages. The mandatory languages and their priorities are C (high priority), C++ (high priority), Java (high priority) and Fortran 77 (low priority). Fortran 77 can be supported via C bindings. The optional languages are Python (medium priority), Perl (medium priority) and Ada (low priority).

Previous number (D4.2.3): R51
Updated: no
Importance: obligatory
Importance for MDs: In MDs, only Java is obligatory. The rest are optional.
Implementation order: 3.3

R52: Degree of Interoperability

It should be possible to use XtreemOS as a backend for GT4 WS-GRAM.

Previous number (D4.2.3): R52
Updated: no
Importance: optional
Implementation order: 6.4

A.6 Highly Available and Scalable Grid Services

R53: Availability of nodes for a specified time

In order to execute parallel applications on top of XtreemOS, it is necessary to provide for a high availability of allocated nodes during the whole job execution. To this end, a fault-tolerance mechanism termed virtual nodes replicating specified nodes is needed. However, this replication has to be specified explicitly by the user or application.

Previous number (D4.2.3): R53
Updated: no
Importance: obligatory
Implementation order: 3.1

R54: High availability of bandwidth between the allocated nodes during the entire runtime of applications

For certain applications, a minimum bandwidth (user-defined) must be guaranteed to communicate efficiently between the allocated nodes. If a bandwidth guarantee cannot be given as the underlying infrastructure is not able to do that, the system must be able to estimate the available bandwidth with high precision. In this case, the goal is a stable minimum bandwidth with high probability (user-defined) during the whole application runtime.

Two types of applications require this feature :

- MPI applications that simultaneously run on several nodes exchanging messages from one to another. In this kind of application, the algorithm is usually built so that the overall process may be blocked till some messages are received.

- Applications that need a tight and permanent connection between two processes.

Previous number (D4.2.3): R54

Updated: no

Importance: obligatory

Implementation order: 5.0

R55: Grid services which have to be provided

We gathered the priorities (1 = lowest, 10 = highest) of the services that must be implemented. In the following the average, minimum and maximum priorities are given:

Service	Avg.	Min.	Max.
Data management	8.1	1	10
Security	7.9	2	10
Authentification	7.7	5	10
Application Deployment	6.7	1	10
Job Monitoring	6.5	1	10
Resource Scheduling	5.8	1	10
Resource Allocation	5.8	1	9
Resource Monitoring	5.7	1	9
Scalability of the whole grid	5.6	1	9
Resource Exploration	5.2	1	8
Decentralization	5.3	1	10
Network Allocation	5.0	1	10
Network Monitoring	4.9	1	10
SLAs	4.9	1	10
Grid services for mobile devices	4.7	1	8
Services to manage Virtual Organization	4.6	1	11
Migration Services	4.0	1	10

Remarks:

- Priorities are not a scale of importance. The average priorities may be used as a guidance in which order the services could be developed. However, a low average priority may not reflect that some applications expressed a high priority for a certain service, e.g., GSDNA, WEBAS and WISS requiring migration services with high priority. Therefore, the respective minimum and maximum values are also given.

- All the applications expressed a high priority on the authentication and security.

- The data management and deployment issues are also vital for all the applications except IMA and JOBMA. GALEB is also less impacted by the data management needs.

- The priority of every single issue has been evaluated higher than 8 by at least three applications.

In conclusion, we can say that every feature and the corresponding interfaces seem equally important. At the beginning of the project, a particular effort could be made on authentication and security.

Previous number (D4.2.3): R55
Updated: no
Importance: obligatory
Implementation order: 4.8

R56: Measurement criteria which must be provided to estimate the quality of Grid services

We gathered the priorities (1 = lowest, 10 = highest) of measurements which must be provided to estimate the quality of the Grid services. In the following the average, minimum and maximum priorities are given:

QoS Measurement	Avg.	Min.	Max.
Number of nodes failed during runtime	6.8	1	10
Ability to reserve resources in advance	6.2	1	10
Number of timeouts waiting for data	5.9	1	10
Time for Authorisation	5.3	2	10
Time to connect Mobile devices	4.0	1	10
Number of not satisfied SLAs	3.8	1	10
Time for Job-Migration	3.4	1	5
Time for Deployment	3.2	1	8
Time for Negotiation	3.0	1	10
Time for Scheduling	2.6	1	5
Time to set up Virtual Organizations	2.6	1	8
Time for Exploring appropriate resources	2.3	1	4

Remarks:

- Priorities are not a scale of importance. The average priorities may be used as a guidance in which order the measurements could be developed. However, a low average priority may not reflect that some applications expressed a high priority for a certain measurement. Therefore, the respective minimum and maximum values are also given.
- Grid services must be established that enable applications and users to gather those information.
- Additional QoS measurements which have to be provided include the time for exploring Grid users (needed by IMA with priority 8) and the time for collecting job information (needed by JOBMA with priority 10).

Previous number (D4.2.3): R56
Updated: no
Importance: obligatory
Implementation order: 5.3

A.7 Application Execution Management

R57: Monitoring System

The monitoring system must provide information about running applications via a monitoring interface or tool and via a notification service.

Monitoring: Provides the means for monitoring the resource consumption during the whole application execution.

Notification service: It must be possible to notify running applications and users whenever predefined conditions are fulfilled.

An interface has to be provided such that an application or the user can configure the monitoring parameters including:

- The **kind of information** and the level of detail at which this information has to be provided (cf. R53, R54)

- The **means and conditions for notifications:** Half of the applications require a notification by email and the other half of the applications needs a callback function. Therefore both means of notification have to be provided. As discussed with WP3.3, the email notification might affect the performance of the overall system. However, it is part of the research to evaluate such effects at the end of the project. The conditions for asserting a notification comprise changes in the application, environment and about situations when certain stages of the overall application execution phase have been reached.

- Several applications require notifications whenever a job status changes (job started, job finished), a job is migrated (migration started, migration finished), or a certain job threshold is exceeded (job surpassed memory threshold, job surpassed storage threshold). Also refer to R56 and R57. Note that the different applications do not request a global job queue but require the functionality that XtreemOS behaves as if it would have one.

Previous number (D4.2.3): R57
Updated: no
Importance: obligatory
Implementation order: 3.7

R58: Job Dependencies

XtreemOS must provide the possibility to specify dependencies between jobs. Thus, whenever a job is submitted to the system, the user must have the possibility to specify jobs that the new job depends on. These dependencies are used for a comprehensive accounting, monitoring, and for the killing of dependent job sets.

Previous number (D4.2.3): R58
Updated: no
Importance: obligatory
Implementation order: 3

R59: Resource Accounting

It must be possible to record the usage of a resources (supported by the kernel) by a user at any given time. A way to implement / use special cost models must also be provided. Furthermore, the accounting has to provide details at what time or within which time interval a certain amount of resources were used. Thus, the final billing must be able to modify cost models according to different times and overall system states, e.g. a computation during the night is normally cheaper than during the day. Note, that it must be possible to automate the resource accounting for jobs using the job dependencies. Thus, it must be possible to ask for the accounting of a specific job and all its dependent jobs.

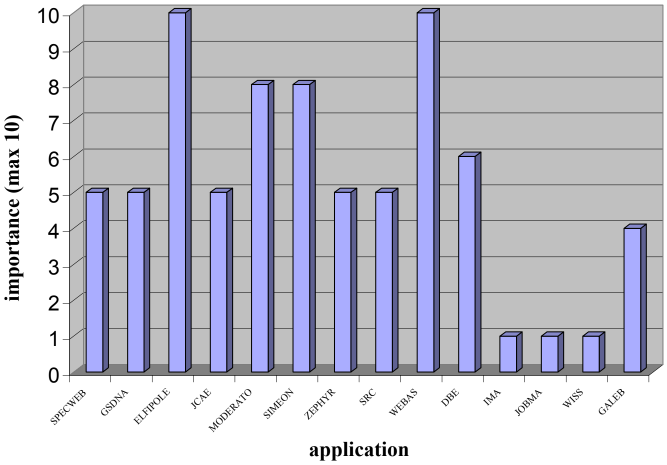


Figure A.8: Rating for resource accounting

Previous number (D4.2.3): R59
Updated: no
Importance: obligatory
Implementation order: 5

R60: The kind of monitoring data that the applications need to query

The monitoring system must provide the following monitoring data to several applications (including dependencies):

- **Job execution:** number of waiting jobs, number of running jobs, number of jobs being migrated, expected time before a job is started, number of jobs earlier in the queue (again, XtreemOS must only provide the functionality as if a global job queue would exist. The applications do not require a global job queue directly.)

- **Job's resource usage:** job CPU usage (in %), total CPU time, memory usage (current/peak/page faults), I/O (bytes read, bytes written, number of reads/writes), threads (number, priorities, start time of each, current/total CPU usage of each thread), number of files currently opened, name of files that are currently opened, file system space currently allocated, number of bytes transferred through the network
- **Job performance metrics:** job's effective file throughput, job's effective network throughput
- **Hardware performance metrics:** job's cache hit rate, job's cache miss rate, job's effective timeslice, jobs priority
- **Host load metrics:** host CPU load, host network load, host file system load

Furthermore, the monitoring system should provide:

- **Host load metrics:** host file system load

Previous number (D4.2.3): R60
Updated: no
Importance: obligatory
Implementation order: 3.7

R61: The kind of information that can be queried from the command line while the application is running

Several of the applications require to gather the following information from the command line during the runtime of the application (including dependencies):

- **Application execution:** number of jobs that an application has submitted, number of jobs that have finished, number of jobs that are running, number of jobs that are waiting (again, XtreemOS must only provide the functionality as if a global job queue would exist. The applications do not require a global job queue directly.)
- **Job execution:** which host is each job assigned to, expected time before a job is started, number of jobs earlier in the queue
- **Job's resource usage:** job CPU usage (in %), total CPU time, memory usage (current/peak/page faults), I/O (bytes read, bytes written, number of reads/writes), threads (number, priorities, start time of each, current/total CPU usage of each thread), number of files currently opened, which files are currently opened, file system space currently allocated, number of bytes transferred through the network

- **Job performance metrics:** job's effective file throughput, job's effective network throughput

- **Host load metrics:** host CPU load, host network load, host file system load

Previous number (D4.2.3): R61
Updated: no
Importance: obligatory
Implementation order: 3.4

R62: Tracing System

Tracing must be provided for both the application execution and the resources being used. Different levels of details are required for both of these, depending on the application, a mechanism should be in place to set these. This requirement only describes the general demand. The specific demand is detailed in the next requirement. Again, the dependencies between jobs must be incorporated.

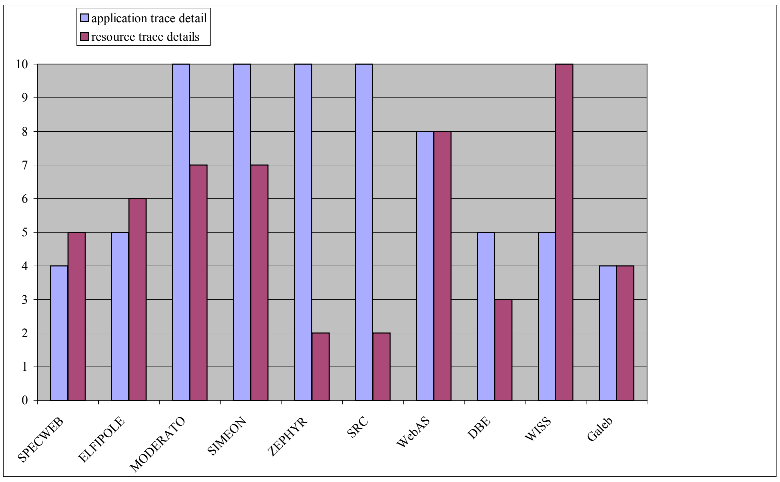


Figure A.9: Rating tracing of application execution vs. tracing of resource

Previous number (D4.2.3): R62
Updated: no
Importance: obligatory
Implementation order: 4.2

R63: Kind of events required to trace from the application's execution

Kind of events required to trace from the application's execution The system must trace the following events. Note that it is not expected that WP3.3 provides all of these functionalities alone but needs to get support from other work packages.

- **Grid scheduling events:** job submitted, job started, job finished, job started migration, job finished migration
- **Job's kernel level events:** system calls, page misses, preemption, timeslice exhaustion, other blocking conditions
- **Resource events:** enqueue job, start job, finish job
- **Others:** application level events triggered using an API designed for that purpose, aggregated information on all/specific events (hardware performance metrics, ...)

Furthermore, the system should trace the event:

- **Communications/parallel library events:** MPI primitives (send, receive, broadcast and variants), OpenMP primitives (enter and exit parallel regions, etc). As the MPI libraries are not directly part of the project, a sophisticated mechanism to add events to the system might be sufficient.

Previous number (D4.2.3): R63
Updated: no
Importance: obligatory
Implementation order: 4.2

R64: Kind of state changes that must be traced from the application's execution

The system must trace the following changes. Note that it is not expected that WP3.3 provides all of these functionalities alone but needs to get support from other work packages.

- **Grid scheduling states:** job waiting, job running, job finished, job migrating
- **Kernel level states:** thread blocked, thread running user space, thread running kernel space
- **Communications/parallel library events:** waiting data, sending data, waiting on synchronization point (barriers)
- **Resource events:** CPU running, CPU idle

Previous number (D4.2.3): R64
Updated: no
Importance: obligatory
Implementation order: 4.3

R65: Trace Format

XtreemOS must provide support for the Paraver format and can additionally develop an XtreemOS-specific trace format.

Previous number (D4.2.3): R65
Updated: no
Importance: obligatory
Implementation order: 5.7

R66: Scheduling

A co-allocation of application on resources of several different sites must be possible. The response times of certain applications of specific customers should be privileged while avoiding starvation of other jobs.

Previous number (D4.2.3): R66
Updated: no
Importance: obligatory
Implementation order: 5.4

R67: Resource Planning

Reservation of resources for specific intervals is necessary and also being able to specify certain characteristics of the required resources (i.e. CPU speed and load, disk space, memory) and to define further constraints regarding the execution of applications (e.g. penalties for late execution, restrictions regarding the choice of nodes, etc.). As XtreemOS is a distributed system, the individual users need information with a minimum accuracy. This accuracy must be defined for individual requests. Furthermore, some applications require changing of resource requirements during runtime. Applications will need detailed information to plan resources in advance and they should be able to confirm resources prior to allocation. Applications need to execute some parts in parallel on different resources (up to 1000 parts). Furthermore, XtreemOS must be able to run a workflow engine on top. Thus, WP3.3 must prove by an example that such a workflow engine can be used to specify startup and stop sequences of whole applications consisting of individual dependent jobs. Furthermore, WP3.3 must ensure that the SAGA interfaces generated in WP3.1 are sufficient to fulfill these requirements. Otherwise, WP3.3 is requested to provide the missing functionality.

Previous number (D4.2.3): R67

Updated: no
Importance: obligatory
Implementation order: 5.4

R68: Stopping execution

It should be possible to stop the execution of an individual application as well as the execution of a more complex application that consists of several jobs. These job interrelationships are described using the job dependencies.

Previous number (D4.2.3): R68
Updated: no
Importance: obligatory
Implementation order: 3.2

R69: Changing owner permissions during Application Runtime

It must be possible to modify the owner permissions during the application runtime. To this end, the User ID and the corresponding credentials must be modified. This requirement must be discussed with WP 3.5. Furthermore, the role of the administrator as the only actor that is allowed to initiate such a change must be discussed.

Previous number (D4.2.3): R69
Updated: no
Importance: obligatory
Implementation order: 8.3

R70: Spawn jobs to other VOs

An application running on a virtual organization can spawn a job to another virtual organization.

Previous number (D4.2.3): R70
Updated: no
Importance: obligatory
Implementation order: 7.9

R71: Manual exploration and selection of hardware

The application needs to be able to select the resources it uses.

Previous number (D4.2.3): R71
Updated: no
Importance: obligatory
Implementation order: 5.1

R72: Co-Allocation

Certain applications need co-allocation of resources of several different sites. It must be possible to prohibit the co-allocation on the application level, e.g. for security reasons.

Previous number (D4.2.3): R72
Updated: no
Importance: obligatory
Implementation order: 4

R73: Distribution

An application must be able to limit its geographical distribution. This includes simple distance measurements between resources as well as the specification of continents and countries. The restriction to execute certain jobs only in certain countries is based on differing laws in different countries and the corresponding security demand of the individual applications. Thus, the system must know where the different resources are located (countries etc.).

Previous number (D4.2.3): R73
Updated: no
Importance: obligatory
Implementation order: 6.9

A.8 Data Management

This section starts with an overview of data characteristics of the applications visualized in figures A.10-A.16. These characteristics should be taken into consideration when designing the data management facilities. Note that a lot of questions in the requirements questionnaire only apply to the 12 file-based applications (cf. Figure A.10). The application that uses a database uses one database whose size ranges from ca. 16 GB to 6 TB.

R74: Concurrent access to open files

It must be possible to concurrently read from files which are open for write access. It must also be possible to concurrently write to files open for reading. The consistency requirements and respective mechanisms remain to be discussed.

Previous number (D4.2.3): R74
Updated: no
Importance: obligatory
Implementation order: 3.7

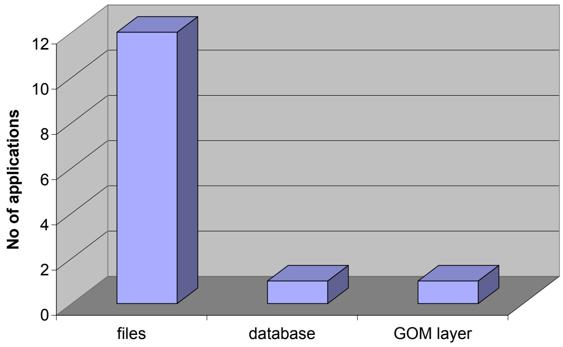


Figure A.10: Files, data bases, Global Object Management (GOM) layer

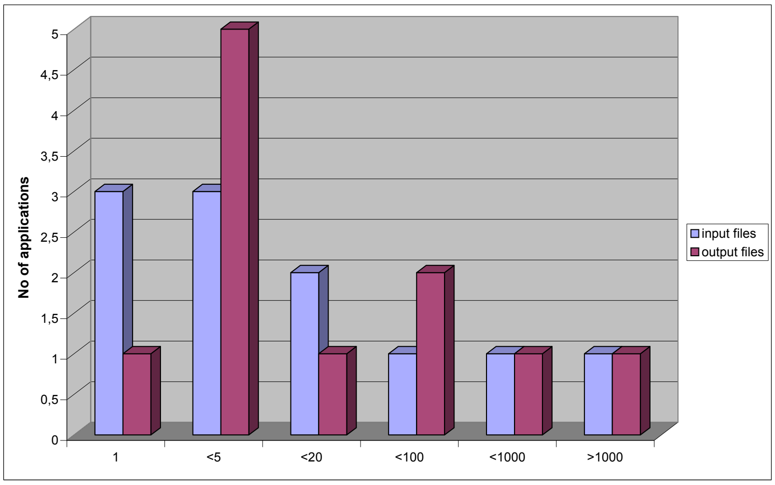


Figure A.11: Number of input and output files

R75: Required Meta Data

The usual UNIX metadata must be accessible, or at the minimum the following: the full path, global Grid user name of the owner, global virtual organization identifier (VO ID) of the owner's VO where the file has be opened for read/write, and the time of last change.

Previous number (D4.2.3): R75
Updated: no
Importance: obligatory

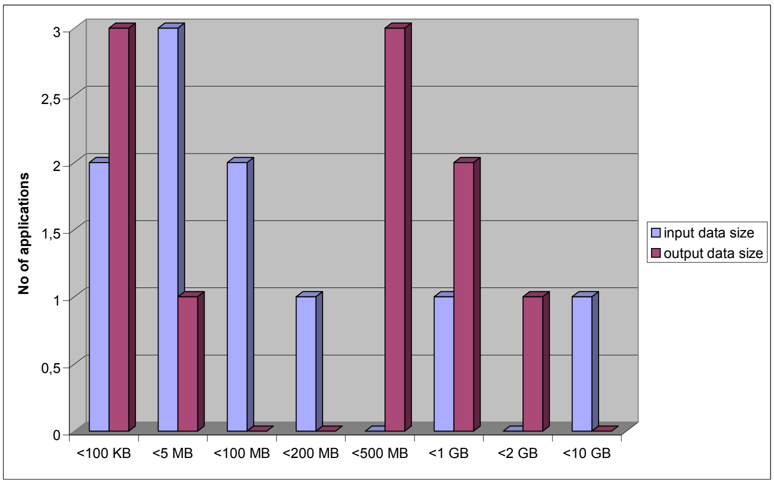


Figure A.12: Estimation of the size of input and output data in MB

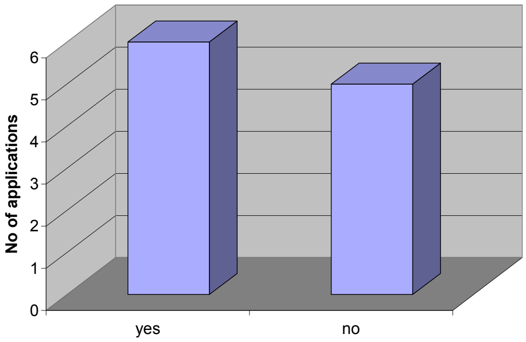


Figure A.13: Can all input and output files be specified at initialization?

Implementation order: 4.1

R76: Directories

Application defined directories are required. The applications must be able to define and use directory structures, which are then used to organize the files.

Previous number (D4.2.3): R76
Updated: no
Importance: obligatory

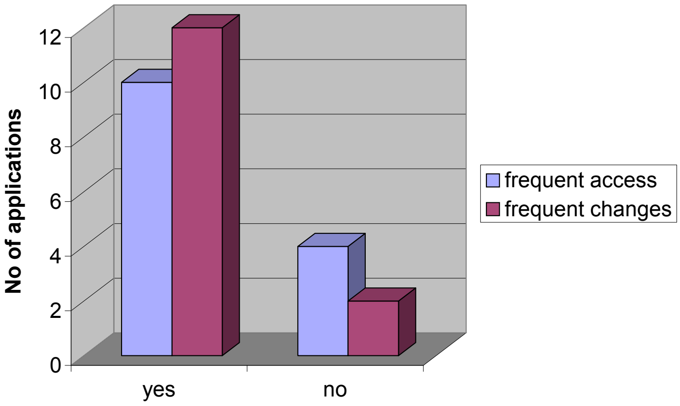


Figure A.14: Are applications accessing the corresponding data frequently?

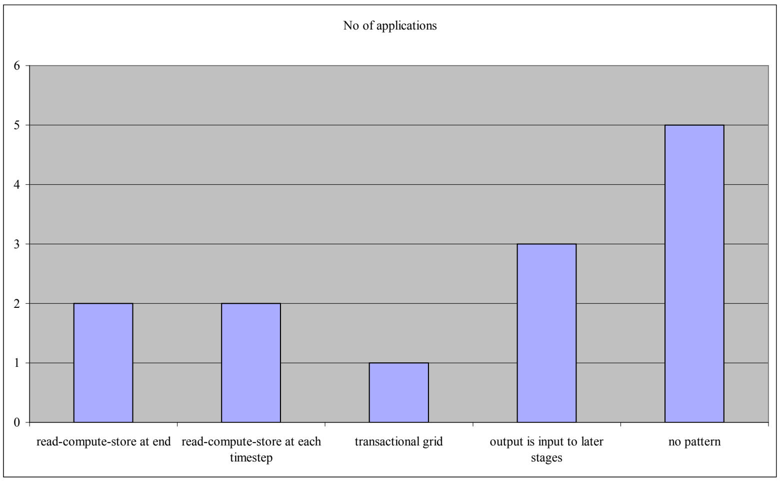


Figure A.15: Data workflow and data access patterns

Implementation order: 4.5

R77: File metadata access control granularity

It must be possible to set the access rights on parts of individual metadata.

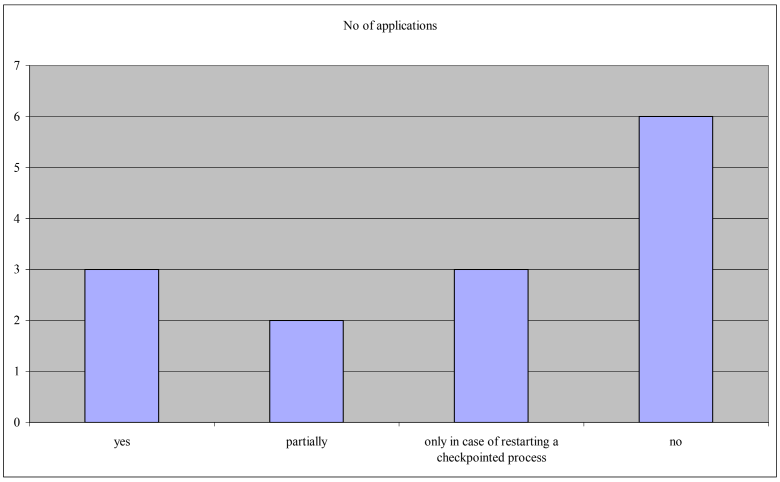


Figure A.16: Pre-fetching of application data before job execution

Previous number (D4.2.3): R77
Updated: no
Importance: obligatory
Implementation order: 8

R78: File data and metadata change monitoring/notification

It should be possible to check for file data and metadata changes. It should also be possible to subscribe to the information on file data and metadata changes, e.g. through registering a callback function.

Previous number (D4.2.3): R78
Updated: no
Importance: optional
Implementation order: 7.4

R79: Data access time

File access time must be below 10 s to prevent time-out errors in applications. The data access time for interactive applications using a database must be below 150 ms. Furthermore, data access is very frequent.

Previous number (D4.2.3): R79
Updated: no
Importance: obligatory
Implementation order: 4.3

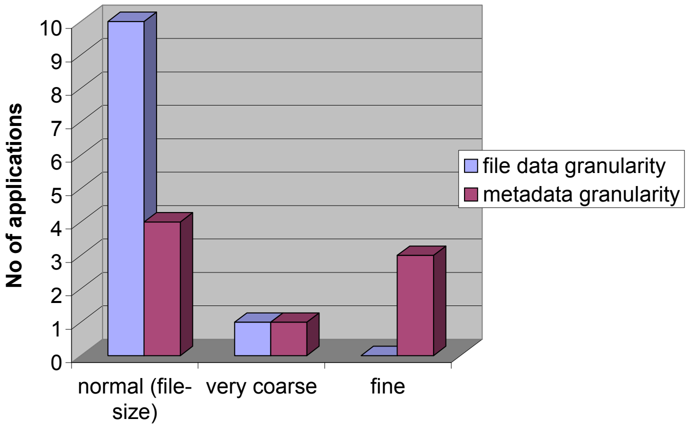


Figure A.17: Granularity of metadata access control required

R80: Replica limitation

It must be possible to limit the number of replicas. Some applications need this possibility because of copyright, storage space, security, and performance constraints.

Previous number (D4.2.3): R80
Updated: no
Importance: obligatory
Implementation order: 6.5

R81: Data versioning

The versioning of data to allow incremental changes is required.

Previous number (D4.2.3): R81
Updated: no
Importance: obligatory
Implementation order: 8.9

R82: Explicit move/copy between resources

It must be possible to explicitly copy/move data between different resources.

Previous number (D4.2.3): R82
Updated: no
Importance: obligatory
Implementation order: 4.7

R83: GOM object layer

The GOM layer must support an object based access/sharing. The GOM layer must support transactional consistency for object sharing. This includes re-startable transaction combined with optimistic synchronization.

Previous number (D4.2.3): R83
Updated: no
Importance: obligatory
Implementation order: 6

R84: Data transmission monitoring

A facility to monitor ongoing data transmission must be provided.

Previous number (D4.2.3): R84
Updated: no
Importance: obligatory
Implementation order: 3

A.9 Security in Virtual Organizations

This section consolidates the requirements for work-package 3.5 on security services for the Grid-OS. Note that the responses to the questions on security services were rather sparse, indicating that either the questions were too vague, or that it is difficult for application providers to effectively make judgments concerning their security requirements. We therefore need to have some agreement on architecture and use cases as soon as possible. The goals of the requirements analysis were to:

1. Discover and prioritize particular threats that we need to address with respect to confidentiality, integrity, availability, accountability and isolation. Note that in WP 3.2 there is a 7.3 prioritization of security services, indicating that participants recognize security as a high priority capability of the Grid and a Grid-OS.

2. Identify the set of security policies that need to be specified and their expressiveness. Note that many participants did not answer questions regarding policies, which suggests that work on policy languages should be treated as a minimal priority.

3. Determine the mechanisms required in order to enforce security policies and the effort associated with developing and integrating them as services in the OS. One comment made by one participant is that we should also include the middleware-level in the assumption of mechanisms, but we should however focus on what needs to be integrated in the OS itself.

4. Derive the requirements for administration support for OS security services, such that we need to consider what new mechanisms would introduce

5. Determine effective means of evaluating the security services once developed, integrated and deployed in a Grid environment

6. Determine if preliminary tests need to be carried out before committing to explicit requirements. This final aim of the requirements analysis is particularly important, as it was not possible for all participants in the questionnaire to commit to answers for every question. Again this can be attributed to either insufficiency in precision of the questions or the inherent difficulty of being qualitative or quantitative where security is concerned.

The requirements have been grouped according to security control objectives that have been determined according to a generic architecture for possible distribution and administration of resources in a Grid Environment, as depicted in figure 3.5.1. This is also consistent with the proposals of the Grid Architecture in [6]. This will be referenced within each requirement specification by way of clarification.

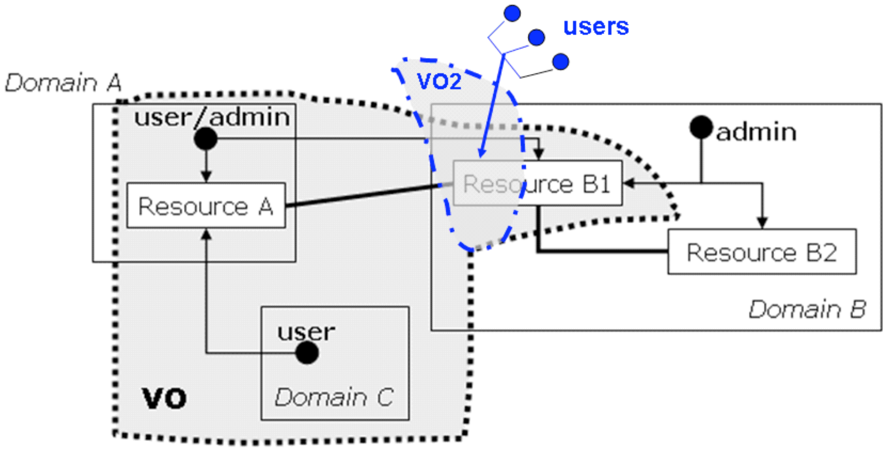


Figure A.18: A generic architecture for distribution, usage and administration of resources in a Grid environment

The possibility of covert channels to data and resources cannot be ignored, as it is possible for a malicious party to bypass the messaging infrastructure if covert channels or alternative mediums for accessing resources are established. Furthermore, the assurance of the administrative processes associated with computational nodes impacts on the way in which security requirements can be enforced. A goal of the security requirements is therefore to determine what additional information is required in messages exchanged between computational nodes, and hence what represents a “correct” message before it can be processed by an application or security service. Finally, as we are aiming towards the development and integration of security mechanisms as services in the OS-layer, security becomes a recursive requirement, where the question of how to secure security services in a Grid OS also arises.

The overall question we ask is therefore, if XtreemOS is installed on all resources (A, B1, B2), can we improve the way the following objectives are met?

1. Confidentiality of stored data
2. Confidentiality of data being communicated between end points
3. Integrity of stored data
4. Integrity of communicated data
5. Identification and authentication of users
6. Authorized access to application services
7. Guaranteed access to application services by authorized parties
8. Accountability of data access and service execution
9. Isolation of data per-VO
10. Isolation of services per-VO

R85: Data stored on resources must only be accessible by users and administrators that are members of a VO with the appropriate read access rights *(objective 1)*

Confidentiality is a fundamental requirement of systems that store, process and exchange sensitive data and information. In a Grid-enabled system the requirement for confidentiality of stored data is to ensure that data can only be accessed and read by services, users and administrators (together known as Principals) that have a "need" to read the data. A principal has a need if the following conditions are true: owner of the data OR (registered as a member of a VO with rights to the data AND assigned to a task that requires access to the data). A principal with such properties is referred to as a "valid principal" otherwise we refer to the principal as an "invalid principal". For example, the user of Domain C can only have access to data on Resource A, may have access to data on resource B1 but no access to data stored on resource B2. A local administrator who belongs to more than one VO at the same time should only be able to act according to a policy or a contract between the VOs he is a member in. This requires a solution where dedicated resource monitoring (e.g. file access) are relying on other roots of trust (e.g. hardware trust anchors) than the local administrator.

Based on the questionnaire, 50% of the participants indicated that confidentiality of stored data is a highly critical requirement. It must be possible to enforce that only owners of data, members of VOs and parties assigned tasks (as roles, rights or responsibilities) in the VO can read data stored on resources (i.e. computational nodes). A difficult requirement is to stop the administration and users of say domain B from accessing data stored on resource B1. 25% of the participants however did not answer, which is perhaps because of a difficulty with understanding how to measure the criticality of security. One potential requirement for the overall framework is therefore to provide some comprehensive metrics that can be applied for future evaluation of the security needs of applications being deployed in a Grid OS. 2 participants (12.5%) did however explicitly state that the confidentiality of stored data was not critical for their application, highlighting the flexibility that must be enabled should a shared Grid OS be used for multiple applications.

Previous number (D4.2.3): R85
Updated: no
Importance: obligatory
Implementation order: 3.9

R86: Confidential data communicated between resources in the same VO must be transported via confidential channels associated with the VO resources. *(objective 2)*

It is assumed that data is transmitted over secure, e.g internal networks but also over insecure channels such as the Internet. There is a need for mechanisms that protects messages and responses in transit between computational nodes over insecure channels. More specifically, confidentiality of data is concerned with the protection of message inputs and the corresponding outputs of responses from invalid observers. An invalid observer has similar properties as that of an invalid principal of stored data in R3.5.1. However, the security policies and mechanisms are now concerned with the properties of the channels over which messages and responses are transmitted. In order to not transmit confidential by an insecure channel, all the data for which a user or application owner can not adjust security preferences (e.g. IPC, process migration, ?) are secured by a cryptographic scheme and protocol by default.

Users and application owners should specify individual security preferences for their communication.

Again 50% indicated that confidentiality of transmitted data was highly critical, assumedly with the same reasoning as by stored data. However, fewer participants were clear about the perceived difficult of preventing and detecting breeches to this requirement. Especially at audit time only 25responded, with one participant stating high and another stating low difficulty, whereas all but one of the respondents indicated that there is a high difficulty associated with detecting runtime breaches. A typical rule-of-thumb is to aim for prevention before detection, such that it should first be endeavored to restrict illegal principals from reading data in transit.

Previous number (D4.2.3): R86
Updated: no
Importance: obligatory
Implementation order: 3.1

R87: Loss of integrity of stored data must be preventable and detectable using hash mechanisms *(objective 3)*

Storage integrity is typically stated as the ability to prevent illegal changes to data. As data in a VO may be sourced from different participants, who may not be "owners" of the data, it must be possible to validate that the data has not been altered by illegal parties. Data should be hashed and digitally signed by a trusted key stored on the operating system. Again a legal party must be a member of a VO and have the appropriate rights to make changes to data.

Previous number (D4.2.3): R87
Updated: no
Importance: obligatory
Implementation order: 5.1

R88: The integrity of data transferred between resources or received from users must be validated before being committed *(objective 4)*

There is then a need for an OS reference monitor mechanism to capture and validate all incoming and outgoing network traffic. The integrity of communicated data is concerned with ensuring that illegal change is not possible to data in transit. This differs from data in storage as the properties of communication channels tend to be more dynamic, based on the location, operating system and medium used by end point nodes. The operating system must therefore be capable of signing and verifying signatures of data in an end-to-end manner. The term "committed" suggests that a transaction framework is necessary, considering the distributed nature of the resources.

Previous number (D4.2.3): R88
Updated: no
Importance: obligatory
Implementation order: 4.9

R89: It should be possible for a user to use a single method of authentication (i.e. single sign-on) to gain authorized access to resources in a VO *(objective 5)*

Identification and authentication are again fundamental requirements for security, as integrity and confidentiality are difficult without the capability to identify and authenticate principals. Identification ensures that different principals (e.g. a source or receiver of a message) are repeatedly distinguishable from each other, while authentication associates attributes used to identify a principal with a unique root attribute such as a legal name or public key. It must be possible for all resources in a VO to identify and authenticate users requesting access to data. Users of resources should not have to be bothered with changing the way they interact due to changes in the hosting of the resource. One example is cross-domain single-sign-on (SSO), which requires an agreement of how tickets and attributes are encoded and verified, which asserts that users have been authenticated and possess the appropriate authorizations to perform actions in the VO.

Network access via socket interfaces and IPC can also be treated as resources but wasn't covered by the questionnaire. There are currently no widely used monitoring frameworks. This is of upcoming importance since these resources can become scarce.

Previous number (D4.2.3): R89
Updated: no
Importance: optional
Implementation order: 4.8

R90: It must be possible to transfer and validate authorizations to virtualized resources when the host is changed *(objective 6,7)*

Authorization requirements precede confidentiality and integrity requirements and depend on identification and authentication. It was therefore not surprising that this was the requirement indicated by most (62.5%) of the participants as highly critical. Authorization is the requirement that principals can only access data that they are authorized to use in order to perform tasks, or, in the case of multilevel security systems, that they have the requisite clearance in the system. The indication of a principal's rights to perform a task is usually indicated using a token, ticket or credential, which are different forms of associating an identity with a specific right. This follows from R3.5.5, as the authorizations should also be consistent across the domains providing resources. However, the challenge is still making sure that the local administrators of hosts do not have to breach the private policies enforced by their operating systems.

The functionality to this requirement must be available all the time in order to not interfere with the quality of service. In the case of a centralized management high availability or failover measures have to be used.

Previous number (D4.2.3): R90
Updated: no
Importance: obligatory
Implementation order: 5.4

R91: It must be possible to validate membership in VOs and ensure access to resources given proven membership and rights *(objective 5,6,7)*

Authorization and guaranteed access are two different requirements although enforced by interdependent security mechanisms/ services. That is, a principal may have been provided with a token, ticket or credential but the appropriate access control policy or service interface is not available at the time of request. The locally evaluated rules that determine if a party is authorized or not (beyond the possession of a token, ticket or credential), must also be agreed to across the set of resource providers. It may not be possible to implement this in the OS, but there need to be "hooks" to higher level services that can perform such evaluations.

Previous number (D4.2.3): R91
Updated: no
Importance: obligatory
Implementation order: 4.4

R92: It must be possible for administrators to record usage (by whom and when) of resources without users being able to deny (repudiate) usage *(objective 8)*

Accountability is the ability to enforce and prove that a principal has performed an action on a given resource at a given time. The requirements for accountability are typically a secure audit service with the ability to timestamp messages. This is for the purposes of non-repudiation, should there be a case where it must be proven that a principal has indeed performed an action, as well as billing. Should also be possible to record within which VO the resource was used.

Previous number (D4.2.3): R92
Updated: no
Importance: optional
Implementation order: 6.3

R93: Isolation of VO users - It must be possible to maintain users for different VOs separately *(objectives 9, 10)*

As users may be involved in multiple VOs, it is then necessary to separate their user data and have a means of determining for which VOs are they currently working in, when accessing data.

Previous number (D4.2.3): R93
Updated: no
Importance: obligatory
Implementation order: 5.6

R94: Isolation of data per-VO: Data of different VOs, hosted on the same physical resource must show non-interference *(objective 9)*

The isolation of data per-VO enables data to be separated between different groups and contracts. Data belonging to a VO should be logically isolated only for that VO, such that changes made in one VO, although referring to the same data element, should not be. A local administrator who can belong to more than one VO at the same time should only be able to act according to a policy or a contract between different these VOs. There is an absolute need for having automated enforcement of policies. This is however not that surprising, as confidentiality is a fundamental security requirement and most organizations with sensitive data would have already invested time and money in acquiring, developing and integrating mechanisms to enforce confidentiality policies. Most participants in the questionnaire indicated that they do have utilities integrated with their applications that already meet the basic confidentiality requirements. However, there is a mix of implementation dependent on the OS-Layer and integrating at Application Layer, which suggests that there still needs to be a consolidating framework that allows reuse of OS security services as well as application layer libraries and security modules. A multi-layered architecture for security services is therefore foreseen, which however means that the integration points between layers must also be analyzed and secured. The deferral of the enforcement of confidentiality policies to third parties was not accepted amongst the participants, such that owners of resources and data must be able to maintain control of who accesses their data, even if the data is stored remotely in a different domain. This has implications for the administration of policies, resources and security services.

Previous number (D4.2.3): R94
Updated: no
Importance: obligatory
Implementation order: 5.2

R95: Isolation of services per-VO: secure access to virtualized resources and services must be customized for each VO *(objective 10)*

In addition to isolation of data, services must also be isolated per VO. That means that each VO can try to achieve a different set of security objectives and information flow policies. This may also apply to a set of instantiations of a VO (VOs that are set up in an automatic way by, e.g. by using template).

It must not be possible for parties in different VOs to recognize that they are sharing resources nor to gain knowledge of what other parties are doing with those resources. If one of two virtualized services to the same physical resource fails, this should not interfere with the other.

Previous number (D4.2.3): R95
Updated: no
Importance: obligatory

Implementation order: 5.4

R96: The reuse and realization of established security standards and utilities is suggested *(all objectives)*

It must be possible to reuse and realize established standards for authentication and authorization in the OS ? e.g. PKI (public key infrastructure), PAM (pluggable authentication modules) and SSH (Secure Shell)

Previous number (D4.2.3): R96
Updated: no
Importance: optional
Importance for MDs: optional: similar or compatible standards will be used for MDs
Implementation order: 1.9

R97: Linking of Trust Management services with OS mechanisms *(objectives 3,4,5)*

There is a need to link mechanisms implemented at the OS layer to higher level reputation and third party trust management services, which influence access control decisions. Decision of the OS rely securely on third party information but are enforced in the kernel, e.g. pluggable reference monitors. Availability of this information must be ensured.

Previous number (D4.2.3): R97
Updated: no
Importance: obligatory
Implementation order: 6.0

R98: Semi-automation of administration and configuration of security infrastructure is necessary *(all objectives)*

It must be possible to set up and configure an XtreemOS security infrastructure in less than 1 working day, and, in worse case, less than 10

Previous number (D4.2.3): R98
Updated: no
Importance: obligatory
Implementation order: 5.0

R99: Semi-automation of adaptation and reconfiguration of the security infrastructure is necessary *(all objectives)*

It must be possible to make adaptations to the infrastructure in less than 1 working day and, in worse case, less than 5. This therefore implies a high degree of automation or a very simple set of guideline for flexible modifications to the infrastructure

Previous number (D4.2.3): R99
Updated: no
Importance: obligatory
Implementation order: 4.0

R100: Multiple bundles or configuration for XtreemOS crypto have to be considered to support different CPU performance constraints. The minimal set of resources and properties (e.g. CPU performance, memory) can be specified per VO resource. *(all objectives)***.**

In some cases partners have indicated that they expect a solution that consumes <0.5% of the CPU, while others have indicated as much as <50%.

Previous number (D4.2.3): R100
Updated: no
Importance: obligatory
Implementation order: 7.2

R101: A standard security assessment criteria and profile should be followed for evaluating the XtreemOS *(all objectives)*

Either we will need to extend existing metrics and evaluation criteria for security architectures, or we can adopt one such as Common Criteria, and first define a Protection Profile according to their specification

Previous number (D4.2.3): R101
Updated: no
Importance: obligatory
Implementation order: 5.2

A.10 Support for Mobile Devices

Note that the implementation orders for requirements R94 to R99 were rated by less than 5% of the applications. Therefore the results may not be representative.

R102: Hardware restrictions: XtreemOS for MD must support ARM architecture for PDAs and mobile phones

Three applications (21%) point out hardware restrictions concerning mobile devices. One of them considers PDA's and Mobile phones to be inappropriate for their requirements, while the other two specifically require ARM processors.

Previous number (D4.2.3): R102
Updated: no
Importance: obligatory
Implementation order: 5.0

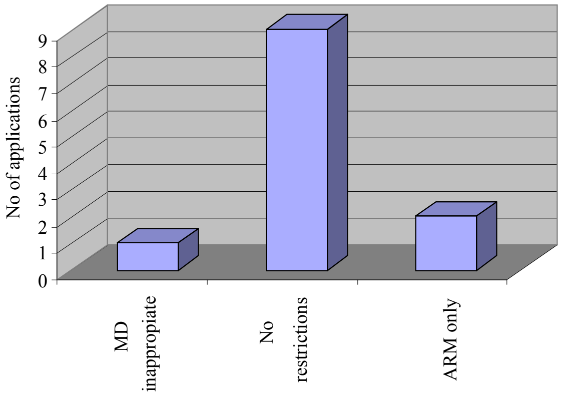


Figure A.19: Typical executing environment

R103: XtreemOS for MD must support Java

All applications which pointed out software restrictions (21%) needed some kind of Java support. Anyway, they will only need Java for monitoring, managing and instant messaging purposes. This means that support for a certain version of Java over ARM architecture will be needed.

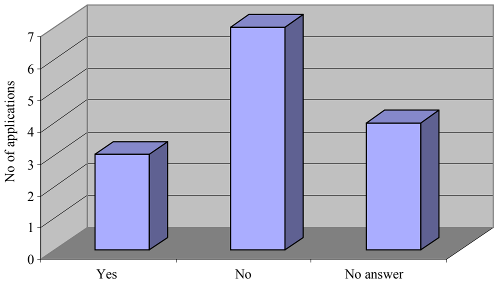


Figure A.20: Java support

Previous number (D4.2.3): R103

Updated: no

Importance: obligatory

Implementation order: 5.3

R104: XtreemOS for MD must support some basic web services protocol stack

One of the applications needs support for a basic web services protocol stack that allows the MD to function as a client to web services.

Previous number (D4.2.3): R104
Updated: no
Importance: obligatory
Implementation order: 5.5

R105: XtreemOS for MD should allow VO management

One application would benefit from using MDs for managing VOs. Only authorized users shall be able to manage VOs from MDs.

Previous number (D4.2.3): R105
Updated: no
Importance: optional
Implementation order: 8.4

R106: MDs should be considered as special nodes

More than 35% of the applications want to identify MDs as special nodes. Due to their limited processing and storage capacity is not expected to execute compute/storage intensive applications on MDs. MDs will be mainly used for monitoring and managing purposes, instant messaging and transaction performance. This means that neither their small processing capacity nor their small storage capacity will be considered an additional resource by applications.

This low capacity together with the fact that MDs aren't permanently connected, makes interesting to mark them as special nodes warning XtreemOS from scheduling a job on them.

Previous number (D4.2.3): R106
Updated: no
Importance: obligatory
Implementation order: 4.2

R107: XtreemOS for MD must provide communication and job monitoring and management facilities

50% of applications benefit from the use of a MD as a node in the Grid: most of them for managing and monitoring purposes, and only one application for performing transactions. Another application needs MDs to exchange instant messages with other users, but this can be achieved by providing a standard socket interface.

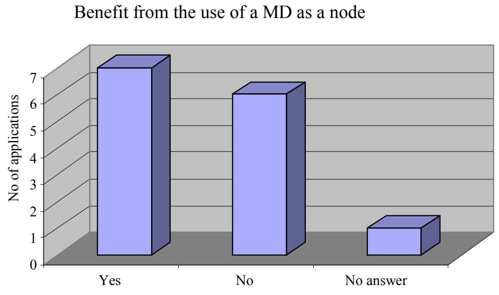


Figure A.21: Expected benefit from using MDs as a node

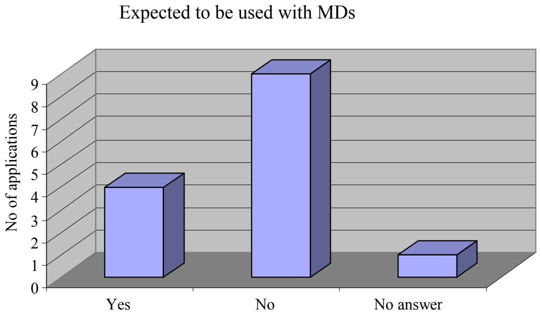


Figure A.22: No. of applications expecting to use MDs

Thus, apart from being marked as special nodes (previous requirement), basic services offered by XtreemOS for mobile devices have to be job management (launch, stop, resume, cancel, result view) and monitoring.

Previous number (D4.2.3): R107
Updated: no
Importance: obligatory
Implementation order: 6.5

R108: XtreemOS for MD should also support lightweight security methods to improve MDs' performance

42% applications allow MDs to use lightweight security methods (e.g. shorter keys to cypher communications) to improve their performance, due to their small processing capacity.

Previous number (D4.2.3): R108
Updated: no
Importance: optional
Implementation order: 8.8

A.11 Installation Requirements

R109: The installation of XtreemOS must be intuitive, familiar and not radically deviate from standard OS installation procedures

All applications (100%) require this feature, as the inclusion of XtreemOS in a large-scale system landscape should not increase the administrative burden faced by administrators. An internal guideline for installing a local testbed has been released and is used as the basis for evaluating this requirement.

Previous number (D4.2.3): R109
Updated: yes
Importance: obligatory
Implementation order: 5.0

R110: All services that can be independently started must be easily started without having to worry about the necessary dependencies

All applications (100%) require this feature, as the inclusion of XtreemOS in a large-scale system landscape should not increase the administrative burden faced by administrators. An internal guideline for installing a local testbed has been released and is used as the basis for evaluating this requirement. The installation guidelines also include how to start up different services within the operating system.

Previous number (D4.2.3): R110
Updated: yes
Importance: obligatory
Implementation order: 5.0

R111: The standard Linux system management utilities must be made available on XtreemOS and work in a backwards-compatible manner. This includes utilities like `top` used for checking memory

All applications (100%) have this basic requirement, as the application scripts and binaries will be making calls to a standard Linux or SAGA API.

Previous number (D4.2.3): R111
Updated: yes
Importance: obligatory

Implementation order: 5.0

R112: The installation of applications designed and compiled for a Linux operating system should not have to undergo recompilation and extensive changing of binaries for the purpose of installation on XtreemOS

All applications (100%) have this requirement and we have therefore made the basic assumption that all applications are previously executable on a Linux kernel.

Previous number (D4.2.3): R112
Updated: yes
Importance: obligatory
Implementation order: 5.0