''''''''''''''''''''''''''''''''!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!!!!!!

Project no. IST-033576

XtreemOS

Integrated Project

Building and Promoting a Linux-based Operating System to Support Virtual Organizations for Next Generation Grids

D4.1.5
LinuxSSI integration and packaging in Debian, Mandriva and RedFlag distribution

Due date of deliverable: 30/11/08
Actual submission date: 14/01/09

Start date of project: June 1st 2006

Type: Documentation

Name of responsible: Antoine Giniès
Editor & editor’s address: aginies@mandriva.com
nvigier@mandriva.com
Mandriva
43 rue d'aboukir
75002 Paris

Version 1.3 / Last edited by Antoine Giniès / Date 14/01/2008

Project co-funded by the European Commission within the Sixth Framework Programme
Dissemination Level

PU	Public	X
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
1	07/01/09	Antoine Giniès	Edge-IT	First release
1.1	13/01/09	Antoine Giniès	Edge-IT	Correction taking into account Samuel Kortas review
1.2	14/01/09	Wang Diancheng	Red Flag	Correct some typos
1.3	14/01/09	Antoine Giniès	Edge-IT	Add missing summary part

Reviewers:
Samuel Kortas (EDF),Wang Diancheng (Red Flag)

Tasks related to this deliverable:

Task No	Task Description	Partner Involved
T4.1.1	Packaging and integrating LinuxSSI in debian, Mandriva and Red Flag	INRIA,EDGE, Red Flag
T4.1.6	LinuxSSI + Linux-XOS resulting in LinuxSSI-XOS	INRIA,EDGE

Executive Summary

LinuxSSI is the core XtreemOS foundation layer for clusters. It is based on the Kerrighed technology, originally developed by INRIA in collaboration with EDF R&D and now with the Kerrighed open source project (www.kerrighed.org).This document describes the work that has been done since June 2008 to integrate recent changes from the OSCAR and Kerrighed projects into Kerrighed OSCAR package. We worked on packaging and integrationLinuxSSI Mandriva, and Red Flag distributions. Integration should provide a package fully operational in the system, and also an easy way to configure or use LinuxSSI. Those packages should be available for partners, and for the first public release.

Table of Contents

• LinuxSSI components

The main packages of LinuxSSI are LinuxSSI kernel-linuxssi and krg-drmaa. Other are libraries and development packages. The kernel used to build LinuxSSI is kernel 2.6.20.

1.1 LinuxSSI

This package contains tools to make a fully functional LinuxSSI cluster, like init scripts. It contains the krgadm command and various start scripts.

1.2 krg-drmaa

It's a DRMAA job submission system for LinuxSSI . This tool is used to submit jobs to LinuxSSI operating system. It was implemented as a part of work package WP2.2. This package contains a configuration file /etc/xos/config/linuxssi-drmaa/krg-drmaa-server.conf which define the default working directory in which jobs are executed.

1.3 Kernel-linuxssi

This package contains the Linux kernel and all modules patched with LinuxSSI. LinuxSSI is a Single System Image operating system for clusters. It is based on Kerrighed which offers the view of a unique SMP machine on top of a cluster of standard PCs.

2 Packaging process

2.1 Debian

At the moment Debian packages development has been stopped since the first release of LinuxSSI (which was a fork of a Kerrighed release). The main goal of LinuxSSI integration was to be able to provide an operating system and a full integration of LinuxSSI not just some packages. So we have to build and provide for a Debian system (which one: Etch, sarge, sid release ?) a full integration of LinuxSSI, in the installation process, boot process, boot configuration process, autodetection of the material. Few people in the consortium use the Debian GNU/Linux OS, so we are not able to maintain and provide up-to-date packages for this System. Moreover packaging kernel component is not as easy as packaging regular user applications, so we need to find an expert in Debian packaging.

2.2 Red Flag

Red Flag is a distribution based on Asianux Linux distribution. You can get Asianux Linux distribution at : http://www.asianux.com

The XtreemOS packages built for Red Flag are available on the ftp server here :
host : ftp://219.237.229.195/release/Asianux3.0SP1/RC
user : xtreemos
passwd : xtreemos123

The packages were built for Asianux 3.0SP1

To install these packages, you need the root permissions. Use su command to get the root permissions.

If you want to only install the binary packages of LinuxSSI, you can download the packages by using the following command :
(NOTE: You can also find these packages in the Asianux 3.0SP1's Tools CD.)

```
mkdir -pv /root/xtreemos
cd /root/xtreemos
wget
ftp://xtreemos:xtreemos123@219.237.229.195/release/Asianux3.0SP1/RC/Packaging/kerrighed-kernel-2.6.20krg-4.AXS3.i386.rpm
wget
ftp://xtreemos:xtreemos123@219.237.229.195/release/Asianux3.0SP1/RC/Packaging/kerrighed-kernel-devel-2.6.20krg-4.AXS3.i386.rpm
wget
ftp://xtreemos:xtreemos123@219.237.229.195/release/Asianux3.0SP1/RC/Packaging/kerrighed-kernel-source-2.6.20krg-4.AXS3.i386.rpm
```

You should be able to install the packages when the "wget" commands are finished. You can then install the packages using the following command :

```
rpm -ivh *.rpm
```

You will also need to do some configuration before using LinuxSSI, such as adding a "xtreemos" user account, creating the kerrighed's nodes informations. To set the kerrighed's nodes informations, follow the instructions on http://www.kerrighed.org.

3 LinuxSSI components for the first release based on a Mandriva GNU/Linux

3.1 common basesystem

XtreemOS was updated and it is now based on the stable 32bits Mandriva GNU/Linux 2008.0 distribution. Key (?) Software available on it are: gcc-4.3, gcc3.3 glibc-2.6.1. For more information about Mandriva GNU/Linux 2008.0 please refer to our website Mandriva.

3.2 the chroot

The goal of this chroot is to provide the same basesytem to all partners to avoid various glibc and compiler versions problems. It leads to error, and it will not be easy to debug all tools. Mandriva GNU/Linux distribution is used to build XtreemOS. XtreemOS must be independant of any GNU/Linux distribution to stay portable. The main goal for this period

(M30) was to provide a full operating system with all components, that's why we focus all our effort on a specific release.

You can get it here: http://people.mandriva.com/~nvigier/2008_chroot_i586.tar.bz2

Extract this chroot somewhere in your system as a root user:

```
[root@node34 /opt] tar xvfj xtreemos_chroot.DATE.tar.bz2
```

It will create a directory called xtreemos_chroot in your current directory. Now remount "bind" the subtree /dev and /proc in the chroot, so that their contents is available in both places.

```
mount -o bind /dev /path_chroot/dev
mount -o bind /proc /path_chroot/proc
```

Basic configuration

To setup your internet access use your current /etc/resolv.conf file to be able to contact your Internet Domain Name Server.

```
cp /etc/resolv.conf path_chroot/etc/resolv.conf
```

You can connect to the chroot either by using the chroot command as root, or by starting an ssh server in the chroot and connecting to this ssh server as a user.

If you use the ssh solution, edit the sshd configuration in the chroot to change the port used. The default port is 22, but this port might be already in use for the ssh server of the host operating system. Change to 24 or any other port that is available. Now start the sshd service to be able to connect to the XtreemOS chroot through the network.

You can automatically do all of that with a initscript. Create a new file in /etc/rc.d/init.d/ called xtreemos, with the following contents:

```
#! /bin/sh
#
# chkconfig: 2345 61 25
# description: start an sshd daemon on a chrooted XtreemOS system
# processname: sshd
# pidfile: /var/run/xtreemos.pid
# pidfile: $root/var/run/sshd.pid
# PLEASE adjust
root="/path_chroot"
# Get functions
. /etc/rc.d/init.d/functions
# See how we were called.
case "$1" in
  start)
        echo  -n "Starting XtreemOS environment: "
        echo
                if [ -f "/var/lock/subsys/xtreemos" ];then echo "Xtreemos envitonment
already
```

```
running, exiting"; exit 1; fi
        mount /dev/ -o bind $root/dev
        mount -a
        chroot $root mount -t proc /proc /proc
        chroot $root mount -t devpts none /dev/pts
        cp -f /etc/resolv.conf $root/etc/resolv.conf
        chroot $root service sshd start
        cp -f $root/var/run/sshd.pid /var/run/xtreemos.pid
        echo
        touch /var/lock/subsys/xtreemos
        ;;
  stop)
        echo -n "Stopping XtreemOS chrooted environment: "
        echo
        chroot $root service sshd stop
        chroot $root umount -a
        chroot $root umount /dev/pts
        chroot $root umount /proc
        umount $root/dev
           echo
           rm -f /var/lock/subsys/xtreemos
           rm -f /var/run/xtreemos.pid
           ;;
     status)
           status xtreemos
           ;;
     restart)
           $0 stop
           $0 start
           ;;
     *)
           echo "Usage: $xtreemos {start|stop|status|restart}"
           exit 1
   esac
   exit 0
```

This initscript should work under Mandriva, Red Flag and Red Hat GNU/linux distribution. For Debian GNU/Linux or other GNU/Linux distribution, contribution are welcome. ADJUST the variable root="path_chroot" to fit your configuration. It's the full path where you extract xtreemos_chroot_DATE.tar.bz2. Now root should be able to start the service, so set this file (/etc/rc.d/ini.d/xtreemos) executable:

```
chmod 755 /etc/rc.d/init.d/xtreemos
```

A user xtreemos has been created in this chroot, and the default password his x. Default password for the root user is xtreemos. Now start/stop/status the xtreemos service with /etc/rc.d/ini.d/xtreemos script, or service xtreemos command. If you want to connect to this chroot from another computer:

```
ssh -p24 xtreemos@IPADDRESS
```

To copy data under this chroot:

```
scp -P24 data/* xtreemos@IPADDRESS:~/
```

3.3 Quick overview of the URPM? Commands

To list all media now available:

```
urpmq --listmedia
```

install a package:

```
urpmi name_of_package
```

If you want to find a file in the RPM database, and you don't know in which package it is, so you don't know which package to install, use the:

```
urpmf file
```

or use:

```
urpmf bin/file
```

You can specify a list of packages which installation should be skipped. Use the file /etc/urpmi./skip.list.

3.4 Spec file

We use two spec files for LinuxSSI. One called kernel-linuxssi.spec to package the LinuxSSI kernel and all modules, and linuxssi.spec to package libraries and development tools. The first release of these spec files was based on a version available on the subversion repository of Kerrighed. The LinuxSSI tools require a specific version of the kernel-linuxssi to avoid any desynchronisation between the tools and the kernel.

Specs files and source code needed to rebuild all LinuxSSI components are available on the gforge.inria.fr repository:

```
svn co svn+ssh://scm.gforge.inria.fr/svn/xtreemos/distribution/packages/2008.0/pkg_name
```

3.5 Rebuild a package

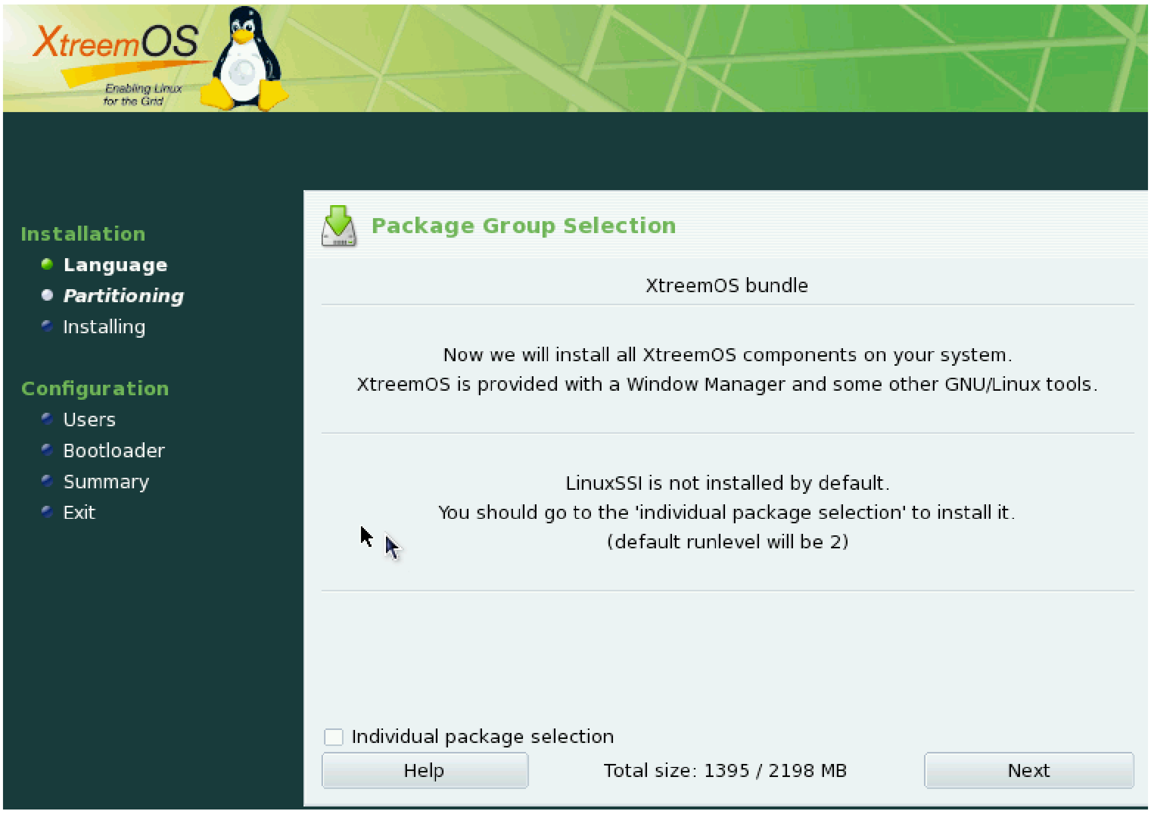
To rebuild a package, checkout from the subversion repository the LinuxSSI component you want to rebuild, and use the rpmbuild command.

```
[xtreemos@ XOS 2008 chroot kernel-linuxssi] $ svn co \ svn+ssh://scm.gforge.inria.fr/svn/
xtreemos/distribution/packages/2008.0/kernel-linuxssi
[xtreemos@ XOS 2008 chroot kernel-linuxssi] $ cd kernel-linuxssi
[xtreemos@ XOS 2008 chroot kernel-linuxssi] $ rpmbuild --define "_topdir $PWD" -ba
SPECS/kernel-linuxssi.spec
```

4 Integration in the first XtreemOS release

We have worked to fully integrate all XtreemOS components in the first release, and especially LinuxSSI which requires some modification in the graphic installer.

A message is displayed in the package selection step, stating that LinuxSSI is not installed by default, and if you choose LinuxSSI component, the system will run in runlevel 2 (single user and a few services).



Package Group Selection

XtreemOS bundle

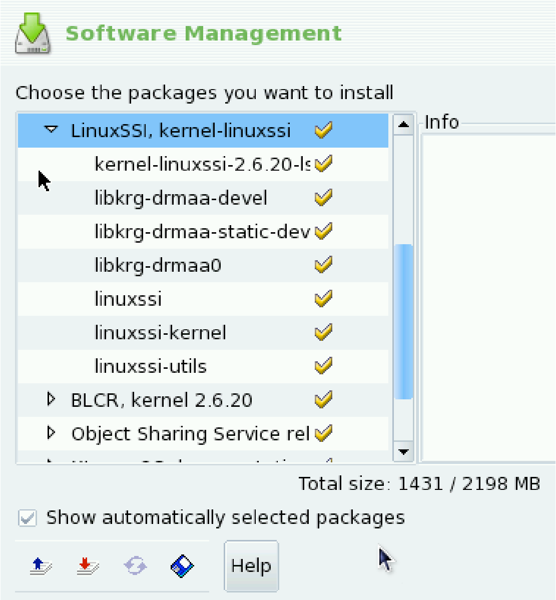
Now we will install all XtreemOS components on your system.
XtreemOS is provided with a Window Manager and some other GNU/Linux tools.

LinuxSSI is not installed by default.
You should go to the 'individual package selection' to install it.
(default runlevel will be 2)

□ Individual package selection

Total size: 1395 / 2198 MB

A LinuxSSI group has been created, to allow to selecting all LinuxSSI components in "one click".



Software Management

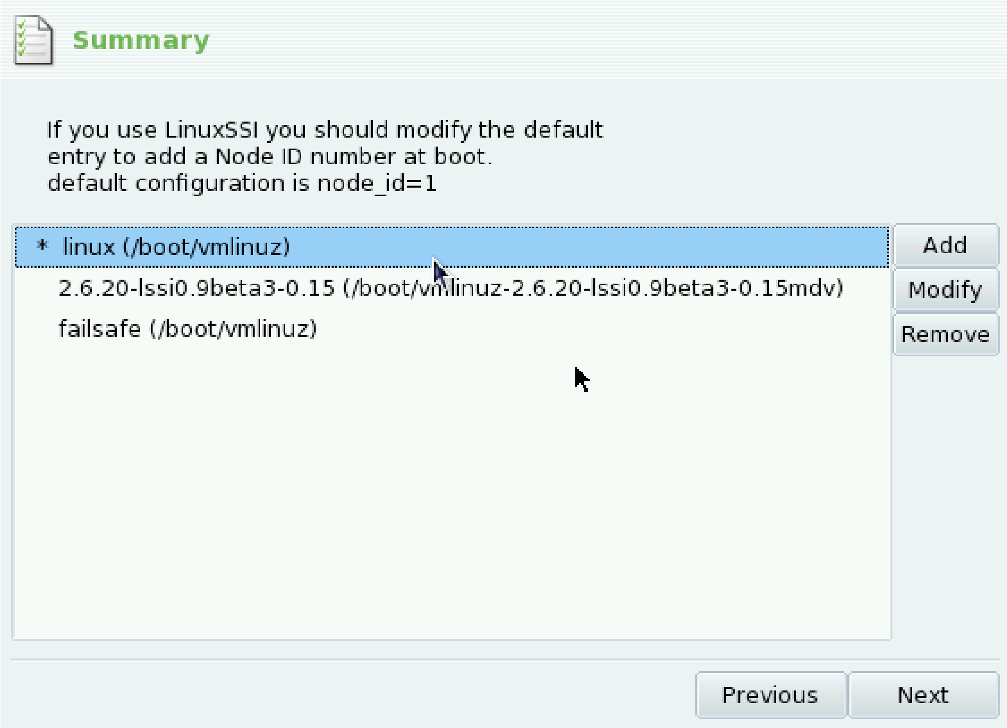
Choose the packages you want to install

- LinuxSSI, kernel-linuxssi ✓
  - kernel-linuxssi-2.6.20-ls ✓
  - libkrg-drmaa-devel ✓
  - libkrg-drmaa-static-dev ✓
  - libkrg-drmaa0 ✓
  - linuxssi ✓
  - linuxssi-kernel ✓
  - linuxssi-utils ✓
- BLCR, kernel 2.6.20 ✓
- Object Sharing Service rel ✓

Total size: 1431 / 2198 MB

☑ Show automatically selected packages

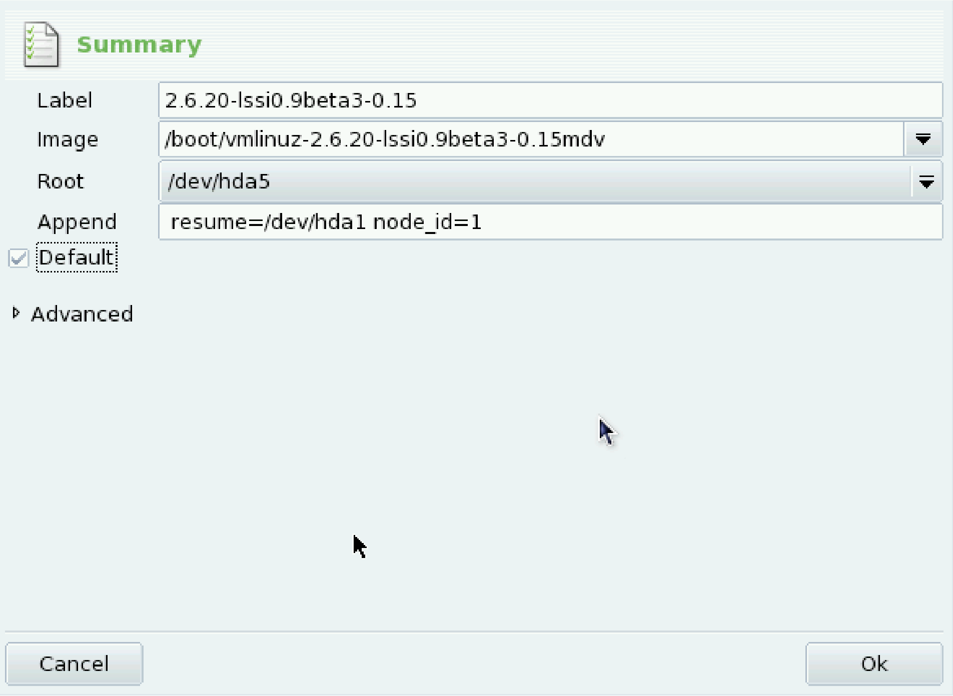
The boot configuration steps automatically appears at the end of the installation process if LinuxSSI has been installed, to be able to set the kernel by default, and set the node_id parameter.



Summary

If you use LinuxSSI you should modify the default
entry to add a Node ID number at boot.
default configuration is node_id=1

- * linux (/boot/vmlinuz)
- 2.6.20-lssi0.9beta3-0.15 (/boot/vmlinuz-2.6.20-lssi0.9beta3-0.15mdv)
- failsafe (/boot/vmlinuz)

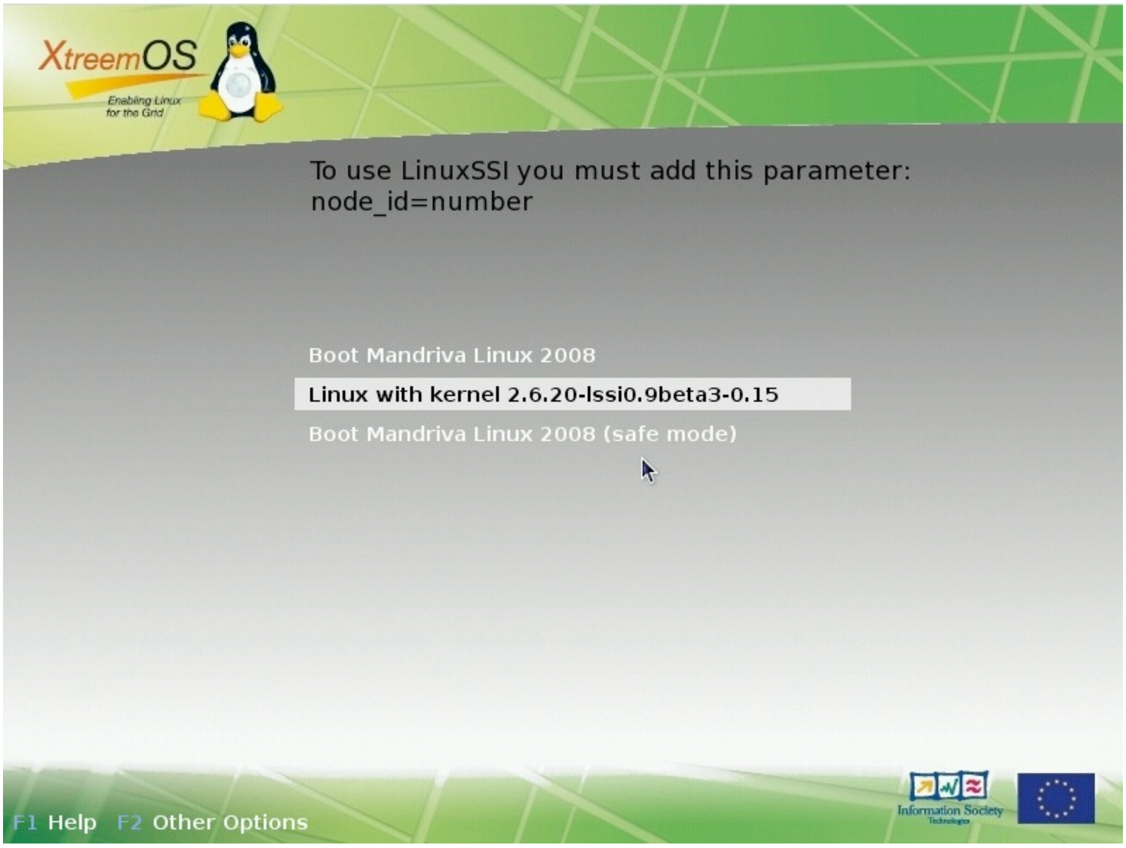


Summary

Label	2.6.20-lssi0.9beta3-0.15
Image	/boot/vmlinuz-2.6.20-lssi0.9beta3-0.15mdv
Root	/dev/hda5
Append	resume=/dev/hda1 node_id=1

☑ Default

The grub menu has been modified to add a message about the node_id parameter



To use LinuxSSI you must add this parameter:
node_id=number

XtreemOS
Enabling Linux for the Grid
Boot Mandriva Linux 2008
Linux with kernel 2.6.20-lssi0.9beta3-0.15
Boot Mandriva Linux 2008 (safe mode)
F1 Help F2 Other Options
Information Society Technologies

5 Limitation of the LinuxSSI kernel packages

LinuxSSI now has an option to be able to automatically assign a node_id to a node based on its IP address. However this step is done at boot process, and use a dhcp option of the kernel, which requires the network driver to be available in the core kernel at boot, not in modules. In order to provide support for as much hardware as possible we usually build all drivers available as modules, as it is not possible to include everything directly in the kernel. So at the moment, using modules for network devices prevents us from using the automatic assignation of node_id. The workaround to fix this limitation, is to rebuild a kernel with the needed modules to support your network interface built into the core kernel. There is no simple way to do that, because we should provide a lot of kernels due to the amount of network cards available in the market.