Project no. IST-033576

XtreemOS

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

Final XOSAGA API engine implementation
D3.1.11

Due date of deliverable: August 31^{st}, 2010
Actual submission date: September 30^{th}, 2010

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

Type: Deliverable
WP number: WP3.1
Task number: T3.1.2

Responsible institution: VUA
Editor & and editor's address: Thilo Kielmann
Vrije Universiteit
Dept. of Computer Science
De Boelelaan 1083
1081HV Amsterdam
The Netherlands

Version 1.0 / Last edited by Thilo Kielmann / September 30, 2010

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
0.99	31/08/10	Kees van Reeuwijk, Thilo Kielmann	VUA	initial version for internal review
1.0	30/09/10	Kees van Reeuwijk, Thilo Kielmann	VUA	final version after review

Reviewers:
Susanna Martinelli (CNR) and Matej Artac(XLAB)

Tasks related to this deliverable:

Task No.	Task description	Partners involved°
T3.1.2	A runtime engine for dynamic call dispatching	VUA*

°This task list may not be equivalent to the list of partners contributing as authors to the deliverable
*Task leader

Executive Summary

This report describes the final status of the implementations of the XOSAGA API as described in deliverable D3.1.10 [5]. There are implementations available for the Java, Python, and C++ engines of XOSAGA; the complete availability of the latter two is new for this report. We also describe how to download and install each of these implementations, and how to run the test suite.

On top of the Java XOSAGA API we built a graphical shell, called Xterior, that allows the user to manipulate files an processes in an interactively. We will also describe this program.

Contents

1 Introduction

This report describes the final status of the Java, Python, and C++ implementations of the XOSAGA API engines. The definitive API of these extensions has been described in deliverable D3.1.10 [5].

For the convenience of the reader we first briefly summarize the now completed set of XOSAGA extensions to the SAGA grid API standard, and we then describe each of the three implementations (for Java, Python, C++) with respect to download repository, building/installation information and documentation.

2 The XOSAGA extensions

XOSAGA extends the SAGA [1] grid API with the following:

- Extended job management:
  - The ability to restart jobs
  - Resource management
  - Reservations

  Where 'resources' are available CPUs, available memory or disk space, etc.
- A SAGA adapter for the XtreemFS distributed file system.
- Memory buffers that are shared between the distributed processes (based on the *Object Sharing Service, OSS* [3], C++ only).
- Shared events: a mechanism to publish events, and let the distributed processes subscribe to streams of these events.
- Shared properties: a mechanism to let the distributed process share pools of information, where each process can insert information in the pool, and all processes can extract it.
- Distributed servers: a distributed server fail-over mechanism.

3 General state of XOSAGA

Briefly, all XOSAGA implementations (Java, Python, C++) implement the entire XOSAGA API, with the exception of shared memory buffers that are not supported in either Java or Python. These buffers rely on memory-mapped regions from OSS [3] that are only supported in C/C++.

4 Prerequisites

The XOSAGA implementations are generally able to adapt to their environment without intervention by the user, although obviously the user sometimes has to supply information such as account information or certificates. However, for shared buffers, events, and properties at least one running Scalaris server is needed that is started by the user. See deliverable D3.2.16 [4] for details.

5 Java implementation

The source tree of the XOSAGA Java implementation can be found in the XtreemOS Subversion repository: `svn://scm.gforge.inria.fr/svn/xtreemos/grid/xosaga/java/trunk/`.

The source code of Java XOSAGA can be compiled using Ant. Simply type '`ant`' in the root directory of the development tree to compile all code into several JAR files. All JAR files (included all dependencies) will then be placed in a new subdirectory '`lib`'. Javadoc for the XOSAGA API can be created by executing '`ant javadoc`' in the root directory of the development tree. This will create a subdirectory '`javadoc`' that contains all documentation in browsable HTML files. To run the tests, do `cd test`, and then do `ant test`. All the tests should succeed, but note that for some sections of the tests you need to have a Scalaris server running, see the previous section.

Java XOSAGA is implemented on top of Java SAGA, the Java implementation of the SAGA API. Java Saga was created and is maintained by the VUA in a separate project. It consists of an engine that dynamically loads adaptors (plugins) for various backends. In particular, it implements a FUSE adaptor that can mount various remote filesystems locally, including XtreemFS [2]. (Note that there was a dedicated XtreemFS adapter in earlier implementations. That adapter has been replaced by the more general FUSE adaptor.)

The source code of Java SAGA resides in a SVN repository at `https://gforge.cs.vu.nl/svn/ibis/saga-implementation/trunk`. However, normally it is not necessary to explicitly access this repository when building Java XOSAGA, since the Java XOSAGA repository contains a SVN external reference to it, and the required source code is updated automatically along with the 'real' XOSAGA source code.

The SAGA user guide explains how to develop and run a SAGA application, and except for the extensions this also applies to XOSAGA applications. The user guide can be found in the subdirectory '`saga/doc`'. In short, you must add all JAR files in the '`lib`' subdirectory to your Java classpath. In addition, the environment variable '`saga.location`' must contain the installation directory

of Java XOSAGA (i.e. the directory that contains the `lib` subdirectory with all the JAR files). For convenience, the script '`saga/bin/run-saga-app`' performs these steps automatically, and requires only a single environment variable `JAVA_SAGA_LOCATION` to indicate the installation directory of Java XOSAGA.

XtreemOS already contains a compiled version of Java XOSAGA in the package `xosaga-java`, which can be installed via URPMI:

```
# urpmi xosaga-java
```

All Java XOSAGA files will be installed in `/usr/share/xosaga-java`. The script `/usr/bin/run-saga-app` will use this location automatically.

6 Python implementation

The source tree of the XOSAGA Python implementation can be found in the XtreemOS Subversion repository: `svn://scm.gforge.inria.fr/svn/xtreemos/grid/xosaga/python/trunk/`

Browsable API documentation can be generated by `epidoc` as follows:

```
cd doc
epydoc --config epydoc.cfg --no-private
```

The resulting html pages are available in the subdirectory `doc/html`.

Python XOSAGA is a thin layer on top of the Java implementation. It requires Jython 2.5 or later as the Python interpreter. The script '`bin/jysaga`' can be used to start Jython plus Java XOSAGA with the right properties and classpath. Similar to the '`run-saga-app`' script in Java XOSAGA, the `jysaga` script uses the environment variable '`JAVA_SAGA_LOCATION`' to locate the Java XOSAGA installation directory.

Like the Java implementation, the Python XOSAGA is built on top of a Python SAGA implementation. In this case the SAGA implementation was implemented within the XtreemOS project, but for software-engineering reasons it was developed and is maintained as a separate project. The sources of PySaga are available from the SVN repository `https://gforge.cs.vu.nl/svn/pysaga`. Note that it is not necessary to check out these sources separately; they are part of the Python XOSAGA source tree.

Like the other XOSAGA implementations, Python XOSAGA is available as a package of the XtreemOS distribution. It can be installed via URPMI as follows:

```
# urpmi xosaga-python
```

7 C++ implementation

C++ XOSAGA extends C++ SAGA with the XtreemOS-specific packages that are also available in Java and Python, plus an implementation of shared buffers on top of the distributed shared memory implementation *Object Sharing Service*, see [3].

The source code of C++ XOSAGA is available from the SVN repository at: `https://scm.gforge.inria.fr/svn/xtreemos/grid/xosaga/cpp/trunk`.

Similar to the other implementations, C++ XOSAGA is built on top of a Saga, in this case the C++ SAGA reference implementation. developed and maintained by Louisiana State University (LSU). The latest release can be found in Subversion at: `https://svn.cct.lsu.edu/repos/saga/core/tags/releases/saga-cpp-1.5.1`.

As in the other implementations, it is not necessary to download these sources separately to get a working XOSAGA implementation. The C++ XOSAGA SVN repository contains an external reference to the SAGA sources, so the sources are downloaded automatically. (They live in the `saga` directory.)

Like the other XOSAGA implementations, C++ XOSAGA is available as a package of the XtreemOS distribution. It can be installed via URPMI as follows:

```
# urpmi xosaga-cpp
```

8 Xterior

Xterior is a GUI application on top of Java SAGA. It features a tabbed interface, where each tab can contain either a file browser or a job submission GUI. The Xterior code is available from the SVN repository at `https://scm.gforge.inria.fr/svn/xtreemos/grid/xterior/trunk` Figure 1 shows an example screenshot of Xterior.

Xterior is also available as package '`xosaga-xterior`' in the XtreemOS distribution.

9 Summary

This report describes the final implementations of the XOSAGA API, in the Java, Python, and C++ languages. With this release, we have covered all functionality according to the API description from deliverable D3.1.10 [5]. The only exception

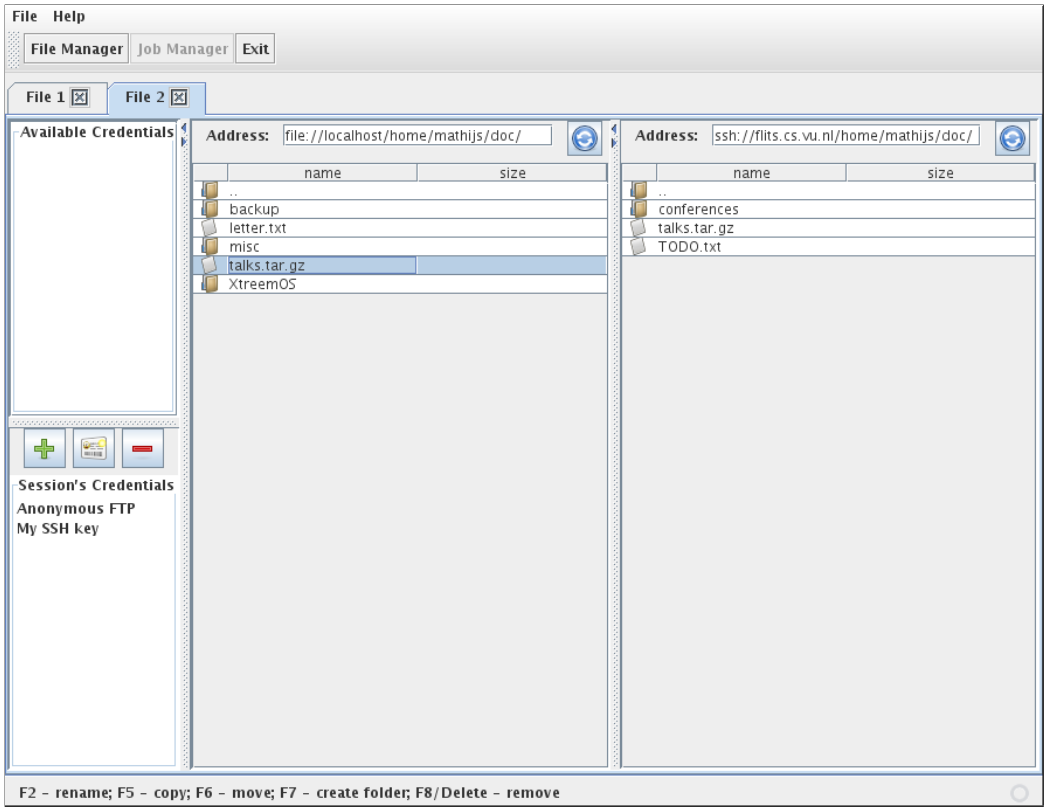

Figure 1: Screenshot from Xterior, performing a remote directory access via SSH. (The file `talks.tar.gz` is copied from the remote directory to the local one.)

is support for shared buffers in languages other than C++, which would neither be possible nor useful. Besides the three languages supported, we have also described the graphical browser Xterior that uses the Java XOSAGA implementation to deliver seamless access to XtreemOS (and other) resources.

For all implementations, we have combined information on source code availability, the bundling of binaries in the XtreemOS releases, and where interface documentation can be found.

References

[1] Tom Goodale, Shantenu Jha, Hartmut Kaiser, Thilo Kielmann, Pascal Kleijer, Andre Merzky, John Shalf, and Christopher Smith. A simple api for grid applications (saga), January 2008.

[2] XtreemOS Consortium. The XtreemOS File System - Requirements and Reference Architecture. Deliverable D3.4.1, Nov 2006.

[3] XtreemOS Consortium. Design Report for the Advanced XtreemFS and OSS Features. Deliverable D3.4.3, May 2008.

[4] XtreemOS Consortium. Final release of highly available and scalable grid services. Deliverable D3.2.16, April 2010.

[5] XtreemOS Consortium. Overview of XOSAGA programming interfaces. Deliverable D3.1.10, March 2010.