Project no. IST-033576

XtreemOS

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

Public release of XtreemOS 2.0
D4.1.23

Due date of deliverable: November 30th, 2009
Actual submission date: December 14th, 2009

Start date of project: June 1st 2006

Type: Deliverable
WP number: WP4.1

Responsible institution: Mandriva
Editor & and editor's address: Antoine Giniès
Nicolas Vigier
Edge-it / Mandriva
55 bd saint martin
75003 Paris
France

Version 0.7 / Last edited by Antoine Giniès / December 14, 2009

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
0.7	11/12/09	Antoine Giniès	EDGE	Integrated corrections
0.6	10/12/09	Sandrine L'Hermitte	INRIA	Final comments before submission
0.5	11/30/09	Peter Linnell	INRIA	final proof and review
0.4	11/30/09	Nicolas Vigier	EDGE	integrated corrections from Matej Artac
0.3	11/26/09	Peter Linnell	INRIA	proofed document
0.2	11/18/09	Antoine Giniès	EDGE	updated document
0.1	10/11/09	Nicolas Vigier	EDGE	Initial document

Executive Summary

This document describes the different procedures and scripts used to manage the software in the release of XtreemOS 2.0. The first chapter introduces the installation of an XtreemOS 2.0 system. The second one describes the platform used for packaging the software. The third chapter explains the procedures and scripts used to manage the differents versions of the software on the subversion repository. The fourth one talks about the packages stored on subversion. The fifth chapter describes the mirroring system, the packages repositories available, and all the process and scripts used when releasing an updated package. Finally, the last chapter explains how the installation CD is created.

The main goal of the XtreemOS 2.0 release is to provide a full Linux Distribution with all XtreemOS components available. Moreover, it respects the FHS (File Hierarchy Standard), and it has been thinked to provide an easy way to use all XtreemOS services. Packaging process is one of the most important task to be able to get a clean environnement, and a clean Linux distribution. All software has been designed by their respective Work Package according to a basic guide available on our internal wiki. An IRC channel was openned on freenode.net to be able to discuss in "live" with all developpers, and speed up the process of integration of XtreemOS components.

Contents

Executive Summary **1**

1 Introduction **3**

2 Packaging **4**

2.1 The build environment 4

3 Managing Software Versions **5**

3.1 Previous XtreemOS release 5

3.2 Scripts to manage versions 5

3.3 How to release a new version 6

3.4 Generating tarballs 7

4 Source packages on SVN **8**

5 Mirrors and Package repositories **10**

5.1 The package repositories 10

5.1.1 Repositories for XtreemOS 2.0 release 10

5.1.2 Installing a package from the testing repository 10

5.1.3 The sources packages 11

5.1.4 The media.cfg file 11

5.1.5 Selecting the mirror automatically 11

5.1.6 The XtreemOS 3.0 release 12

5.1.7 Scripts for managing repositories 12

5.2 The mirroring system 12

6 Creating Installation CD **13**

6.1 BCD 13

6.2 XtreemOS configuration 13

6.2.1 The xos32.xml file 13

6.3 Signing packages 17

6.4 Drakx installer 18

6.4.1 rpmsrate file 18

6.4.2 Compssuser file 18

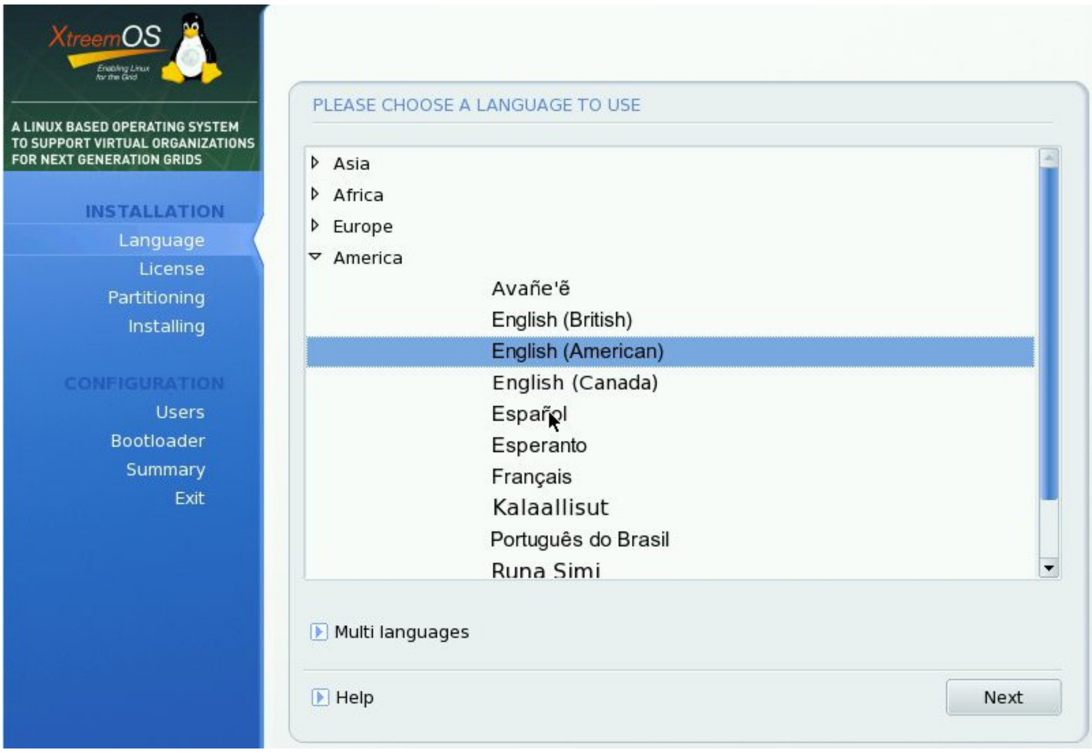
6.5 XtreemOS window manager 20

7 Conclusion **22**

8 References **22**

1 Introduction

This document presents the XtreemOS Linux distribution based on a Mandriva 2009.0. This Linux distribution is the second public release and includes all XtreemOS components. It provides both Linux-XOS and LinuxSSI-XOS, as well as some XtreemOS-G components. It has been fully redesigned to improve the integration of XtreemOS components while the development process has integrated a host of improvements based on the experience of the prior 1.0 release.



2 Packaging

This chapter discusses the packaging of the 2.0 XtreemOS Linux distribution. It details how to build packages for the 2.0 release and outlines the packages available.

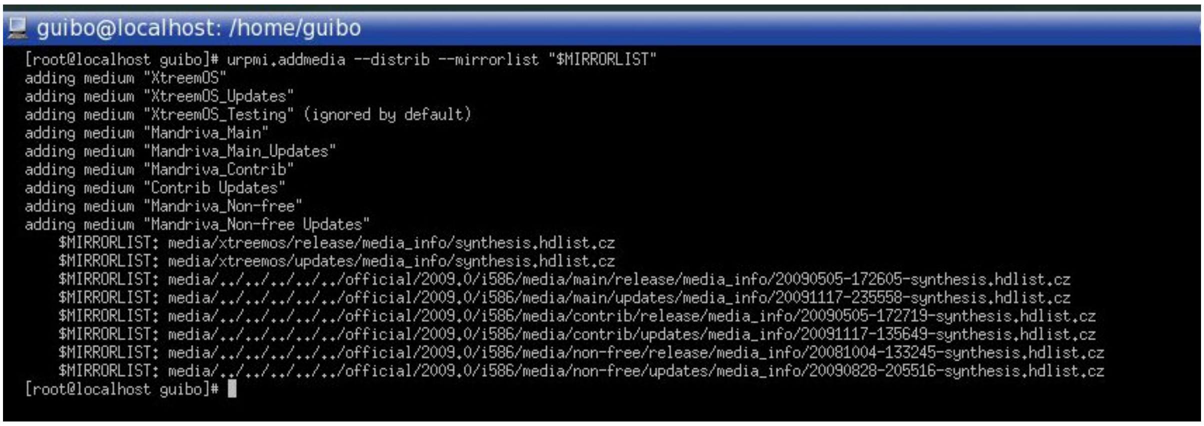
2.1 The build environment

The XtreemOS 2.0 release is based on a Mandriva 2009.0 release. XtreemOS is built on two architectures: 32 bit and 64 bit versions. We need two computers with the minimal installation of Mandriva 2009.0. Both systems should be up to date, with an official Mandriva 2009.0 repository added to the package manager media list. You can find more information about this chroot in a previous document D4.1.1, *Initial LinuxSSI integration and packaging in Debian, Mandriva and RedFlag distribution.*

```
[xtreemos@ XOS 2.0 x86_64 ~]$ cat /etc/mandriva-release
Mandriva Linux release 2009.0 (Official) for x86_64

[xtreemos@ XOS 2.0 i686 ~]$ cat /etc/mandriva-release
Mandriva Linux release 2009.0 (Official) for i586
```

Each time a package has been built it should be installed on the system to keep it up to date with latest XtreemOS components, otherwise it can lead to errors, or create dependencies on packages with obsolete libraries or tools. Since the release of Mandriva 2009.0, Java is now open-source, so we redistribute it with the XtreemOS release.



```
[root@localhost guibo]# urpmi.addmedia --distrib --mirrorlist "$MIRRORLIST"
adding medium "XtreemOS"
adding medium "XtreemOS_Updates"
adding medium "XtreemOS_Testing" (ignored by default)
adding medium "Mandriva_Main"
adding medium "Mandriva_Main_Updates"
adding medium "Mandriva_Contrib"
adding medium "Contrib Updates"
adding medium "Mandriva_Non-free"
adding medium "Mandriva_Non-free Updates"
    $MIRRORLIST: media/xtreemos/release/media_info/synthesis.hdlist.cz
    $MIRRORLIST: media/xtreemos/updates/media_info/synthesis.hdlist.cz
    $MIRRORLIST: media/../../../../../official/2009.0/i586/media/main/release/media_info/20090505-172605-synthesis.hdlist.cz
    $MIRRORLIST: media/../../../../../official/2009.0/i586/media/main/updates/media_info/20091117-235558-synthesis.hdlist.cz
    $MIRRORLIST: media/../../../../../official/2009.0/i586/media/contrib/release/media_info/20090505-172719-synthesis.hdlist.cz
    $MIRRORLIST: media/../../../../../official/2009.0/i586/media/contrib/updates/media_info/20091117-135649-synthesis.hdlist.cz
    $MIRRORLIST: media/../../../../../official/2009.0/i586/media/non-free/release/media_info/20081004-133245-synthesis.hdlist.cz
    $MIRRORLIST: media/../../../../../official/2009.0/i586/media/non-free/updates/media_info/20090828-205516-synthesis.hdlist.cz
[root@localhost guibo]# 
```

3.

3 Managing Software Versions

3.1 Previous XtreemOS release

For the previous release, developers used the subversion repository to develop their software and when it was ready, created a tar ball manually before uploading it to the gforge files section.

We had some problems with this method :

- Some developers would forget to update version numbers when uploading a new version
- Tar balls were created manually, and often included some small errors : temporary files not removed, files renamed, root directory different for each version, etc ...
- The software on svn was not always up to date. Some small modification were sometimes included in the tar balls, but not committed on svn.
- Most of the time, no tags were created for svn for new versions
- Viewing differences between two versions was not very simple. It required downloading the two versions and run diff manually.
- gforge files pages are not convenient for listing versions available, nor downloading/uploading files quickly.

3.2 Scripts to manage versions

To correct this we created scripts to manage versions of the software on svn. The developers tag the version of the software on svn using the script, after which we use another script to download the version from svn and automatically generate a tarball that we use for packaging.

The following scripts are available :

- *release-version*: create a new version of a component. The version to create should be written inside the VERSION file. examples:
  - release trunk version of xtreemos-localconfig :

    ```
    $ release-version xtreemos-localconfig trunk
    ```

  - release branch 1.0 version of xtreemos-localconfig :

    ```
    $ release-version xtreemos-localconfig 1.0
    ```

- *get-tarball*: get a tarball from a version of a component on svn. examples :
  - get version 1.5 of xtreemos-localconfig :

    ```
    $ get-tarball xtreemos-localconfig 1.5
    [...]
    xtreemos-localconfig-1.5.tar.bz2 has been created.
    ```

  - get trunk version of xtreemos-localconfig :

    ```
    $ get-tarball xtreemos-localconfig trunk
    [...]
    xtreemos-localconfig-trunk.tar.bz2 has been created.
    ```

- *diff-versions*: show differences between two versions of a component. examples :
  - see differences between version 1.4 and 1.5 :

    ```
    $ diff-versions xtreemos-localconfig 1.4 1.5
    ```

  - see differences between version 1.5 and trunk :

    ```
    $ diff-versions xtreemos-localconfig 1.5 trunk
    ```

  * diff-versions-cl: show commit logs between two versions. examples :
  - see commits between version 1.4 and 1.5 :

    ```
    $ diff-versions-cl xtreemos-localconfig 1.4 1.5
    ```

  - see differences between version 1.5 and trunk :

    ```
    $ diff-versions-cl xtreemos-localconfig 1.5 trunk
    ```

- *generate-changelog*: generate ChangeLog file from svn logs. example :

  ```
  $ svn co url/MyComponent
  $ cd MyComponent
  $ $SCRIPTSDIR/update-changelog ChangeLog
  ```

- *list-versions*: list all versions of a component example :

  ```
  $ list-versions
  $ list-versions xtreemos-localconfig
  + svn ls svn://scm.gforge.inria.fr/svn/xtreemos/grid/xtreemos-localco
  1.4/
  1.5/
  2.0/
  ```

- *list-branches*: same as list-versions, but for branches

3.3 How to release a new version

Releasing a new version is quite simple. The first thing is to create or update is a *VERSION* file containing the version number, in the trunk or a branch. Next step is to run the script *release-version* with the name of the component and trunk or the name of a branch as the arguments. This script will create a tag on the svn corresponding to the version. Then it's done. You can now generate a tarball using the *get-version* script, get a diff between two version or branches using the *diff-versions* script, get the commit logs between two version or branches using the *diff-versions-cf* script.

3.4 Generating tarballs

Creating tarballs is done using the *get-tarball* script with the name of the component and the version or branch. This script will download the files from the subversion repository, then if a *Makefile* is present run *make dist-clean*, if a *build.xml* file is present run *ant dist-clean*. Ant or Makefile rules are usually used to remove any files that are included the subversion repository, but that we do not want to include in the tarball.

4 Source packages on SVN

All packages released for XtreemOS are stored in the subversion repository:

svn+ssh://scm.gforge.inria.fr/svn/xtreemos/distribution/packages/2.0

Each sub-directory is the name of a XtreemOS components. To create an RPM package we need the SOURCES directory which contains the source code of the program, patches and any external files needed to build or add some documentation with the program along with the SPECS directory which contains the spec file to be able to generate the SRPMS and RPMS packages.

We use a custom *rpmmacros* file to create package with an *xos2.0* dist suffix instead of using *mdv2009.0* one. In the home directory of the user used to build packages, we create a file called */.rpmmacros* which will contain:

```
%_topdir /home/xtreemos/rpm
%_tmppath /home/xtreemos/rpm/tmp
%_packager Mandriva team <xtreemos@mandriva.com>
%_signature gpg
%_gpg_path ~/.gnupg
%distribution XtreemOS Linux
%vendor XtreemOS
%_signature             gpg
%_gpg_name              xtreemos
%distribution           Mandrake Linux
%_pgp_path /home/xtreemos/.signature.gpg
%distsuffix xos
%mes_release 2.0
%mkrel(c:) %{-c: 0.%{-c*}.}%{1}%{?subrel:.%subrel}%{?distsuffix:
%distsuffix}%{?!distsuffix:mdv}%{?mes_release:%mes_release}%{?!
mes_release:%{?mandriva_release:%mandriva_release}}
```

So if we build the package *cdaclient*, the full rpm name will be: cdaclient-0.2.7-1xos2.0.noarch.rpm not cdaclient-0.2.7-1mdv2009.0.noarch.rpm

For example if we want to build the *cdaclient* XtreemOS components, checkout the cdaclient components in a directory with *svn* and build it:

```
[xtreemos@ XOS 2.0 x86_64 cdaclient]$ cat ~/scripts/rb
#!/bin/sh
rpmbuild --define "_topdir $PWD" -ba $1
[xtreemos@ XOS 2.0 x86_64 cdaclient]$ pwd
/home/xtreemos/packages2.0/cdaclient
[xtreemos@ XOS 2.0 x86_64 cdaclient]$ ~/scripts/rb SPECS/cdaclient.spec
Executing(%prep): /bin/sh -e /home/xtreemos/rpm/tmp/rpm-tmp.81119
+ umask 022
+ cd /home/xtreemos/packages2.0/cdaclient/BUILD
+ ’[’ 1 -eq 1 ’]’
+ ’[’ 1 -eq 1 ’]’
+ cd /home/xtreemos/packages2.0/cdaclient/BUILD
+ rm -rf cdaclient-0.2.7
+ /usr/bin/bzip2 -dc /home/xtreemos/packages2.0/cdaclient/SOURCES/
cdaclient-0.2.7.tar.bz2
```

```
+ tar -xf -
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ cd cdaclient-0.2.7
+ exit 0
Executing(%build): /bin/sh -e /home/xtreemos/rpm/tmp/rpm-tmp.81119
+ umask 022
+ cd /home/xtreemos/packages2.0/cdaclient/BUILD
+ cd cdaclient-0.2.7
..........
Wrote: /home/xtreemos/packages2.0/cdaclient/SRPMS/cdaclient-0.2.7-1
xos2.0.src.rpm
Wrote: /home/xtreemos/packages2.0/cdaclient/RPMS/noarch/cdaclient-0
.2.7-1xos2.0.noarch.rpm
Executing(%clean): /bin/sh -e /home/xtreemos/rpm/tmp/rpm-tmp.44672
+ umask 022
+ cd /home/xtreemos/packages2.0/cdaclient/BUILD
+ cd cdaclient-0.2.7
+ /bin/rm -Rf /home/xtreemos/rpm/tmp/cdaclient-0.2.7-1xos2.0-buildroot
+ exit 0
```

5 Mirrors and Package repositories

5.1 The package repositories

RPM packages can be installed installed manually using the rpm command, however users need to manually install all the required dependencies. Alternatively, they can use urpmi to automatically download and installed the required dependencies from the packages repositories. A repository is a directory containing packages with a file containing a list of the packages and metadata about the packages: the list of files, descriptions, dependencies, etc... This list is called an hdlist, and can be generated using the genhdlist2 command.

5.1.1 Repositories for XtreemOS 2.0 release

For the XtreemOS 2.0 release, there are a few packages repositories available :

- *The Mandriva 2009.0 repositories*: the XtreemOS 2.0 release is based on Mandriva 2009.0. The base packages and all the software a user wishes to install can be installed from those repositories.
- *the XtreemOS release repository*: this repository contains all the XtreemOS packages as they were at the time of the XtreemOS 2.0 release.
- *the XtreemOS updates repository*: this repository contains packages which have been updated after the release. It contains only tested updates, created to fix specific bugs. Packages are almost always tested before being uploaded in this repository. Users should be able to install all updates from this repository with no regressions.
- *the XtreemOS testing repository*: this repository contains update candidates. The packages from this repository has not been extensively tested and are recommended for use by normal end users. For this reason, the testing repository is added by default, but not enabled. Only developers or experienced testers should enable this repository. When the packages from this repository have been tested and confirmed to work they can be moved to the updates repository.

5.1.2 Installing a package from the testing repository

The XtreemOS Testing repository is automatically added, but it is not enabled by default. The repository can be enabled, however doing so is not recommended to avoid blindly installing packages from this repository. The following commands can be used to install and query packages available from the testing repository.

The testing repository is by default named *XtreemOS_Testing*. Before running any of the following commands, the hdlist for the *XtreemOS_Testing* repository needs to be updated, with the following command :

```
# urpmi.update XtreemOS_Testing
```

The following commands can then be used :

- To list updates available from the testing repository, for the currently installed packages :

  ```
  $ urpmq --auto-select --media XtreemOS_Testing
  ```

- To install all updates available from the testing repository, for the currently installed packages :

  ```
  # urpmi --auto-select --searchmedia XtreemOS_Testing
  ```

- To list all packages available in the testing repository :

  ```
  $ urpmq -y --media XtreemOS_Testing .
  ```

- To install a single package from the testing repository :

  ```
  # urpmi --searchmedia XtreemOS_Testing packagename
  ```

5.1.3 The sources packages

The packages repositories previously mentioned are repositories for the binary packages. They are available for both *i586* and *x86_64*. For each repository there is a corresponding repository containing the source packages. The sources packages can be extracted and contain all the sources for building the software along with the necessary RPM instructions.

5.1.4 The media.cfg file

The file media.cfg is the file that describes all the repositories and categorizes if they should be added by default or not. The media.cfg for xtreemos lists the Mandriva 2009.0 repositories and the XtreemOS repositories. Running the following command will setup all the repositories required to run XtreemOS :

```
# urpmi.addmedia --distrib [MIRROR_URL]/MandrivaLinux/devel/xtreemos/2.0/
```

5.1.5 Selecting the mirror automatically

In the previous command line, the user needs to manually specify the mirror he wants to use. It is also possible to use aria2 to automatically select the fastest mirror available :

```
# urpmi.addmedia --distrib --mirrorlist
```

This command will look at the file */etc/product.id* to know which distribution we are running. On *XtreemOS 2.0 x86_64*, this will contain :

```
$ cat /etc/product.id
vendor=XtreemOS,distribution=XtreemOS Linux,type=xtreemos,version=2.0,
branch=Official,release=1,arch=x86_64,product=Free
```

From this file, it will construct the url to load the list of mirrors available :

```
http://api.mandriva.com/mirrors/xtreemos.2.0.x86_64.list
```

This list is updated nightly via a script run on Mandriva servers to check the availability of the XtreemOS packages on the mirrors and remove the unavailable mirrors, as well as, add the new mirrors.

5.1.6 The XtreemOS 3.0 release

The development for the 3rd release has started. The package repositories for the 2nd release are stored in the directory *2.0*. Another directory, *3.0*, has been created to store the development packages for the new version, with repositories similar to the XtreemOS 2.0 repositories.

5.1.7 Scripts for managing repositories

As we are often building new packages and adding them to various repositories, to save time and avoid mistakes, we have created scripts automating the task of moving packages, generating hdlists and finally removing previous versions of the packages, signing the new packages with gpg. The scripts also prepare an email to send to the developers mailling list to announce the new packages.

The following scripts are available :

- move-noarchrpms-to-release-2
- move-noarchrpms-to-release-3
- move-noarchrpms-to-testing-2
- move-noarchrpms-to-testing-3
- move-noarchrpms-to-updates-2
- move-rpms-to-release-2
- move-rpms-to-release-3
- move-rpms-to-testing-2
- move-rpms-to-testing-3
- move-rpms-to-updates-2
- move-srpms-to-release-2
- move-srpms-to-release-3
- move-srpms-to-testing-2
- move-srpms-to-testing-3
- move-srpms-to-updates-2

The number at the end of the script name indicate the version of Xtreemos, 2.0 or 3.0.

5.2 The mirroring system

The packages and install ISOs are stored on the Mandriva mirrors. We use scripts to synchronize the XtreemOS packages from our build machines to the primary Mandriva mirror, using rsync. This primary mirror and the XtreemOS repositories are then replicated to several other mirrors around the world.

6 Creating Installation CD

This section explains the new tools created to furnish the XtreemOS 2.0 distribution and the necessary steps to obtain the distribution.

6.1 BCD

The BCD tool is a program to generate an ISO CD image based on Mandriva Distribution. It has been rewritten from scratch to replace the old MKCD which is no longer usable with the current release and very difficult to maintain. This tool was developed and tested specifically for XtreemOS 2.0, as MKCD has some major issues with the update media. Moreover, MKCD does not properly support the RPM *suggest* tag. Thus, it was decided to write a simpler tool which provides better dependencies, requires, suggest and conflicts of a packages through the rpm manager (in our case *urpmi*). BCD easily supports updates media and creates ISO CD images with the latest packages available in a repository. As this tool is quite new, it has not been tested extensively on a lot of distributions it may have unknown bugs and does need some improvements. While it is an immature application, the positive results using in the creation of XtreemOS 2.0 ISO, lead us to use it and replace MKCD to build our Free and Corporate product at Mandriva.

One major BCD feature is the support of easy to write and understand configuration files in XML format. BCD tool uses the *genhdlist2* tool from the system to create hdlists and synthesis which contains data about packages available and it can support the *file-deps* file providing specific requirements for packages. It supports adding of an *isolinux* entry (to add some more installation option) and can change the theme of *all.rdz* files (file used to boot the system on a CDROM). To add a package on the ISO or exclude one you just have to write a list with the name of the package or the provides of this packages into a file. This file will be used for one or more media. BCD can use custom rpmsrate: the file which defines the weight of packages per categories, so if the package should be installed or not, and a custom compssusers.pl file which defines categories of packages you want to install in the installation step process.

Thus BCD makes the creation of a custom Linux distribution based on a Mandriva product a simpler task. BCD uses *urpmi* to calculate dependencies, conflicts, requires and suggests of all packages. So, with some experience with this tool, discovering potential problems with ISO creation is not difficult. Moreover BCD can support extra external media with hdlists, therefore easing the use of external repositories of packages and manages all dependencies and possible rpm conflicts.

6.2 XtreemOS configuration

We used two XML configuration files; one for each architecture (32 bits and 64bits). All necessary information and configuration files used to build ISOs are available on the public subversion repository of XtreemOS:

svn://scm.gforge.inria.fr/svn/xtreemos/distribution/build-xos-distribution

6.2.1 The xos32.xml file

In this chapter we will explain the *xos32.xml* file used to create the 32-bit XtreemOS 2.0 distribution.

- Name of the Linux Distribution

```
<productname desc="The Linux product">
        <nameid>XtreemOS</nameid>
```

- You must specify on which distribution this ISO will be built, the architecture and the repository location. XtreemOS 2.0 is based on a Mandriva 2009.0 and in this instance we want to build a 32-bit release.

```
<!-- you must specify on which distribution this ISO will be built --
<based_on>2009.0</based_on>
<arch>i586</arch>
<repo desc="Where is the repository of the distribution">
        <path>/mnt/BIG/dis</path>
</repo>
```

- All isolinux information is stored there. If you wish to add a custom isolinux entry this is the place to store it.

```
<isolinux desc="isolinux info">
       <defaultpath>isolinux</defaultpath>
       <fullpath>/home/xtreemos/export_i386/isolinux</fullpath>
       <!-- add hdt entry in syslinux.cfg -->
      <entry></entry>
</isolinux>
```

- We used a custom installer patched to support XOS. Please see the next chapter for more details.

```
<installer desc="path to the Mandriva installer (stage1/stage2)">
      <defaultpath>install</defaultpath>
      <fullpath>/home/xtreemos/export_i386</fullpath>
      <patch>/home/xtreemos/Build/pieces/patch-oem.pl</patch>
      <advertising>
             <defaultpath>advertising</defaultpath>
             <fullpath>/home/xtreemos/Build/pieces/advertising</fullp
      </advertising>
</installer>
```

- The theme used to build the 2.0 release. The theme must be available to be able to patch *altX/all.rdz* files.

```
<theme name="Mandriva-Free">
         <bootsplash>Mandriva-Free</bootsplash>
         <bootsplash_path>/usr/share/bootsplash</bootsplash_path>
         <gfxboot_path>/usr/share/gfxboot</gfxboot_path>
</theme>
```

- List of media to take into account to build the release. The first media must be named *Main* because the main media is mandatory for all other media. Morever the *Main* media is always the name used in a Mandriva repository.

```
<media desc="media to take into account to build the ISO">
```

- The main media list includes the list of packages and packages to exclude. We use a specific list for languages to in order to reduce the number of language packs installed. We want to use the *suggest* tag of all packages available in this media. The media will be taken in the *media/main* directory of the repository. In this media we want to use the *release* and *updates* sub-media, but we don't want to activate use the *bakcport/testing* sub-media.

  ```
  <list name="Main" desc="Main_media">
          <input name="base">
                  <file>lists/input_main</file>
          </input>
          <input name="xos">
                  <file>lists/xos_list</file>
          </input>
          <input name="lang">
                  <file>lists/languages</file>
          </input>
          <exclude name="exclude">
                  <file>lists/exclude_main</file>
          <exclude>
          <nosuggests>no</nosuggests>
          <!-- will be taken in {repo}{version}{arch}/media -->
          <mediadir>main</mediadir>
          <destmedia>main</destmedia>
          <pubkey>release/media_info/pubkey</pubkey>
          <release>yes</release>
          <updates>yes</updates>
          <backport>no</backport>
          <testing>no</testing>
  </list>
  ```

- Same as above except that we are using the *contrib* media, while all packages will be stored in the *media/main* directory of the ISO.

  ```
  <list name="Contrib" desc="Contrib_media">
       <input name="base">
            <file>lists/input_main</file>
       </input>
       <input name="xos">
            <file>lists/xos_list</file>
       </input>
       <exclude name="exclude">
            <file>lists/exclude_contrib</file>
       </exclude>
       <nosuggests>no</nosuggests>
       <mediadir>contrib</mediadir>
       <destmedia>main</destmedia>
       <pubkey>release/media_info/pubkey</pubkey>
       <release>yes</release>
       <updates>yes</updates>
       <backport>no</backport>
       <testing>no</testing>
  </list>
  ```

- This section describes the XtreemOS media, which uses an external media directory with *hdlists*.

```
<list name="xos" desc="xos_media">
     <media_supp>Contrib</media_supp>
        <input name="base">
            <file>lists/input_main</file>
        </input>
        <input name="xos">
            <file>lists/xos_list</file>
        </input>
        <input name="lang">
            <file>lists/languages</file>
        </input>
        <exclude name="exclude">
            <file>lists/exclude_main</file>
        </exclude>
        <nosuggests>no</nosuggests>
        <mediadir>Main2</mediadir>
        <destmedia>main</destmedia>
        <external name="xos">
            <fullpath_mediadir>/home/xtreemos/repo_xos/2.0/i586/media
        </external>
        <pubkey>release/media_info/pubkey</pubkey>
        <release>yes</release>
        <updates>yes</updates>
        <backport>no</backport>
        <testing>no</testing>
</list>
</media>
```

- The *mediainfo* section defined where to find custom *rpmsrate* and *compssusers* files.

```
<mediainfo desc="All files relatives to media_info dir">
     <version>2.0</version>
     <fullpath>/home/xtreemos/Build/pieces</fullpath>
     <mediainfo_dir>media/media_info</mediainfo_dir>
     <rpmsrate>rpmsrate</rpmsrate>
     <compssusers>compssUsers.pl</compssusers>
     <filedeps>file-deps</filedeps>
     <filter>lists/extra-filter</filter>
     <askmedia desc="add for other media ?">
           <todo>yes</todo>
     </askmedia>
     <suppl desc="add an external media ?">
           <todo>yes</todo>
     </suppl>
     <xmlinfo>
           <todo>no</todo>
     </xmlinfo>
</mediainfo>
```

- ISO parameter is the last section of the BCD configuration file. It defines the name of the release, ISO header and there is also a section to customize *genisoimage* options.

```
    <iso>
       <tag>valladodid</tag>
       <minor>1</minor>
       <subversion>1</subversion>
       <product>Download</product>
       <branch>Devel</branch>
       <type>basic</type>
       <size>700</size>
       <mediacfg>media/media_info/media.cfg</mediacfg>
       <hdlist>yes</hdlist>
       <synthesis>yes</synthesis>
       <header desc="Iso header info">
              <systemid>XtreemOS Linux</systemid>
              <volumeid>XOS-2.0-i586</volumeid>
              <volumesetid>XtreemOS Linux - 2.0 - i586 CD</volumesetid>
              <publisherid>XtreemOS</publisherid>
              <datapreparer>XtreemOS-Linux</datapreparer>
              <applicationid>XtreemOS Linux - 2.0</applicationid>
              <copyrightid>XtreemOS Linux</copyrightid>
       </header>
       <genisoimage desc="iso generation tool">
              <options>-f -r -J -hide-rr-moved -nobak -cache-inodes -no-e
              <bootcat>isolinux/boot.cat</bootcat>
       <builddir>build</builddir>
              <isodir>iso</isodir>
       </genisoimage>
    </iso>
    </productname>
```

6.3 Signing packages

As is recommended practice, XtreemOS packages are signed with a unique GPG key, so users downloading the RPM from a remote repository can verify the authenticity of RPMs. The installation process step imports the trusted GPP key in the RPM database. Each time a user installs an rpm with *urpmi* or *rpm*, the system verifies the package's signature. If the package has not been signed or if the key is not the trusted one, the package manager stops the installation process and asks if the user really wants to install it.

To be able to sign RPMs using GPG, the *rpm* manager must be configured to run GPG and find the correct key to sign RPMs. All this information is stored in the *rpmmacros* file in the home directory of the user.

They are a few commands to know about RPM signatures. Therefore we first verify ASCII armor of our XtreemOS pubkey:

```
[xtreemos@ XOS 2.0 x86_64 Build]$ gpg -a Rpm-GPG-KEY-XOS
pub  1024D/41F7F501 2008-07-07 xtreemos
sub  2048g/BEC74164 2008-07-07
```

Then, the ASCII armored XtreemOS public key can be added to the RPM database using –*import* option:

```
[xtreemos@ XOS 2.0 x86_64 Build]$ sudo rpm --import RPM-GPG-KEY-XOS
```

Now we check that this public key was imported correctly:

```
[xtreemos@ XOS 2.0 x86_64 Build]$ rpm -qa gpg-pubkey* | grep -i 41F7F501
gpg-pubkey-41f7f501-487298a9
```

If you want to remove a GPG key, use the *-e* option or *rpm*:

```
[xtreemos@ XOS 2.0 x86_64 Build]$ rpm -e gpg-pubkey-41f7f501-487298a9
```

To check that an RPM has been signed with the XtreemOS GPG key use the *-K* option:

```
[xtreemos@ XOS 2.0 x86_64 Build]$ rpm -Kv cdaclient-0.2.7-1xos2.0.noarch.
cdaclient-0.2.7-1xos2.0.noarch.rpm:
    Header V4 DSA signature: OK, key ID 41f7f501
    Header SHA1 digest: OK (f465a40b9d42058714ee1c60b1b790b9858e7d07)
    MD5 digest: OK (0a473434b410b27f92a7d82a603c6ce0)
    V4 DSA signature: OK, key ID 41f7f501
```

6.4 Drakx installer

Drakx is the installer and it has been modified to support XtreemOS release. This installer is split into two parts: *stage1* which provides the installation method, through CDROM, network or usb. *Stage1* loads *stage2* which is the graphic part of the installation process. We modified the *stage1* to use a linus-kernel-2.6.27.24 without any patches. In stage2, we add support for XtreemOS components through Perl modules: xtreemfs_osd, xtreemfs_mrc, xtreemfs_dir, xosd, xvoms, rss, srds and xtreemos_check. They can be used to configure XtreemOS components while installing XtreemOS 2.0 and they are used in the drakxtools packages which provides the *drakxosconfig* tool, a graphic interface to configure some XtreemOS services.

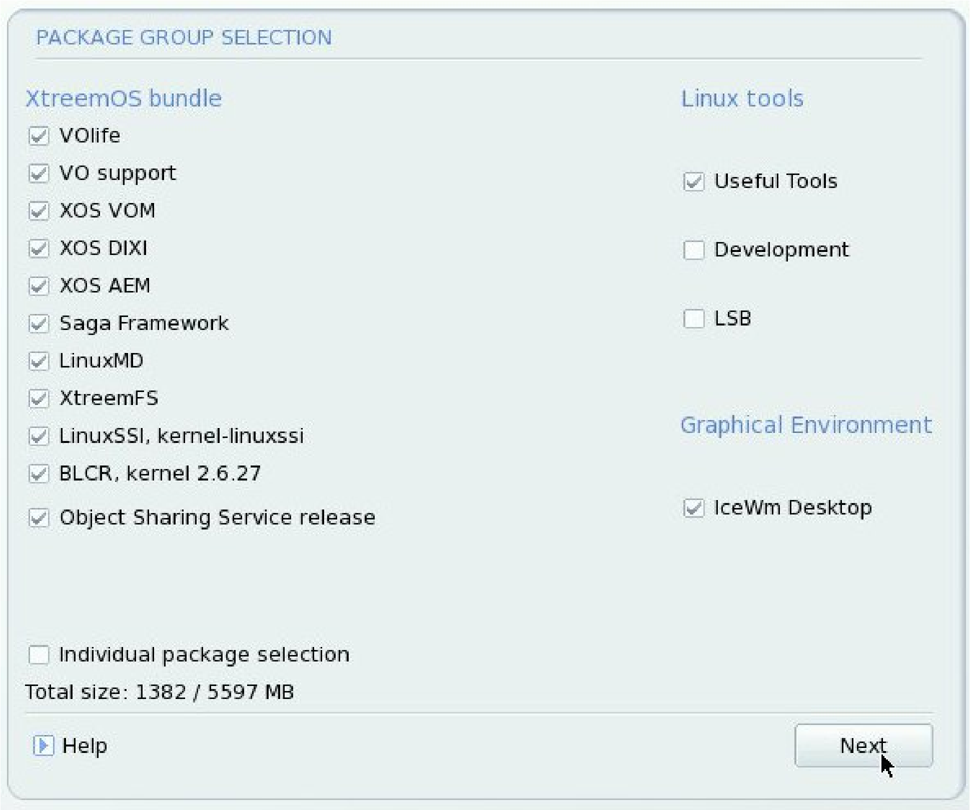
6.4.1 rpmsrate file

The *rpmsrate* file defines if a package should be installed or not, if the categories has been chosen during the Package Installation Step of the installer. To support all XtreemOS components we add some extra categories to the typical rpmsrate: CAT_XOS, CAT_XOS_VOSUPPORT, CAT_VOM, CAT_DIXI, CAT_AEM, CAT_SAGA, CAT_LINUXMD, CAT_LINUXSSI, CAT_OSS, CAT_BLCR, CAT_VOLIFE, CAT_XTREEMFS, CAT_CORE, CAT_RESOURCE, CAT_CLIENT, CAT_XOSDOC. Each category has a list of packages to install, depending on the requirements of XtreemOS services.

6.4.2 Compssuser file

The *compssuser* file is used to display a graphic interface for choosing the package categories to install. In the following example, if we choose to select VOLIFE categories, the installer will look into the *rpmsrate* file to install all packages from CAT_XOS CAT_VOLIFE. By default all categories with the flag *default_selected* => *1* are installed.

```
N_("XtreemOS bundle") =>
[
  { label => N_("VOlife"),
    descr => N_("VOlife backend and frontend"),
    flags => [ qw(XOS VOLIFE) ],
    default_selected => 1,
  },
  { label => N_("VO support"),
    descr => N_("System Database and authentication library plug ins for 
    flags => [ qw(XOS XOS_VOSUPPORT) ],
    default_selected => 1,
  },
  { label => N_("XOS VOM"),
    descr => N_("Core classes needed to run DIXI, XATI and the services")
    flags => [ qw(XOS VOM) ],
    default_selected => 1,
  },
  { label => N_("XOS DIXI"),
    descr => N_("DIXI, XATI and the services"),
    flags => [ qw(DIXI) ],
    default_selected => 1,
  },
  { label => N_("XOS AEM"),
    descr => N_("AEM"),
    flags => [ qw(AEM) ],
    default_selected => 1,
  },
  { label => N_("Saga Framework"),
    descr => N_("The SAGA Framework"),
    flags => [ qw(SAGA) ],
    default_selected => 1,
  },
  { label => N_("XtreemOS Client"),
    descr => N_("XOS Client"),
    flags => [ qw(XOS_VOSUPPORT) ],
    default_selected => 1,
  },
  { label => N_("XtreemFS"),
    descr => N_("XtreemFS is an object-based file system"),
    flags => [ qw(XTREEMFS) ],
    default_selected => 1,
  },
  { label => N_("LinuxSSI, kernel-linuxssi"),
    descr => N_("kernel-linuxssi"),
    flags => [ qw(LINUXSSI) ],
    default_selected => 1,
  },
  { label => N_("BLCR, kernel 2.6.27"),
    descr => N_("kernel-2.6.27, blcr module"),
    flags => [ qw(BLCR) ],
    default_selected => 1,
  },
  { label => N_("Object Sharing Service release"),
    descr => N_("oss, liboss0"),
    flags => [ qw(OSS) ],
    default_selected => 1,
    },
  ],
```



PACKAGE GROUP SELECTION

XtreemOS bundle

☑ VOlife

☑ VO support

☑ XOS VOM

☑ XOS DIXI

☑ XOS AEM

☑ Saga Framework

☑ LinuxMD

☑ XtreemFS

☑ LinuxSSI, kernel-linuxssi

☑ BLCR, kernel 2.6.27

☑ Object Sharing Service release

Linux tools

☑ Useful Tools

☐ Development

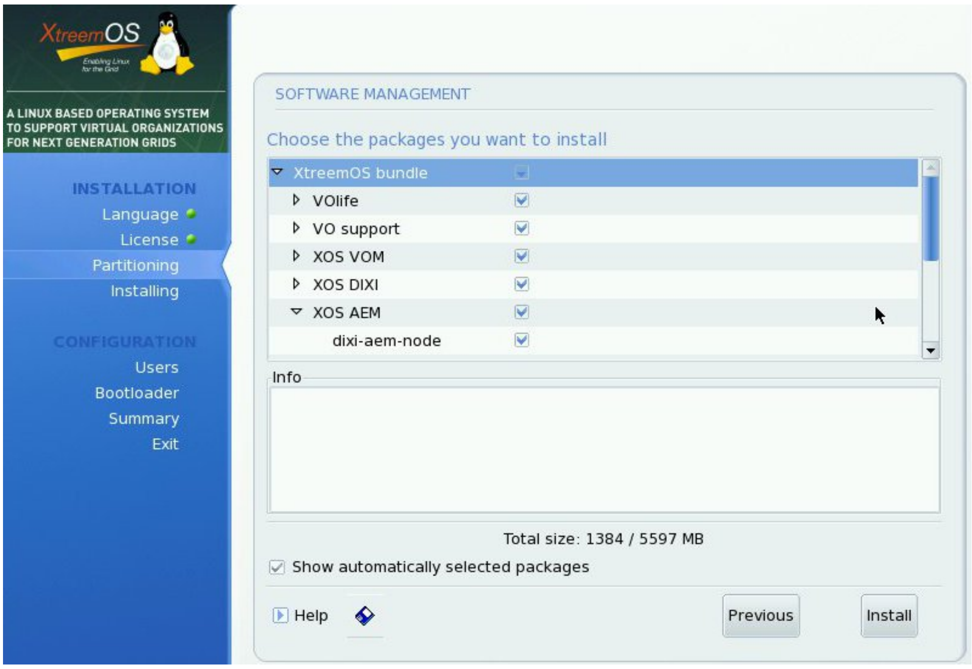
☐ LSB

Graphical Environment

☑ IceWm Desktop

☐ Individual package selection

Total size: 1382 / 5597 MB



XtreemOS
Enabling Linux for the Grid
A LINUX BASED OPERATING SYSTEM TO SUPPORT VIRTUAL ORGANIZATIONS FOR NEXT GENERATION GRIDS
INSTALLATION
Language
License
Partitioning
Installing
CONFIGURATION
Users
Bootloader
Summary
Exit

SOFTWARE MANAGEMENT

Choose the packages you want to install

XtreemOS bundle	
VOlife	☑
VO support	☑
XOS VOM	☑
XOS DIXI	☑
XOS AEM	☑
dixi-aem-node	☑

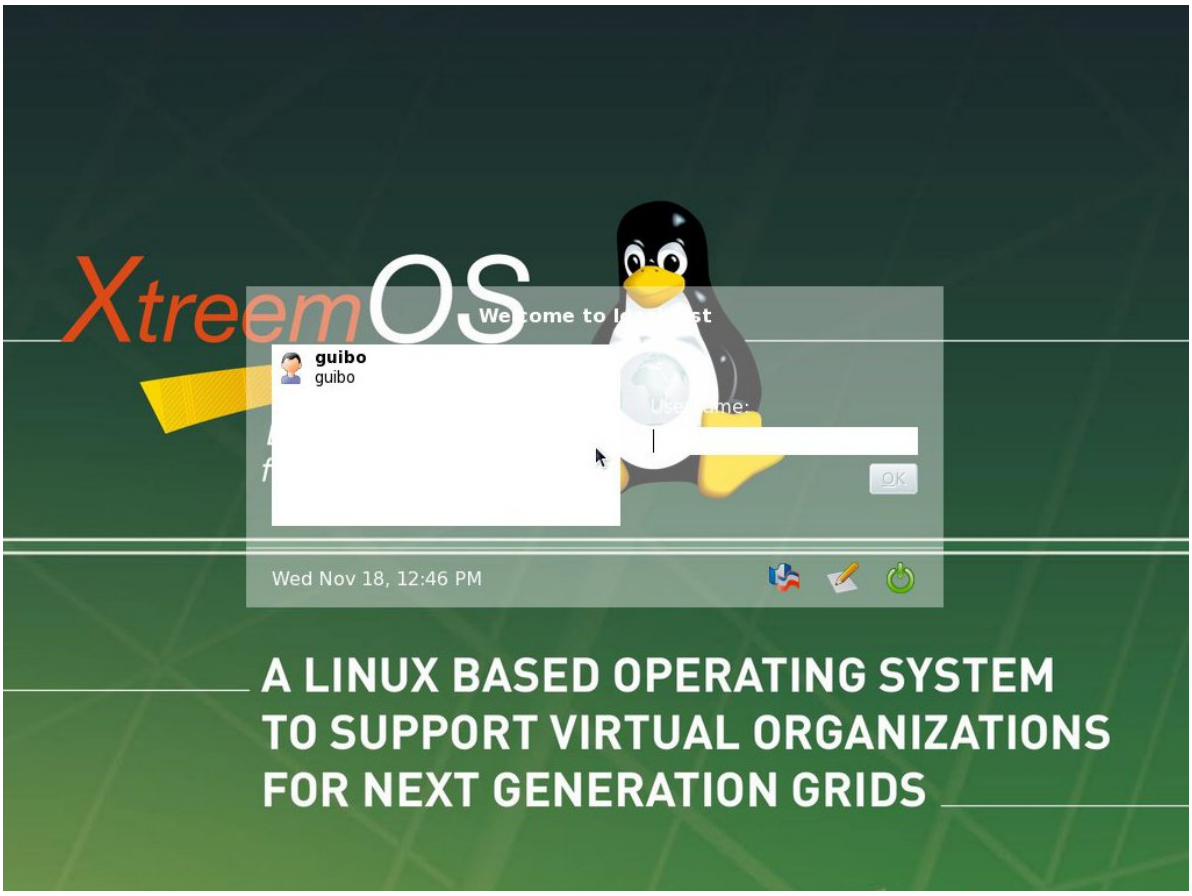
Info

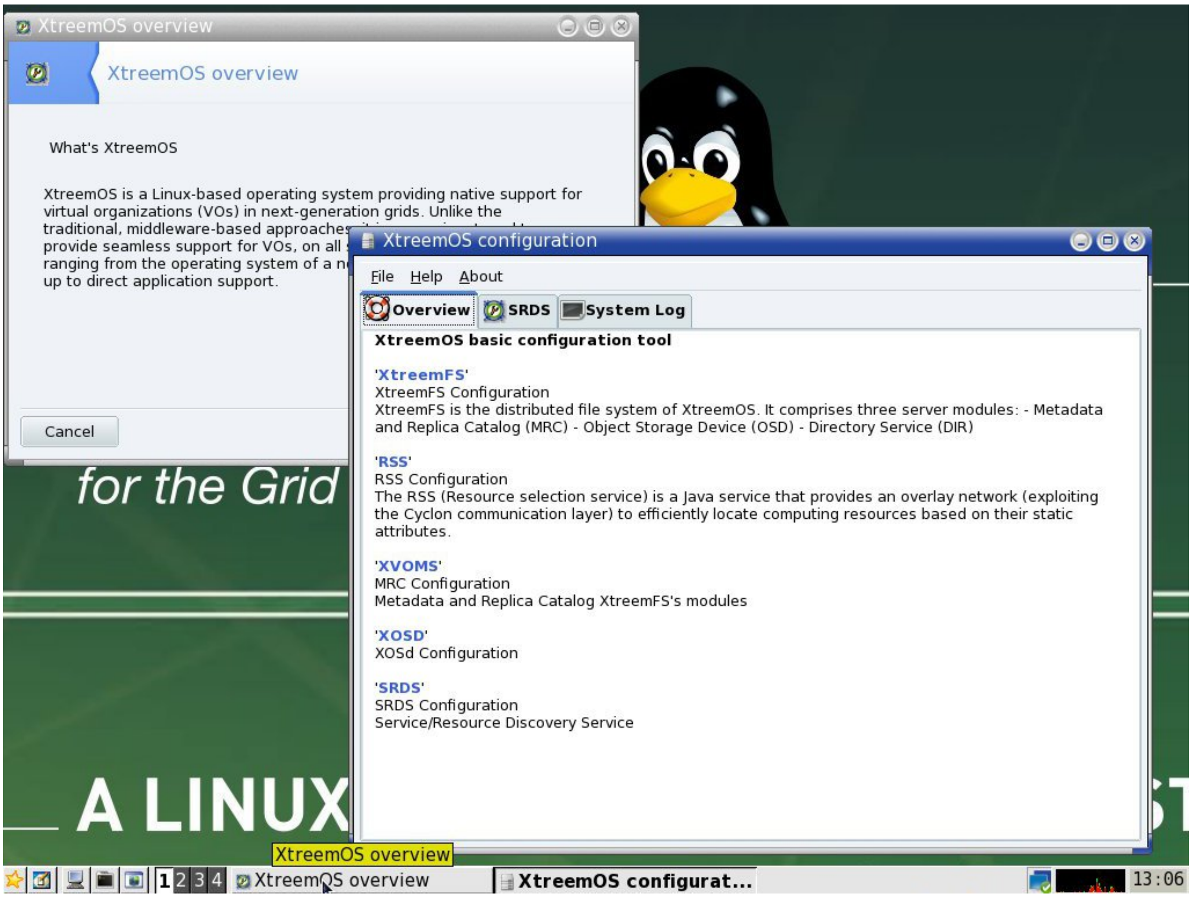
Total size: 1384 / 5597 MB

☑ Show automatically selected packages

6.5 XtreemOS window manager

We created a specific theme for this release. Specifically for the boot process and under the Window manager. We used the work done for the graphic chart of XtreemOS to create wallpaper and a splashy boot image. IceWM was chosen to be the window manager because it is powerful, quite fast, and easy to customize. Each time a user logs on the system through GDM we pop up the *drakxos* wizard which describes every XtreemOS components, and *drakxosconfig*, the basic XtreemOS GUI tool to configure XtreemOS services.


XtreemOS
We come to l st
guibo
guibo
Username:
OK
Wed Nov 18, 12:46 PM
A LINUX BASED OPERATING SYSTEM
TO SUPPORT VIRTUAL ORGANIZATIONS
FOR NEXT GENERATION GRIDS



What's XtreemOS

XtreemOS is a Linux-based operating system providing native support for virtual organizations (VOs) in next-generation grids. Unlike the traditional, middleware-based approaches
provide seamless support for VOs, on all
ranging from the operating system of a n
up to direct application support.

XtreemOS basic configuration tool

'XtreemFS'
XtreemFS Configuration
XtreemFS is the distributed file system of XtreemOS. It comprises three server modules: - Metadata and Replica Catalog (MRC) - Object Storage Device (OSD) - Directory Service (DIR)

'RSS'
RSS Configuration
The RSS (Resource selection service) is a Java service that provides an overlay network (exploiting the Cyclon communication layer) to efficiently locate computing resources based on their static attributes.

'XVOMS'
MRC Configuration
Metadata and Replica Catalog XtreemFS's modules

'XOSD'
XOSd Configuration

'SRDS'
SRDS Configuration
Service/Resource Discovery Service

The web browser Firefox has been modified to integrate predefined bookmarks specific to XtreemOS. IceWM's desktop has a menu item in the tool bar to launch the documentation in PDF format.

7 Conclusion

The XtreemOS release 2.0 was released just after the Valladodid meeting in October 2009. It is a bit late according to the scheduled agenda, but this release contains a lot of improvements and bug fixes compared to the first one. Moreover it includes new features like task-xtreemos package to automatically installed package to be able to run a service.

Next 3.0 release will provide a new script tool to easily configure all XtreemOS services and deploy them over nodes, this was a drawback on the 2.0 release.

8 References

The BCD tool:

http://svn.mandriva.com/cgi-bin/viewvc.cgi/soft/build_system/bcd/

XtreemOS subversion repository:

https://scm.gforge.inria.fr/svn/xtreemos

One of the XtreemOS mirror:

ftp://ftp.proxad.net/pub/Distributions_Linux/MandrivaLinux/devel/xtreemos

The urpmi tool:

http://wiki.mandriva.com/en/Tools/urpmi

All tools and scripts available to build the XtreemOS 2.0 release:

svn+ssh://scm.gforge.inria.fr/svn/xtreemos/distribution/build-xos-distrib

Mandriva packaging policies :

http://wiki.mandriva.com/en/Policies#Packaging_Policies

XtreemOS packages on Doc4 Mandriva :

http://doc4.mandriva.org/