



WORK PACKAGE 3.2 INSTALLATION GUIDE

CROSSGRID-TEMPLATE

Task 3.2 Installation Guide

Document Filename:	CG3.2-D3.3-v1.0-CSIC-installationguide.doc
Work package:	WP3 New Grid Services and Tools
Partner(s):	UAB, CSIC, Datamat
Lead Partner:	UAB
Config ID:	CG3.x-D3.x-v1.0-CSIC-Final
Document classification:	Public

Abstract: This document is intended for Site Administrator of CrossGrid testbed resources and may also be useful for people that need detailed information about the WP3 release...



Information Society
Technologies

Delivery Slip

	Name	Partner	Date	Signature
From				
Verified by				
Approved by				

Document Log

Version	Date	Summary of changes	Author
1-0-DRAFT-A	16/1/2003	Draft Version	Álvaro Fernández
1.1	20/2/2003	Added Tests to see the specific behaviour of the modified RB	Álvaro Fernández

CONTENTS

1. OVERVIEW	6
2. PREREQUISITES.....	8
2.1. PREREQUISITES FOR BUILDING THE SOFTWARE.....	8
2.2. PREREQUISITES FOR INSTALLING THE SOFTWARE	10
2.2.1. PREREQUISITES FOR LOGGING AND BOOKEEPING SERVICES	10
2.2.2. PREREQUISITES FOR RB AND JSS	11
2.2.3. PREREQUISITES FOR INFORMATION INDEX.....	12
2.2.4. PREREQUISITES FOR THE USER INTERFACE	12
3. CROSSGRID SPECIFIC DEFAULTS.....	15
3.1. LB LOCAL-LOGGER AND LB-SERVER DEFAULTS	15
3.2. RB AND JSS DEFAULTS	15
3.3. INFORMATION INDEX DEFAULTS	15
3.4. USER INTERFACE DEFAULTS	15
4. OBTAINING THE WP3 SOFTWARE	16
5. BUILDING, INSTALLING AND CONFIGURING THE SOFTWARE, SERVICES AND TOOLS.....	17
5.1. INTRODUCTION	17
5.2. BUILD PROCEDURE.....	17
5.2.1. Installing Required Software for the Build Procedure	17
5.2.2. Setting environment variables.....	17
5.2.3. Compiling the code.....	19
5.3. INSTALLATION AND CONFIGURATION OF MODULES.....	20
5.3.1. LOGGING AND BOOKEEPING SERVICES	21
5.3.1.1. Local logger.....	21
5.3.1.1.1. Install Required Software	21
5.3.1.1.2. Install the RPMs	21
5.3.1.1.3. Configuration settings.....	21
5.3.1.1.4. Environment variables	21
5.3.1.2. LB server.....	23
5.3.1.2.1. Install Required Software	23
5.3.1.2.2. Install the RPMs	23
5.3.1.2.3. Configuration settings.....	24
5.3.1.2.4. Environment variables	24
5.3.2. RB AND JSS SERVICES.....	24
5.3.2.1. RB and JSS.....	24
5.3.2.1.1. Install Required Software	24
5.3.2.1.2. Install the RPMs	26
5.3.2.1.3. Configuration settings.....	26
5.3.2.1.4. Environment variables	26
5.3.3. INFORMATION INDEX SERVICES	28
5.3.3.1. INFORMATION INDEX.....	28
5.3.3.1.1. Install Required Software	28
5.3.3.1.2. Install the RPMs	28
5.3.3.1.3. Configuration settings.....	28
5.3.3.1.4. Environment variables	29



5.3.4. USER INTERFACE SERVICES.....	29
5.3.4.1. USER INTERFACE.....	29
5.3.4.1.1. Install Required Software.....	29
5.3.4.1.2. Install the RPMs.....	29
5.3.4.1.3. Configuration settings.....	30
5.3.4.1.4. Environment variables.....	30
6. TESTING YOUR INSTALLATION.....	32
6.1. OPERATING THE SYSTEM.....	32
6.2. LB LOCAL-LOGGER.....	32
6.2.1. Starting and stopping daemons.....	32
6.2.2. Troubleshooting.....	33
6.3. LB SERVER.....	34
6.3.1. Starting and stopping daemons.....	34
6.3.2. Purging the LB database.....	34
6.3.3. Troubleshooting.....	35
6.4. RB AND JSS.....	36
6.4.1. Starting PostGreSQL.....	36
6.4.2. Starting and stopping JSS and RB daemons.....	36
6.4.3. RB and JSS databases clean-up.....	37
6.4.4. RB troubleshooting.....	37
6.4.5. JSS troubleshooting.....	38
6.5. INFORMATION INDEX.....	38
6.5.1. Starting and stopping daemons.....	38
6.6. TESTING THE NEW FEATURES.....	38
6.7. MULTIPLE CE SELECTION.....	38
7. REFERENCES.....	42



1.OVERVIEW

Current development of Crossgrid middleware is based on the Datagrid WP1 WMS Software, with the corresponding modifications to add the new functionality required by the Crossgrid applications. The release used is EDG 1.2.2.

This guide describes the installation procedure of the Crossgrid middleware that is being developed in Task 3.2. Since the process of installation and configuration is identical to the Datagrid one, this guide is tightly related with the Datagrid WP1 WMS Software Administrator and User Guide [1], and it will continuously refer to the that document.

This document merely gives the basic steps for installation, but the Datagrid document is required for a more deeply understanding of some aspects of the installation and configuration.

The structure of this guide is as follows:

Section 2 describes the prerequisites to build and install the software, describing all the external packages that are required and giving instructions to make it easy to find and install them.

Section 3 describes the specific Crossgrid defaults that involve the installation of the software, explaining also where the software will be installed in the directory tree, etc...

Section 4 serves as a point to consult where to download the software.

Section 5 explains in detail how to build the software, and also how to install it. Each module is explained separately.

Section 6 explains how to start the different modules and test its correct behaviour.

2. PREREQUISITES

This section outlines the required packages to build, and to install and configure the software.

The best choice to install and configure the software is to install the rpms as directed in the *Installation and Configuration* section of this document. (see section)

If rpms binaries of the software are not suitable for your machines, or the source code is the only source available of the software, you will have to build the software.

2.1. PREREQUISITES FOR BUILDING THE SOFTWARE

Since Crossgrid middleware is based on Datagrid, requirements for building the software are the same. In Section 3 of [1] some requirements are outlined for the installation.

The EDG WP1 and consequently Crossgrid middleware software runs and has been tested on platforms running *Globus Toolkit 2.0 Beta Release 21* on top of *Linux RedHat 6.2*.

Hereafter are listed the software packages, apart from *Crossgrid release*, that are required to be installed locally on a given site in order to be able to build the software. They are (as shown in Section 3.1 of [1]):

- Globus Toolkit 2.0 Beta 21 or higher (download at <http://datagrid.in2p3.fr/distribution/globus/beta-21>)
- Python 2.1.1 (download at <http://datagrid.in2p3.fr/distribution/config/external.html>)
- Swig 1.3.9 (download at <http://datagrid.in2p3.fr/distribution/config/external.html>)
- Expat 1.95.1 (download at <http://datagrid.in2p3.fr/distribution/config/external.html>)
- Expat-devel 1.95.1 (download at <http://datagrid.in2p3.fr/distribution/config/external.html>)
- MySQL Version 9.38 Distribution 3.22.32, for pc-linux-gnu (i686) (download at http://datagrid.in2p3.fr/distribution/config/external_services.html)
- MySQL Version 11.15 Distribution 3.23.42, for pc-linux-gnu (i686)
(download at <http://datagrid.in2p3.fr/distribution/external/RPMS/>). Hereafter the needed rpms:
 - MySQL-shared-3.23.42-1
 - MySQL-client-3.23.42-1
 - MySQL-3.23.42-1
 - MySQL-devel-3.23.42-1
- Postgresql 7.1.3 (http://datagrid.in2p3.fr/distribution/config/external_services.html)
- Classads library (download at <http://datagrid.in2p3.fr/distribution/external/RPMS/classads-0.0-edg2.i386.rpm>)

-
- CondorG 6.3.1 for INTEL-LINUX-GLIBC21 (download at <http://datagrid.in2p3.fr/distribution/external/RPMS/CondorG-6.3.1-edg5.i386.rpm>)
 - Perl IO Stty 0.02, Perl IO Tty 0.04 (download at <http://datagrid.in2p3.fr/distribution/config/external.html>)
 - MyProxy-0.4.4 (download at <http://datagrid.in2p3.fr/distribution/external/RPMS/>). Hereafter the needed rpms:
 - myproxy-server-0.4.4-edg6.i386.rpm (for the MyProxy Server machine)
 - myproxy-client-0.4.4-edg6.i386.rpm (for the UI machine)
 - Perl 5 (download at <http://datagrid.in2p3.fr/distribution/config/external.html>)
 - gcc version 2.95.2
 - GNU make version 3.78.1 or higher
 - GNU autoconf version 2.13
 - GNU libtool 1.3.5
 - GNU automake 1.4
 - GNU m4 1.4 or higher
 - RPM 3.0.5
 - sendmail 8.11.6

2.2.PREREQUISITES FOR INSTALLING THE SOFTWARE

The compiled software requires of some external packages to run properly in the installed and configured machines. What comes next is a summary of the presented in section 4 of [1]. Please refer to that guide for a more detailed explanation of the software requirements.

2.2.1.PREREQUISITES FOR LOGGING AND BOOKEEPING SERVICES

LB services can be split into two components:

☞ **LB local logger:** responsible for accepting messages from their sources and forwarding them to the LB server.

The only required software for this module is the Globus Toolkit 2.0 (actually only GSI rpms are needed). Globus 2 rpms are available at <http://datagrid.in2p3.fr/distribution/globus> under the directory *beta-xx/RPMS* (recommended beta is 21 or higher). All rpms can be downloaded with the command

```
wget -nd -r <URL>/<rpm name>
```

and installed with

```
rpm -ivh <rpm name>
```

☞ **LB Server:** responsible for accepting messages from the LB local logger services, saving them on permanent storage and supporting queries about those messages, generated by the consumer API.

The only required software for this module is the Globus Toolkit 2.0 (actually only GSI rpms are needed). Globus 2 rpms are available at <http://datagrid.in2p3.fr/distribution/globus> under the directory *beta-xx/RPMS* (recommended beta is 21 or higher). All rpms can be downloaded with the command

```
wget -nd -r <URL>/<rpm name>
```

and installed with

```
rpm -ivh <rpm name>
```

Besides Globus Toolkit 2.0 for the LB server, it is also necessary to install MySQL Distribution 3.22.31 or higher.

At least packages MySQL-3.22.32 and MySQL-client-3.22.22 have to be installed for creating and configuring the LB database.

The rpm of MySQL Ver 9.38 Distribution 3.22.32, for pc-linux-gnu (i686) is available at http://datagrid.in2p3.fr/distribution/config/external_services.html

More information about configuration of MySQL for the LB Server can be found on section 4.1.1.2 of [1].

The *LB local-logger* services must be installed on all the machines hosting processes pushing information into the LB system, i.e. the machines running RB and JSS, and the gatekeeper machine of the CE. An exception is the submitting machine (i.e. the machine running the User Interface) on which this component can be installed but is not mandatory.

LB Server services need to be installed only on a server machine that usually coincides with the RB server one.

2.2.2. PREREQUISITES FOR RB AND JSS

The Resource Broker is the component of the Datagrid WMS that has been mainly modified in CrossGrid. The requirements are the same as outlined in section 4.2.1 of [1]:

For the installation of RB and JSS the Globus Toolkit 2.0 rpms available at <http://datagrid.in2p3.fr/distribution/globus> under the directory *beta-xx/RPMS* (recommended beta is 21 or higher) are required to be installed on the target platform. All needed rpms can be downloaded with the command

```
wget -nd -r <URL>/<rpm name>
```

and installed with

```
rpm -ivh <rpm name>
```

The Globus gridftp server package must also be installed and configured on the same host (see <http://marianne.in2p3.fr/datagrid/documentation/EDG-Install-HOWTO.html> for details).

It is important to recall that the Globus *gridmap file* located in */etc/grid-security* on the RB server machine must be filled with the certificate subjects of all the users allowed to use the Resource Broker functionalities. Users being mapped into the *gridmap file* have to belong to a group having the same name of the user itself. At the same time the dedicated user *dguser* has to belong to *all* these groups.

Moreover on the same platform the following products are expected to be installed:

–**LB local-logger services (see section)**

–**PostgreSQL (RB and JSS) :**

- Required PostgreSQL version is 7.1.3 or higher. The following packages need to be installed (keeping the order in which they are listed): postgresql-libs, postgresql-devel, postgresql, postgresql-server, postgresql-tcl, postgresql-tk and postgresql-docs.

PostgreSQL also needs packages cyrus-sasl-1-5-11 (or higher), openssl-0.9.5a and openssl-devel-0.9.5a (or higher). All of them can be found at the following URL:

<http://datagrid.in2p3.fr/distribution/external/RPMS>

–**Condor-G (JSS)**

Condor-G release required by JSS is CondorG 6.3.1 for INTEL -LINUX-GLIBC21. The Condor-G installation toolkit can be found at the following URL:

<http://www.cs.wisc.edu/condor/downloads/condorg.license.html>.

whilst it is available in rpm format (to be installed as root) at:

<http://datagrid.in2p3.fr/distribution/external/RPMS>

–**ClassAd library (RB and JSS)**

The ClassAd release required by JSS and RB is classads -0.9 (or higher). The ClassAd library documentation can be found at the following URL:

<http://www.cs.wisc.edu/condor/classad>.

whilst it is available in rpm format (to be installed as root) at:

<http://datagrid.in2p3.fr/distribution/external/RPMS>

–**ReplicaCatalog from the WP2 distribution (RB)**

The ReplicaCatalog release required by RB is ReplicaCatalogue-gcc32dbg-2.0 (or higher) that is available in rpm format (to be installed as root) at:

<http://datagrid.in2p3.fr/distribution/wp2/RPMS>

2.2.3. PREREQUISITES FOR INFORMATION INDEX

The Information Index (II) is the service queried by the Resource Broker to get information about resources for the submitted jobs during the matchmaking process. An II must hence be deployed for each RB/JSS instance.

For installing the II, apart from the *informationindex* and the *informationindex-profile* rpms the following Globus Toolkit 2.0 and Datagrid rpms are needed:

```
-globus_ssl_utils-gcc32dbg_rtl          version >= 2.1
-globus_gram_reporter-noflavor_data     version >= 2.0
-globus_gss_assist-gcc32dbg_rtl        version >= 2.0
-globus_libtool-gcc32dbgpthr_rtl       version >= 1.4
-globus_openssl-gcc32dbg_rtl           version >= 0.9.6b
-globus_openssl-gcc32dbg_pgm            version >= 2.0.14
-globus_libtool-gcc32dbg_rtl           version >= 1.4
-globus_openssl-gcc32dbgpthr_rtl       version >= 0.9.6b
-globus_openssl-gcc32dbg_rtl           version >= 2.0.14
-globus_mds_back_giis-gcc32dbg_pgm     version >= 0.3
-globus_mds_gris-noflavor_data         version >= 2.2
-globus_cyrus_sasl-gcc32dbg_rtl        version >= 1.5.27
-globus_cyrus_sasl-gcc32dbgpthr_rtl    version >= 1.5.27
-globus_gssapi_gsi-gcc32dbg_rtl        version >= 2.0
-globus_openssl-gcc32dbgpthr_rtl       version >= 2.0.14
-edg-info-main                          version >= 1.0.0
```

The above listed rpms are available at <http://datagrid.in2p3.fr/distribution/globus> under the directory *beta-xx/RPMS* (recommended beta is 21 or higher) and at <http://datagrid.in2p3.fr/distribution/datagrid/wp6>.

All the needed packages can be downloaded with the command

```
wget -nd -r <URL>/<rpm name>
```

and installed with

```
rpm -ivh <rpm name>
```

2.2.4. PREREQUISITES FOR THE USER INTERFACE

This section describes the steps needed to install and configure the User Interface, which is the software module of the WMS allowing the user to access main services made available by the components of the scheduling sub-layer.

In order to install the UI, apart from the *userinterface* and *workload-profile* rpms (see Install section for details) you will need the following Globus Toolkit 2.0 and Datagrid rpms available respectively at <http://datagrid.in2p3.fr/distribution/globus> and <http://datagrid.in2p3.fr/distribution/datagrid/wp6>:

```
-globus_gss_assist-gcc32dbgpthr_rtl-2.0-21
-globus_gssapi_gsi-gcc32dbgpthr_rtl-2.0-21
-globus_ssl_utils-gcc32dbgpthr_rtl-2.1-21
-globus_gass_transfer-gcc32dbg_rtl-2.0-21
-globus_openssl-gcc32dbgpthr_rtl-0.9.6b-21
-globus_ftp_control-gcc32dbg_rtl-1.0-21
-globus_user_env-noflavor_data-2.1-21
-globus_gss_assist-gcc32dbg_rtl-2.0-21
-globus_gssapi_gsi-gcc32dbg_rtl-2.0-21
-globus_ftp_client-gcc32dbg_rtl-1.1-21
-globus_ssl_utils-gcc32dbg_rtl-2.1-21
-globus_ssl_utils-gcc32dbg_pgm-2.1-21
-globus_gass_copy-gcc32dbg_rtl-2.0-21
-globus_gass_copy-gcc32dbg_pgm-2.0-21
-globus_openssl-gcc32dbg_rtl-0.9.6b-21
-globus_common-gcc32dbg_rtl-2.0-21
-globus_profile-edgconfig-0.9-1
-globus_io-gcc32dbg_rtl-2.0-21
-globus_core-edgconfig-0.6-2
-obj-globus-1.0-4.edg
-globus_cyrus_sasl-gcc32dbgpthr_rtl-1.5.27-21
-globus_libtool-gcc32dbgpthr_rtl-1.4-21
-globus_mds_common-gcc32dbg_pgm-2.2-21
-globus_openldap-gcc32dbg_pgm-2.0.14-21
-globus_openldap-gcc32dbgpthr_rtl-2.0.14-21
-globus_core-gcc32dbg_pgm-2.1-21
```

Moreover the set of security configuration rpm's for all the Certificate Authorities in Testbed1 available at <http://datagrid.in2p3.fr/distribution/datagrid/security/RPMS/> have to be installed together with the rpm to be used for renewing your certificate for your CA. This is available at <http://datagrid.in2p3.fr/distribution/datagrid/security/RPMS/local/>.

The Python interpreter, version 2.1.1 has also to be installed on the submitting machine. The rpm for this package is available at <http://datagrid.in2p3.fr/distribution/external/RPMS> as:

```
-python-2.1.1-3.i386.rpm
```

Information about Python and the package sources can be found at www.python.org.

Since the Linux RH 6.2 and RH 7.2 distribution already encompasses Python -1.5 installed and the recent standard Python2 rpms from RedHat and from python.org avoid the conflict with previous versions by only create python2* binaries, the UI scripts use "python2" executable as Python

interpreter. Before using the UI commands it is hence important to check that the “python2” executable is available on the submission platform and if it is not the case the necessary symbolic link should be created.

All the needed packages can be downloaded with the command

```
wget -nd -r <URL>/<rpm name>
```

and installed with

```
rpm -ivh <rpm name>
```

3.CROSSGRID SPECIFIC DEFAULTS

Each module of this Crossgrid software release is installed in different places depending of its nature. For more information including a detailed directory tree of the installation, see section 4 of [1].

3.1.LB LOCAL-LOGGER AND LB-SERVER DEFAULTS

This module is not modified in this Cross grid release, so it is the same as the EDG.

Software is installed by default in the “*/opt/edg*” directory, and some init scripts are stored in “*/etc/rc.d/init.d*” to start the services.

3.2.RB AND JSS DEFAULTS

This is the main modified module of EDG to satisfy the requirements of the Crossgrid development.

By default the software installs in the “*/opt/edg*” directory, and some other files are stored in the “*/etc/sysconfig*” path. Other init scripts to start and stop the RB and JSS processes are contained in “*/etc/rc.d/init.d*”.

The directory *bin* (under the installation path) contains all the RB and JSS server process executables *Rbserver*, *jssserver* and *jsspaser*. In *etc* are stored the configuration files (see next points).

3.3.INFORMATION INDEX DEFAULTS

This module is not modified in this Crossgrid release, so it is the same as the EDG.

Software is installed by default in the “*/opt/edg*” directory, and some init script are stored in “*/etc/rc.d/init.d*” to start the services.

Under the installation path in *etc* are stored the configuration files and *var* (initially empty) is used by the II to store files created at start-up, containing *args* and *pid* of the II process. The *information_index* script file can be used both from */etc/rc.d/init.d* and *<install-path>/sbin* to start the II.

3.4.USER INTERFACE DEFAULTS

By default the software installs in the “*/opt/edg*” directory.

The executables of the user interface can be found under the “*bin*” directory, under the installation path

4.OBTAINING THE WP3 SOFTWARE

The easiest way to obtain the software is to access the CVS repository at KFZ (gridportal.kfz.de) and to get the source code for the task3.2.

Rpm files with the compiled code are expected to reside at the central software repository (<http://gridportal.fzk.de/distribution/crossgrid/releases>); when the Complete CrossGrid releases are available to download, after the first Santiago de Compostela meeting, to be held next February.

5. BUILDING, INSTALLING AND CONFIGURING THE SOFTWARE, SERVICES AND TOOLS

5.1. INTRODUCTION

The next section will provide a guide for conveniently build, install and configure all the modules that form the distribution.

First is presented a general build procedure for building all the source code tree, and afterwards is presented each module, detailing a procedure for building the module alone, installing and configuring to make it work properly.

5.2. BUILD PROCEDURE

Following the guidelines described at section 3 of [1] we present a build procedure for all the source tree.

5.2.1. Installing Required Software for the Build Procedure

Required software for the compilation is detailed in section of this document. After installing all the required packages for the compilation, it is necessary to set some environment variables.

5.2.2. Setting environment variables

Before starting the compilation, some environment variables related to the WMS components can be set or configured by means of the *configure* script. This is needed only if package defaults are not suitable. Involved variables are listed below:

-GLOBUS_LOCATION	base directory of the Globus installation The default path is <i>/opt/globus</i> .
-MYSQL_INSTALL_PATH	base directory of the MySQL installation The default path is <i>/usr</i> .
-EXPAT_INSTALL_PATH	base directory of the Expat installation. The default path is <i>/usr</i> .
-GDMP_INSTALL_PATH	base directory of the Gdmp installation The default path is <i>/opt/edg</i> .
-PGSQL_INSTALL_PATH	base directory of the Pgsq1 installation. The default path is <i>/usr</i> .
-CLASSAD_INSTALL_PATH	base directory of the Classad library installation. The default path is <i>/opt/classads</i> .
-CONDORG_INSTALL_PATH	base directory of the Condor installation. The default path is <i>/opt/CondorG</i> .

- PYTHON_INSTALL_PATH base directory of the Python installation.
The default path is */usr*.

- SWIG_INSTALL_PATH base directory of the Swig installation .
The default path is */usr/local*.

- MYPROXY_INSTALL_PATH base directory of the MyProxy installation .
The default path is */usr/local*.

In order to build the *whole* WP1 package, all the environment variables in the previous list must be set. To build the *User Interface* module, the environment variables that need to be set are the following:

- GLOBUS_LOCATION
- CLASSAD_INSTALL_PATH
- PYTHON_INSTALL_PATH
- SWIG_INSTALL_PATH
- EXPAT_INSTALL_PATH

If you plan to build the *Job Submission* and *Resource Broker* module, variables to be set are:

- GLOBUS_LOCATION
- MYSQL_INSTALL_PATH
- EXPAT_INSTALL_PATH
- GDMP_INSTALL_PATH
- PGSQL_INSTALL_PATH
- CLASSAD_INSTALL_PATH
- CONDORG_INSTALL_PATH

If you plan to build the *Proxy* module, variables to be set are:

- GLOBUS_LOCATION
- MYPROXY_INSTALL_PATH
-

The *LB server* and *Local Logger* modules need the following environment variables to be built:

- GLOBUS_LOCATION
- MYSQL_INSTALL_PATH
- EXPAT_INSTALL_PATH

Finally, the *LB library* module needs:

- GLOBUS_LOCATION

-EXPAT_INSTALL_PATH

and the *Information Index* module only:

-GLOBUS_LOCATION

5.2.3. Compiling the code

The first thing to do to compile the software is to download the source tree, and change into the *Workload* directory. Following the guidelines of section 3.2.2 of [1] the next steps are:

1. *Configure the machine-dependent variables for the compilation:*

Run the following commands:

```
./recursive-autogen.sh
```

At this point the *configure* command can be run. The configure script has to be invoked as follows:

```
./configure <options>
```

For compiling all the code with the standard configuration you can type the command without options. The list of options, including options for modifying the installation path and defining the path for locating the required software, can be found again in section 3.2.2 of [1].

During the configure step, 12 spec files (i.e. *wl-userinterface.spec*, *wl-locallogger.spec*, *wl-lbserver.spec*, *wl-logging_dev.spec*, *wl-jss_rb.spec*, *wl-information.spec*, *wl-userinterface-profile.spec*, *wl-jss_rb-profile.spec*, *wl-information-profile.spec*, *wl-lbserver-profile.spec* and *wl-locallogger-profile.spec*, *wl-workload-profile.spec*) are created in the following source sub-directories to produce a flavour specific version:

- Workload/UserInterface
- Workload/Proxy
- Workload/Logging
- Workload/JobSubmission
- Workload/InformIndex
- Workload

2 *Compile the code*

Once the configure script has terminated its execution, check that the *make* from the GNU distribution is in your path and then always in the *Workload* source code directory run:

```
make
```

then:

```
make apidoc
```

and then:

```
make check
```

to build the test code. If the two previous steps complete successfully, the installation of the software can be performed. In order to install the package in the installation directory specified either by the `--prefix` option of the `configure` script or by the default value (i.e. `/opt/edg`), you can now issue the command:

```
make install
```

It is possible to run "make clean" to remove object files, executable files, library files and all the other files that are created during "make" and "make check". The command:

```
make -i dist
```

can be used to produce in the `workload-X.Y.Z` directory, located in the Workload's base directory, a binary gzipped tar ball of the Workload distribution. This tar ball can be both transferred on other platforms and used as source for the RPM creation.

For creating the RPMs for Workload 1.0 (according to the configure options you have used) make sure that your `PATH` is set in such a way that the `GNU autotools`, `make` and the `gcc` compiler can be used and edit the file `$(HOME)/rpmmacros` (if this file does not exist in your home directory, then you have to create it) to set the following entry:

```
%_topdir      <your home dir>/rpm/redhat
```

Then you can issue the command:

```
make rpm
```

that generates the RPMs in `$(HOME)/rpm/redhat/RPMS`.

5.3.INSTALLATION AND CONFIGURATION OF MODULES.

5.3.1.LOGGING AND BOOKEPING SERVICES

From the installation point of view LB services can be split in two main components:

- The LB services responsible for accepting messages from their source s and forwarding them to the logging and/or bookkeeping servers, which we will refer as *LB local-logger* services.
- The LB services responsible for accepting messages from the *LB local-logger* services, saving them on their permanent storage and supporting q ueries generated by the consumer API, that we will refer as *LB server* services.

The *LB local-logger* services must be installed on all the machines hosting processes pushing information into the LB system, i.e. the machines running RB and JSS, and the gatekeeper machine of the CE. An exception is the submitting machine (i.e. the machine running the User Interface) on which this component can be installed but is not mandatory:

The *LB server* services need to be installed only on a server machine that usually coincides with the RB server one.

5.3.1.1.Local logger

The steps to install the software are the following:

5.3.1.1.1.Install Required Software

5.3.1.1.1.1.Install the required software described in section .

5.3.1.1.2.Install the RPMs

Install the rpms related to the *local-logger* (note: names can vary depending of the standardization of names that will be set in the Santiago de Compostela meeting)

```
rpm -ivh workload-profile.X.Y.Z-K.i386.rpm
rpm -ivh locallogger-X.Y.Z-K.i386.rpm
rpm -ivh locallogger-profile-X.Y.Z-K.i386.rpm
```

5.3.1.1.3.Configuration settings

Both the *LB local-logger* and *LB server* have no configuration files so no action is needed for this task.

5.3.1.1.4.Environment variables

All LB components need the following environment variables to be set:

-X509_USER_KEY	the user private key file path
-X509_USER_CERT	the user certificate file path
-X509_CERT_DIR	the trusted certificate directory and ca-signing-policy directory
-X509_USER_PROXY	the user proxy certificate file path as required by GSI.

However, in case of LB daemons, the recommended way for specifying location of security files is using `--cert`, `--key`, `--CAdir` options explicitly.

The Logging library i.e. the library that is linked into UI, RB, JSS and Jobmanager, reads its immediate logging destination from the variable `DGLOG_DEST`.

It defaults to `"x-dglog://localhost:15830"` which is the correct value, hence it normally does not need to be set but on the submitting machine. Correct format for this variable is:

```
DGLOG_DEST=x-dglog://HOST:PORT
```

where as already mentioned `HOST` defaults to `localhost` and `PORT` defaults to `15830`.

On the submitting machine if the variable is not set, it is dynamically assigned by the UI with the value:

```
DGLOG_DEST=x-dglog://<LB_CONTACT>:15830
```

where `LB_CONTACT` is the hostname of the machine where the LB server currently associated to the RB used for submitting jobs is running.

The Logging library functions timeout is read from the environment variable `DGLOG_TIMEOUT`. It defaults to 2 seconds that is the correct value for local logging. On the submitting machine the value for this variable is set dynamically by the UI to 10 seconds (recommended value for non-local logging is 10 to 15 seconds) and it is anyway configurable through the UI configuration.

Finally, there is `LBDB`, the environment variable needed by the LB Server daemons (`ileventd`, `bkservice` and `bkpurge`). `LBDB` represents the MySQL database connect-string, defaults to `"lbserver/@localhost:lbserver"` and in the recommended set-up (see section) does not need to be set. Otherwise it should be set as follows:

```
LBDB=USER_NAME/PASSWORD@DB_HOSTNAME:DB_NAME
```

where

- `USER_NAME` is the name of database user,
- `PASSWORD` is user password for the database
- `DB_HOSTNAME` is the name of the host where the database is located
- `DB_NAME` is the name of the database.

5.3.1.2.LB server

The steps to install the software are the following:

5.3.1.2.1.Install Required Software

5.3.1.2.1.1.Install the required software described in section

Some configuration settings are required for MySQL:

At least packages MySQL-3.22.32 and MySQL-client-3.32.22 have to be installed for creating and configuring the LB database.

LB server stores the logging data in a MySQL database that must hence be created. The following assumes the database and the server daemons (bkserver and ileventd) run on the same machine, which is considered to be secure, i.e. no database authentication is used. In a different set-up the procedure has to be adjusted accordingly as well as a secure database connection (via ssh tunnel etc.) established. The action list below contains placeholders DB_NAME and USER_NAME, real values have to be substituted. They form the database connection string required on some LB daemons invocation. Suggested value for both DB_NAME and USER_NAME is 'lbserver', this value is also the compiled-in default (i.e. when used, the database connection string needn't be specified at all).

The following needed steps require MySQL root privileges:

1)Create the database:

```
mysqladmin -u root -p create DB_NAME
```

where DB_NAME is the name of the database.

2)Create a dedicated LB database user:

```
mysql -u root -p -e 'grant create,drop,select,insert, \
update,delete on DB_NAME.* to USER_NAME@localhost'
```

where USER_NAME is the name of the user running the *LB server* daemons.

3)Create the database tables:

```
mysql -u USER_NAME DB_NAME < server.sql
```

where *server.sql* is a file containing sql commands for creating needed tables. *server.sql* can be found in the directory “<install path>/etc” created by the *LB server* rpm installation.

5.3.1.2.2.Install the RPMs

Install the rpms related to the *LB server* (note: names can vary depending of the standardization of names that will be set in the Santiago de Compostela meeting)

```
rpm -ivh workload-profile.X.Y.Z-K.i386.rpm
rpm -ivh lbserver-X.Y.Z-K.i386.rpm
```

```
rpm -ivh lbserver-profile-X.Y.Z-K.i386.rpm
```

5.3.1.2.3. Configuration settings

Both the *LB local-logger* and *LB server* have no configuration files so no action is needed for this task.

5.3.1.2.4. Environment variables

All LB components need the following environment variables to be set:
See previous section of the local-logger environment variables ()

5.3.2. RB AND JSS SERVICES

The Resource Broker and the Job Submission Services are the WMS components allowing the submission of jobs to the CEs. They are dealt with together since they always reside on the same host and consequently are distributed by means of a single rpm.

5.3.2.1. RB and JSS

5.3.2.1.1. Install Required Software

5.3.2.1.1.1. Install the required software described in section . Details about configuration of the required software follow:

–**LB local-logger services (see section)**

–**PostgreSQL (RB and JSS) :**

Required PostgreSQL version is 7.1.3 or higher. The following packages need to be installed (respecting the order in which they are listed): postgresql-libs, postgresql-devel, postgresql, postgresql-server, postgresql-tcl, postgresql-tk and postgresql-docs.

PostgreSQL also needs packages cyrus-sasl-1-5-11 (or higher), openssl-0.9.5a and openssl-devel-0.9.5a (or higher). All of them can be found at the following URL:

<http://datagrid.in2p3.fr/distribution/external/RPMS>

Hereafter are reported the configuration options that must be used when installing the package:

```
--with-CXX  
--with-tcl  
--enable-odbc
```

Postgresql 7.1.3 is also available in rpm format (to be installed as root) at the URL :

<http://datagrid.in2p3.fr/distribution/external/RPMS>

Once PostgreSQL has been installed, you need as root to create a new system account *dguser* using the (RH specific) command

```
adduser -r -m dguser
```

This command allows indeed creating a system account having a home directory. Then follow the steps reported here to create an empty database for JSS:

```

su - postgres                (become the postgres user)
createuser -d -A dguser      (create the new database user dguser)
su - dguser                 (become the user dguser)
createdb <DBNAME>          (create the new database for JSS)

```

The name of the created database must be the same as the one assigned to the *Database_name* attribute in file *jss.conf* (see section 4.2.4.2 of [1] for more details), otherwise JSS will use as default the "*template1*" database. Avoiding use of the template database is anyway strongly recommended.

The RB server uses instead another database named "*rb*", which is created by RB itself.

– Condor-G (JSS)

Installation and configuration are quite straightforward and for details the reader can refer to the README file included in the Condor-G package. Main steps to be performed after having unpacked the package as root are:

- become *dguser* (su - dguser)
- make sure the directory where you are going to install CondorG is owned by *dguser*
- make sure the Globus Toolkit 2.0 has been installed on the platform
- run the */opt/CondorG/setup.sh* installation script
- remove the link *~dguser/.globus/certificates* created by the installation script

Moreover some additional configuration steps have to be performed in the Condor configuration file pointed to by the *CONDOR_CONFIG* environment variable set during installation. In the *\$CONDOR_CONFIG* file the following attributes need to be modified:

```

RELEASE_DIR      = $(CONDORG_INSTALL_PATH)
CONDOR_ADMIN     = <a valid e-mail address of the Condor-G administrator>
UID_DOMAIN       = < the domain of the machine (e.g. pd.infn.it)>
FILESYSTEM_DOMAIN = < the domain of the machine (e.g. pd.infn.it)>
HOSTALLOW_WRITE  = *
CRED_MIN_TIME_LEFT = 0
GLOBUSRUN = $(GLOBUS_LOCATION)/bin/globusrun

```

and the following entries need to be added:

```

SKIP_AUTHENTICATION      = YES
AUTHENTICATION_METHODS   = CLAIMTOBE
DISABLE_AUTH_NEGOTIATION = TRUE
GRIDMANAGER_CHECKPROXY_INTERVAL = 600
GRIDMANAGER_MINIMUM_PROXY_TIME = 180

```

The environment variable *CONDORG_INSTALL_PATH* is also set during installation and points to the path where the Condor-G package has been installed.

The current version of Condor-G for working properly requires file */etc/grid-security/certificates/ca-signing-policy.conf* that has been instead eliminated from the Globus Toolkit 2.0 distribution and

must hence be created by the administrator. This need will be removed with next release of Condor -G that will be fully Globus Toolkit 2.0 compliant.

–ClassAd library (RB and JSS)

The ClassAd release required by JSS and RB is classads-0.9 (or higher). The ClassAd library documentation can be found at the following URL:

<http://www.cs.wisc.edu/condor/classad>.

whilst it is available in rpm format (to be installed as root) at:

<http://datagrid.in2p3.fr/distribution/external/RPMS>

There is no special configuration information to be detailed here.

–ReplicaCatalog from the WP2 distribution (RB)

The ReplicaCatalog release required by RB is ReplicaCatalogue-gcc32dbg-2.0 (or higher) that is available in rpm format (to be installed as root) at:

<http://datagrid.in2p3.fr/distribution/wp2/RPMS>

There is no special configuration information to be detailed here.

5.3.2.1.2. Install the RPMs

Install the rpms related to the *RB and JSS* (note: names can vary depending of the standardization of names that will be set in “Santiago de Compostela” Meeting)

```
rpm -ivh workload-profile.X.Y.Z-K.i386.rpm
```

```
rpm -ivh proxy-X.Y.Z-K.i386.rpm
```

```
rpm -ivh jobsubmission-X.Y.Z-K.i386.rpm
```

```
rpm -ivh jobsubmission-profile-X.Y.Z-K.i386.rpm
```

5.3.2.1.3. Configuration settings

Once the rpm has been installed, the RB and JSS services must be properly configured. This can be done editing the two files *rb.conf* and *jss.conf* that are stored in *<install-path >/etc*. Actions to be performed to configure the Resource Broker and the Job Submission Service are described in the section 4.2.4 of [1].

Reading of this section is strongly recommended.

5.3.2.1.4. Environment variables

RB

Environment variables that have to be set for the RB are listed below:

- PGSQL_INSTALL_PATH the Postgres database installation path. Default value is */usr/local/pgsql*
- PGDATA the path where are stored the Postgres database data

Files. Default value is */usr/local/pgsql/data*
 -GDMP_INSTALL_PATH the gdmp installation path. Default value is */opt/edg*.

Setting of PGSQL_INSTALL_PATH and PGDATA is only needed if installation is not performed from rpm. Moreover \$GDMP_INSTALL_PATH/lib has to be added to LD_LIBRARY_PATH. Finally, there are other environment variables needed at run-time by RB. They are:

-EDG_WL_RB_CONFIG_DIR the RB configuration directory
 -X509_HOST_CERT the user certificate file path
 -X509_HOST_KEY the user private key file path
 -X509_USER_PROXY the user proxy certificate file path
 - GRIDMAP location of the Globus *grid-mapfile* that translates X509 certificate subjects into local Unix usernames. The default is */etc/grid-security/grid-mapfile*.

Anyway, all variables in the last group are set by the *broker* start-up script.

JSS

Environment variables that have to be set for the JSS are listed below:

-PGSQL_INSTALL_PATH the Postgres database installation path. Default value is */usr/local/pgsql*
 -PGDATA the path where are stored the Postgres database data Files. Default value is */usr/local/pgsql/data*
 -PGUSER the user that has been used to start postgres services. Default value is *postgres*
 -CONDOR_CONFIG The CondorG configuration file path. Default value is */home/dguser/CondorG/etc/condor_config*
 -CONDORG_INSTALL_PATH the CondorG installation path. Default value is */home/dguser/CondorG*

Setting of these variables is only needed if installation is not performed from rpms. However don't forget to check them in the file */opt/edg/etc/wl-jss_rb-env.sh* when you install rpms. Moreover:

-\$CONDORG_INSTALL_PATH/bin
 -\$CONDORG_INSTALL_PATH/sbin
 -\$PGSQL_INSTALL_PATH/bin (only if installation is not performed from rpm)
 must be included in the PATH environment variable and

-\$CONDORG_INSTALL_PATH/lib,
 -\$PGSQL_INSTALL_PATH/lib (only if installation is not performed from rpm)
 have to be added to LD_LIBRARY_PATH. Finally, there are other environment variables needed at run-time by JSS. They are:

-EDG_WL_JSS_CONFIG_DIR the JSS configuration directory
 -X509_HOST_CERT the user certificate file path

- X509_HOST_KEY the user private key file path
- X509_USER_PROXY the user proxy certificate file path
- GRIDMAP location of the Globus *grid-mapfile* that translates X509 certificate subjects into local Unix usernames. The default is */etc/grid-security/grid-mapfile*.

Anyway all variables in the last group are set into the *jobsubmission* start-up script.

5.3.3.INFORMATION INDEX SERVICES

The Information Index (II) is the service queried by the Resource Broker to get information about resources for the submitted jobs during the matchmaking process. An II must hence be deployed for each RB/JSS instance.

This section describes the steps to be performed to install and configure the Information Index service.

5.3.3.1.INFORMATION INDEX

5.3.3.1.1.Install Required Software

Install the required software described in section . No special configuration settings is required.

5.3.3.1.2.Install the RPMs

Install the rpms related to the *information index* (note: names can vary depending of the standardization of names that will be set in “Santiago de Compostela” Meeting)

In order to install the Information Index service, the following command has to be issued with root privileges:

```
rpm -ivh workload-profile.X.Y.Z-K.i386.rpm
rpm -ivh informationindex.X.Y.Z-K.i386.rpm
rpm -ivh informationindex-profile.X.Y.Z-K.i386.rpm
```

5.3.3.1.3.Configuration settings

The II has two configuration files that are located in *<install-path>/etc* and are named:

- grid-info-slapd-giis.conf
- grid-info-site-giis.conf

In *grid-info-slapd-giis.conf* are specified the schema file locations and the database type, whilst in *grid-info-site-giis.conf* are listed the entries for the GRISes that are registered to this II. Each entry has the following format:

```
dn: service=register, dc=mi, dc=infn, dc=it, o=grid
objectclass: GlobusTop
objectclass: GlobusDaemon
```

```
objectclass: GlobusService
objectclass: GlobusServiceMDSResource
Mds-Service-type: ldap
Mds-Service-hn: bbq.mi.infn.it
Mds-Service-port: 2135
Mds-Service-Ldap-sizelimit: 20
Mds-Service-Ldap-ttl: 200
Mds-Service-Ldap-cachettl: 50
Mds-Service-Ldap-timeout: 30
Mds-Service-Ldap-suffix: o=grid
```

The field *Mds-Service-hn* specifies the GRIS address; the *Mds-Service-port* specifies the GRIS port (2135 is strongly recommended) whilst the other entries are related to *ldap sizelimit* and *ldap ttl*. To add a new GRIS to the given II, it suffices to add a new entry like the one just showed, to the *grid-info-site-giis.conf* file.

Another file that can be used to configure the II is the start -up script *information_index*. In this file is indeed specified the number of the port that is used by the II to listen for requests whose default is 2170. This value can be changed to make II listen on another port provided it matches with the value of the *MDS_port* attribute in the RB configuration file *rb.conf* (see section 4.2.4.1 of [1]).

5.3.3.1.4.Environment variables

The only environment variable needed for the II to run is the Globus installation path *GLOBUS_LOCATION* that is anyway set by the start -up script *information_index*.

5.3.4.USER INTERFACE SERVICES

This section describes the steps needed to install and configure the User Interface, which is the software module of the WMS allowing the user to access main services made available by the components of the scheduling sub -layer.

5.3.4.1.USER INTERFACE

5.3.4.1.1.Install Required Software

Install the required software described in section No special configuration settings are required.

5.3.4.1.2.Install the RPMs

Install the rpms related to the *user interface* (note: names can vary depending of the standardization of names that will be set in the Santiago de Compostela Meeting)

In order to install the Information Index service, the following command has to be issued with root privileges:

```
rpm -ivh workload-profile.X.Y.Z-K.i386.rpm
rpm -ivh userinterface-profile.X.Y.Z-K.i386.rpm
rpm -ivh userinterface-X.Y.Z-K.i386.rpm
```

5.3.4.1.3.Configuration settings

Configuration of the User Interface is accomplished editing the file “<install-path>/etc/UI_ConfigENV.cfg:” to set opportunely the contained parameters. They are explained in detail in section 4.4.4 of [1]

5.3.4.1.4.Environment variables

Environment variables that have to be set for the User Interface are listed hereafter:

-X509_USER_KEY	the user private key file path. Default value is <i>\$HOME/.globus/userkey.pem</i>
-X509_USER_CERT	the user certificate file path. Default value is <i>\$HOME/.globus/usercert.pem</i>
-X509_CERT_DIR	the trusted certificate directory and ca-signing-policy directory. Default value is <i>/etc/grid-security/certificates</i>
-X509_USER_PROXY	the user proxy certificate file path. Default value is <i>/tmp/x509up_u<UID></i> where UID is the user identifier on the machine as required by GSI.

Moreover there are:

-EDG_WL_UI_CONFIG_PATH	Non standard location of the UI configuration file <i>UI_ConfigENV.cfg</i> . This variable points to the file absolute path.
-EDG_WL_LOCATION	UI install path. It has to be set only if installation has been made in a non default location. It defaults to <i>/opt/edg</i>
-GLOBUS_LOCATION	The Globus rpms installation path.

The last two variables are anyway set automatically once the *userinterface-profile* rpm is installed.

The Logging library, i.e. the library that is linked into UI for logging the jobs transfer events, reads its immediate logging destination from the variable DGLOG_DEST. Correct format for this variable is:

DGLOG_DEST=x-dglog://HOST:PORT

where HOST defaults to *localhost* and PORT defaults to *15830*. On the submitting machine if the variable is not set it is dynamically assigned by the UI with the value:

```
DGLOG_DEST=x-dglog://<LB_CONTACT>:15830
```

where LB_CONTACT is the hostname of the machine where the LB server currently associated to the RB used for submitting jobs is running.

6. TESTING YOUR INSTALLATION

6.1. OPERATING THE SYSTEM

For security purposes all the WMS daemons run with proxy certificates. These certificates are generated from the start-up scripts that are described in the following section, before the applications are started. Lifetime of proxies created by the start-up scripts is 24 hours. In order to provide the daemons with valid proxies for all their lifetime the administrators need to ensure regular generation of new proxies. This can be achieved adding the following lines to the machine */etc/crontab*:

```
57 2,8,14,20 * * * root service locallogger proxy
57 2,8,14,20 * * * root service lbserver proxy
57 2,8,14,20 * * * root service broker proxy
57 2,8,14,20 * * * root service jobsubmission proxy
```

This will make proxies be created by *cron*.

6.2. LB LOCAL-LOGGER

6.2.1. Starting and stopping daemons

To run the LB local-logger services, it suffices to issue as root the following command:

```
/etc/rc.d/init.d/locallogger start
```

if the *locallogger-profile* rpm has been installed. Otherwise you can use

```
<install path>/sbin/locallogger start
```

This makes both the *dglogd* and the *interlogger* processes start.

The same can be done issuing the following command s:

```
<install path>/sbin/dglogd <options>
<install path>/sbin/interlogger <options>
```

Both daemons recognize a common set of options:

```
--key=<keyfile>      host certificate private key file (this option overrides value of the
                    environment variable X509_USER_KEY). An example of option
                    usage:
                    --key=/etc/grid-security/hostkey.pem

--cert=<certfile>    host certificate file (this option overrides value of the environment
                    variable X509_USER_CERT). An example of option usage:
```

```
--cert=/etc/grid-security/hostcert.pem
```

--CAdir=<certdir> trusted certificate and ca-signing-policy directory (this option overrides value of the environment variable X509_CERT_DIR). An example of option usage:
--CAdir=/etc/grid-security/certificates

--file-prefix=<file path> Absolute path of the file where are stored locally the logged events. The default value is */tmp/dglog*, which can result in risk of data loss in case of reboot. Note that the same value must be specified for *dglogd* and *interlogger*.

--debug make the process run in foreground to produce diagnostics

Using the options explicitly is recommended rather than relying on the correspondent environment variables.

Stopping the *LB local-logger* services can be performed using the *locallogger* script with the *stop* option.

6.2.2.Troubleshooting

If the LB local-logger services are started in debug mode (i.e. using the *--debug* option), the daemons log fatal failures with *syslog()*.

6.3.LB SERVER

6.3.1.Starting and stopping daemons

To run the *LB server* services, it suffices to issue as root the following command:

```
/etc/rc.d/init.d/lbserver start
```

if the *lbserver-profile* rpm has been installed. Otherwise you can use

```
<install path>/sbin/lbserver start
```

This makes both the *bkserver* and the *ileventd* processes start.

The same can be done issuing the following commands:

```
<install path>/sbin/ileventd <options>
```

```
<install path>/sbin/bkserver <options>
```

Both daemons recognize a common set of options:

<code>--key=<keyfile></code>	host certificate private key file (this option overrides value of the environment variable <code>X509_USER_KEY</code>). An example of option usage: <code>--key=/etc/grid-security/hostkey.pem</code>
<code>--cert=<certfile></code>	host certificate file (this option overrides value of the environment variable <code>X509_USER_CERT</code>). An example of option usage: <code>--cert=/etc/grid-security/hostcert.pem</code>
<code>--CAdir=<certdir></code>	trusted certificate and ca-signing-policy directory (this option overrides value of the environment variable <code>X509_CERT_DIR</code>). An example of option usage: <code>--CAdir=/etc/grid-security/certificates</code>
<code>--debug</code>	make the process run in foreground to produce diagnostics

Using the options explicitly is recommended rather than relying on the correspondent environment variables.

Stopping the *LB server* services can be performed using the *lbserver* script with the *stop* option.

6.3.2.Purging the LB database

The *bkpurge* process, whose executable is installed in `<install path>/sbin`, is not a daemon but an utility which should be run periodically (e.g. using a cron job) in order to remove inactive jobs (i.e. those that have already entered the *Cleared* status since a certain amount of time) from the LB database. This utility recognizes the following set of options:

```
--log                data being purged from database are dumped on the stdout
--outfile=<file>    data being purged from database are dumped in the file named
                    <file>
--mysql=<database>  name of the database to be purged. It must be the same used
                    by bkserver (this option is not required in the standard set-up
--timeout=<timeout>[smhd] removes data for all jobs that entered the “Cleared” status
                    since more than <timeout> [seconds/minutes/hours/days].
--debug             print diagnostics on the stderr
--nopurge           dry run mode. It doesn't really purge (useful for debugging
                    purposes)
--aborted, -a      delete from the database data also for jobs that have entered
                    the “Aborted” status
```

If `--log` is specified, the data in ULM format are dumped to *stdout* (or `<file>`). Normally information is appended to the file. The file is locked with *flock* (`_LOCK_EX`) to prevent race conditions, e.g. rotating logs.

An example of usage of this utility could be the issuing once a day, using a cron job, of a *bkpurge* like:

```
bkpurge --log --outfile=/var/log/dglb-data.log --timeout=14d
```

6.3.3.Troubleshooting

If the LB server services are started in debug mode (that is using the `--debug` option) the daemons log fatal failures with `syslog()`.

6.4.RB AND JSS

6.4.1.Starting PostGreSQL

Both RB and JSS use the service offered by the database. It must be started **before** one of these daemons using its own startup script:

```
/etc/rc.d/init.d/postgresql start
```

or using RedHat service command:

```
service postgresql start
```

Stopping is achieved by the same commands with the stop parameter:

```
/etc/rc.d/init.d/postgresql stop
```

or

```
service postgresql stop
```

6.4.2.Starting and stopping JSS and RB daemons

The packages **-profile.X.Y.Z.rpm* provide the SysV RedHat-like scripts that allow starting these daemons. In particular startup of RB or JSS can be achieved issuing directly:

```
/etc/rc.d/init.d/broker start  
/etc/rc.d/init.d/jobsubmission start
```

or, indirectly, using RedHat dedicated commands:

```
service broker start  
service jobsubmission start
```

In the same way stopping is achieved by:

```
/etc/rc.d/init.d/broker stop  
/etc/rc.d/init.d/jobsubmission stop
```

or

```
service broker stop  
service jobsubmission stop
```

The startup script for JSS also starts and stops the underlying CondorG service. If any of the configuration steps described in section RB and JSS has been followed, these scripts will start the daemons with the correct selected users. However do not forget to put the right files (*hostkey.pem* and *hostcert.key*) in the locations pointed respectively by the variables *X509_HOST_KEY* and *X509_HOST_CERT* (this must be located in the subdirectory *hostcert* of the home directory of the *dguser* account).

Startup scripts can also be used to know the current status of the daemons using the *status* option:

```
service broker status
service jobsubmission status
```

Moreover it is strongly recommended to set the configuration of the machine in such a way that all these services (PostgreSQL, RB and JSS) will be started at the startup of the system. For these issue, refer to the RedHat *chkconfig* SysV script manager command.

6.4.3.RB and JSS databases clean-up

Hereafter are reported the instructions for cleaning -up the PostgreSQL databases used by the RB and the JSS to store persistent information about handled jobs. They can be useful when a re -start in a clean context is needed or in case the content of the databases has been corrupted following a serious failure of some component.

Resource Broker

```
pgsql -U postgres <RB_db_name>
delete from job;
"\q" (to quit)
```

RB_db_name is the name of the database used by the Resource Broker (usually set to *rb*)

Job Submission Service

```
pgsql -U postgres template1
delete from condor_submit;
"\q" (to quit)
```

template1 is the default name of the database used by the JSS. It is configurable through the *Database_name* parameter of the *jss.conf* file.

6.4.4.RB troubleshooting

The RB supplies with a log file recording its various events. This file can be used to debug abnormal behaviors of the service. . The RB log -file name and other properties can be changed by directly modifying the *rb.conf* configuration file. You can change the name of the file, the debug level and the maximum file size in bytes, as well.

6.4.5.JSS troubleshooting

The script responsible to start JSS also includes the definition of the JSS log files. There are two of them and their pathname is set respectively to: `/var/tmp/JSSserver.log` and `/var/tmp/JSSparser.log`. As before, modifying these locations implies a modification of the `/etc/rc.d/init.d/jobsubmission` script in the following two lines:

```
SERVERLOG=/var/tmp/JSSserver.log  
PARSERLOG=/var/tmp/JSSparser.log
```

6.5.INFORMATION INDEX

6.5.1.Starting and stopping daemons

To start/stop the II, the following command has to be used as root:

```
/etc/rc.d/init.d/information_index {start | stop}
```

6.6.TESTING THE NEW FEATURES

6.7.MULTIPLE CE SELECTION

As it was said before, one of the new features is the selection of multiple CEs for submitting a mpi job to the selected resources.

1.- What we need first is a JDL file defining the mpi job that we want to submit to the resource broker. This jdl file shall contain the specifications and requirements of the job, and also the new fields that we have established to correctly define the mpi jobs.

This new fields are:

JobType	Field that defines that is a mpi job. Possible values are: mpi - defines an mpi job sequential - (default) common sequential job
NumCPU	Field that defined the required number of cpus to execute the mpi job

Examples of jdl files that define mpi jobs can be found in the cvs under the path:

http://gridportal.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp3/wp3_2-scheduling/etc/tests/

For testing purposes we can choose one of the jdl files, for example <file.jdl>:

```

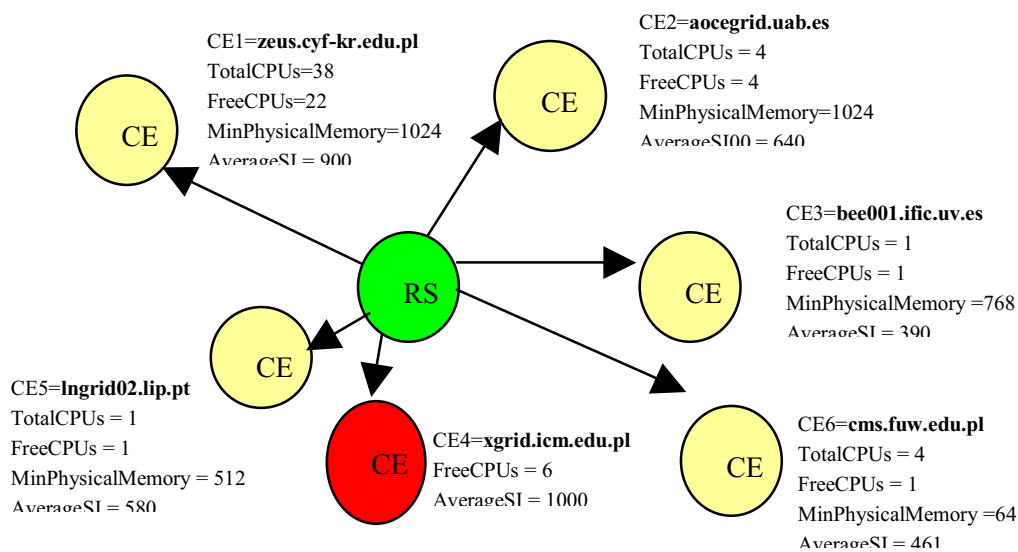
Executable      = "mpi_app";
JobType         = mpi;
NumCPU         = 3;
Arguments       = "-n";
StdOutput       = "std.out";
StdError        = "std.err";
Requirements    = other.LRMSType=="PBS";
Rank           = other.AverageSI00;
OutputSandbox  = {"std.out","std.err"};
    
```

We can see that in this description we are looking for groups of CEs whose queue type is PBS and that have at least 3 free CPUs counting all the involved CEs in the group, to run our mpi job.

2.-Next we will submit the jdl file to the modified RB that supports this new syntax in the jdl file. The command to get the available CEs would be, as usual:

dg-job-list-match file.jdl

With this command we are sending the jdl file to the RB to obtain the list of available CE . We can suppose the next scenario:



In this scenario, the CE marked in red (xgrid.icm.edu.pl) is supposed to not being MPI enabled, so it will be not considered while forming groups of Ces where to execute mpi jobs.

The other Ces are mpi enabled, so it will be considered.

For the proposed exampled we would obtain the next output in the screen:

```
bee004:~> dg-job-list-match -c UI_ConfigENV.cfg test5.jdl
Connecting to host bee004.ific.uv.es, port 7771

Groups of CEs that match job requeriments :
[Groups with 1 CEs]
[Rank=640] aocegrid.uab.es:2119/jobmanager-pbs-workq
        TotalCPUs=4   FreeCPUs=4

[Rank=900] zeus.cyf-kr.edu.pl:2119/jobmanager-pbs-workq
        TotalCPUs=38  FreeCPUs=22

[Groups with 3 CEs]
[Rank=477]
bee001.ific.uv.es:2119/jobmanager-pbs-qgrid TotalCPUs=1   FreeCPUs=1
cms.fuw.edu.pl:2119/jobmanager-pbs-workq   TotalCPUs=4   FreeCPUs=1
lngrid02.lip.pt:2119/jobmanager-pbs-qgrid  TotalCPUs=1   FreeCPUs=1

Note: currently only submission to single CEs is supported

*****
                COMPUTING ELEMENT IDs LIST
The following CE(s) matching your job requirements have been found:

- aocegrid.uab.es:2119/jobmanager-pbs-workq
- zeus.cyf-kr.edu.pl:2119/jobmanager-pbs-workq

*****
```

We can see in this output that we have two single Ces that fulfil all the requirements that are
 aocegrid.uab.es
 zeus.cyf-kr.edu.pl

And that we can form one group with 3 Ces that fulfils all the requirements (has 3 free CPUs adding the single CPUs form any one of them).

For a more detailed explanation of how the process of resource selection is carried, document [2] can be consulted.

7. REFERENCES

[1] DATAGRID WP1 – WMS Software administrator and User Guide (DataGrid -01-TEN-0118-0_8.doc)download at <http://server11.infn.it/workload-grid/documents.html>

[2] The EU-Crossgrid Approach for Grid Application Scheduling (Elisa Heymann, Álvaro Fernandez, Miquel A. Senar, Jose Salt). To be published at post -proceedings in the Springer Lecture Notes in Computer Science. Presented at Across Grids Conference and available at <http://alpha.ific.uv.es/~alferca/SchedulingEuCrossgridApproach.doc>