DataGrid

Installation Guide

Integration Team (WP6)

Document identifier:	**DataGrid-06-TED-0105-1-25**
Date:	**October 14, 2003**
Workpackage:	**Integration Team (WP6)**
Partners:	**Contributions from all partners**

Abstract: This document is intended for Site Administrators of DataGrid testbed resources and may also be useful for people in the experiments (DataGrid WP8-10) who need detailed information about the EDG middleware v1.4.x. The aim of this document is to give an overview of the European Datagrid Middleware released for Testbed1 as well as to lead site administrators through the whole installation, configuration and customization process. See chapter 1 for step-by-step installation instructions. This document contains descriptions both for sites using LCFG and for those relying on manual procedures.

Change Log

Version	Date	Comment
1	31 Dec 2003	First public version
2	12 Jan 2003	Updated version.

Contents

List of Tables

1 Overview

The European DataGrid (EDG) operates an application testbed which allows users from various application areas to test the EDG software in a semi-production environment. Volunteer sites provides the hardware on which run the necessary EDG services. This manual describes the installation of the EDG software to allow your site to join the application testbed.

In parallel, EDG operates a development testbed to test new releases of the EDG software before deployment on the application testbed. The descriptions in this manual are likely to lag behind the software in the development release, but is nonetheless useful.

1.1 EDG Services

Certificate Authority

Virtual Organization Membership Service

Computing Element

Storage Element

Replica Location Service

Resource Broker

Information System

1. Please read the EDG Usage Guidelines which can be found on the WP6 documentation web page[1].
2. Ensure that prerequisites listed in section 1.2 are satisfied.
3. Based on the size and complexity of your planned testbed site decide whether you want to use LCFG, provided by WP4, or do with manual installation. For small sites, or sites constrained by their current sysadmin practice manual installation is recommended. If you decide to use LCFG setup an an LCFG server [See chapter 2 for instructions]. In the very near future configuration files for a new version of LCFG (LCFGng) will be available. You should consider using this. Differences are documented by WP4 and a reference is given in chapter 2.
4. You should request host certificates as specified in chapter 4. Additionally, it is recommended that you request a user certificate for yourself to test the installation.
5. Download and install needed packages on your host following the instructions given in the next sections of this document. In case you are using LCFG also download the configuration files of the wanted version from the edg CVS web page[2].
6. Follow the procedures outlined in Section 4 for each of the machine type you intend to install.
7. Test your installation as described in Section 5.
8. Contact the edg integration team to announce your site being ready for being included in the edg testbed.

[1] http://marianne.in2p3.fr/datagrid/documentation

[2] http://marianne.in2p3.fr/datagrid/repository

1.2 Prerequisites

The following is a list of prerequisites for installation of the European DataGrid software.

1. The current reference platform for EDG Middleware is RedHat Linux 6.2 with all of the recommended package and kernel upgrades. This does not imply that the software will not work on other versions, other Linux flavors (like Debian, S.u.S.E., etc.), or other architectures (e.g. Solaris). It just means that the RedHat Linux 6.2 is the only tested platform and that others will not be supported for Testbed 1. The move to RedHat Linux 7.3 is scheduled for spring 2003.

2. All RPMs for Testbed 1.4.x have been created with RPMv3 to ensure maximum portability even if the patched version of RedHat 6.2 supports RPMv4.

3. All the nodes in your farm must be NTP (Network Time Protocol) clients to make sure their clocks are synchronized with all other Testbed 1 machines (desynchronization is one common reason of Globus job submission failures). See 8.1 for further details.

4. If your site consists of multiple machines, then you must be running a shared file system. In particular for job submissions to work correctly, each worker account must have a shared home area on all machines.

5. For all but the simplest sites, you will need a local resource management system (batch system). Currently PBS and LSF are supported by EDG.

6. Request host and user certificates (see 8.2.1).

1.3 EDG specific defaults

Most of the EDG software is installed under the root directory `/opt/edg`. The Globus Toolkit(TM) version 2.0 is installed under `/opt/globus`. Some nodes use parts of the 2.4 release of globus in parallel. These components can be found under `/opt/globus-24`. Be aware that some of the packages might not be *not* relocatable.

1.4 Obtaining EDG Software

The package repository, hosted by CC-IN2P3, provides access to the packaged Globus, DataGrid, and required external software. All software is packaged as source and binary RPM files. A central CVS repository, intended mainly for developers, maintains the sources of the DataGrid code.

In addition the CVS server maintains the configuration files and examples needed if your site is based on LCFG these can be found under edg-release[3]. Contact the Integration team to get information about the currently deployed version.

All of the packages are available directly through a web interface[4]. However, given the sheer number of packages in the EDG Testbed release, this is convenient only for downloading individual packages.

The *most convenient* access is via a set of configuration files. These files contain links to various subsets of the recommended RPMs. The `wget` command can be used to retrieve all of the referenced RPMs. See the EDG packages link[5] on the WP6 website[6] for further details.

[3] http://datagrid.in2p3.fr/cvsweb/edg-release/
[4] http://datagrid.in2p3.fr/pkgs/raw
[5] http://marianne.in2p3.fr/autobuild/rpmlist/
[6] http://marianne.in2p3.fr/datagrid/

If `wget` is not installed on your system you can get it from rpmfind.net[7]. More recent version of `wget`, in particular the one distributed with RH 7.2, does not work correctly. The version referenced here is recommended.

[7] http://rpmfind.net/linux/RPM/redhat/6.2/i386/wget-1.5.3-6.i386.html

2 WP4 Middleware Installation (LCFGng setup)

2.1 Introduction

Using LCFGng (LCFG new generation) it is possible to install and configure automatically a set of machines (a farm) from scratch. More important, the system allows to manage the configuration that you have and to recreate the exact settings.

This tool is useful not only if you have a large number of nodes but also if you want to automate your testbed installation and configuration. It is also possible to perform automatic upgrading/downgrading operations.

LCFGng configuration templates and tools (LCFGng components) have been developed in order to install and configure all the types of testbed machines: Computing Element (CE), Worker Node (WN), Storage Element (SE) and User Interface(UI) (for the Resource Broker and the Network Monitoring machine RPM lists exist but a manual configuration will be needed until LCFGng is introduced).

LCFGng needs a central server (LCFGng server) from which LCFGng clients fetch configuration files and RPMs (via HTTP and a NFS export).

On the LCFGng server both HTTP and DHCP servers are needed, and a disk partition is shared with the clients via NFS. The clients, during the boot process, take the IP number via DHCP and the configuration via HTTP.

The information about the server and clients installation is too long to fit in this document. The following sections are a short summary of the main points of the LCFGng server and client installation. Details can be found in the referenced documentation.

Soon the current 1.4x version of EDG will be ported to LCFGng. Information about the changes can be found on the WP4 website.

2.2 LCFGng server installation

To install testbed machines via LCFGng the first step is the creation of an LCFGng server.

The WP4 web pages[1] provide all of the information necessary to install a LCFGng server.

The server contains the operative systems for the clients and all the DataGrid RPMs to be installed in the clients. It also holds the client configuration files derived from templates.

2.3 Testbed Machine Installation

The Notes about farm installation by means of LCFGng[2] contains information about LCFGng installation of testbed machines.

The "edg-release"[3] module, available via CVS, provides RPM lists and the configuration files.

In this repository you can find the RPM lists in the "ng_rpmlist" directory and the configuration files in the "ng_source" dir. A few of these files must be customized following your site-specific configuration. In particular the file "site-cfg.h" contains almost all the site-specific configurations, including the Globus

[1] http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/documentation/
[2] http://www.lnl.infn.it/datagrid/wp4-install/
[3] http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/edg-release/

configuration parameters (there is an example in CVS). The LCFGng components automatically configure Globus and DataGrid middleware reading the parameters in this file.

The meaning of the different Globus and DataGrid middleware parameters for CE,SE,WN and UI are explained in subsequent sections of this guide.

After the installation LCFGng also sets up the NFS home directories sharing between the CE and the WN.

Two items are not yet fully automated via LCFGng. The system administrator must manually:

1. install the host certificates
2. for the CE and WN configure the local resource manager (instructions for PBS are provided in this guide).

Depending on the type of node, ranging from few to many additional manual configuration steps are involved.

Node	Function	RPM List	Manual Configuration
CE	Gateway and local job control	yes	only local batch system
WN	Worker node	yes	no
UI	User interface	yes	no
NM	Network monitoring node	yes	yes
RC	Replica Catalog	yes	yes
SE	Storage Interface Element	yes	no
BDII	Information index node	yes	yes
MP	MyProxy server	yes	no
RB	Resource broker	yes	yes
MDS	Top node of the MDS hierarchy	yes	no

3 Daemons

The numerous grid services are provided by a set of daemons running on the testbed machines. These daemons are typically configured via a text file, log status information to a file, and are controlled via a SysV-type script. This section describes where to find the appropriate files and how to use the scripts.

Most information in this section is given in the form of tables. Many of the configuration files are managed by LCFG, even if using a configuration management tool it is useful to have access to this information.

3.1 Configuration and Log Files

Table 3.1: EDG Gatekeeper Daemon Information

Description:	Globus gatekeeper with callout to LCAS (Local Centre Authorization Service)
Supplier:	Globus + EDG
Responsible WP:	WP4
SysV name:	globus-gatekeeper
SysV directives:	start, stop, status, restart, reload, condrestart
Username:	root
Process name(s):	edg-gatekeeper (1 process)
Process ID (PID) file(s):	none (?)
Lock file(s):	/var/lock/subsys/globus-gatekeeper
Log file(s):	/var/log/globus-gatekeeper.log
Configuration file(s):	/opt/globus/etc/globus-gatekeeper.conf, (LCAS 1.x: /opt/edg/etc/lcas/allowed_users.db, /opt/edg/etc/lcas/ban_users.db, /opt/edg/etc/lcas/timeslots.db, (LCAS 1.1 and higher) lcas.db)
LCFG object name:	edg-lcfg-globuscfg (edg-gatekeeper), edg-lcas (LCAS)
Inbound ports:	2119/tcp
Outbound ports:	
Other daemons contacted:	none (in the future the LCAS daemon)
Link(s) to documentation:	http://www.dutchgrid.nl/DataGrid/wp4/lcas/edg-lcas-1.1 (LCAS)

Tables 3.1 to 3.21 list for different daemons information like the locations of configuration files and log files.

Table 3.22 summarizes the utilization of ports by various services.

Table 3.2: EDG Resource Broker Daemon Information

Description:	Deamon for the resource management
Supplier:	Globus + EDG
Responsible WP:	WP1
SysV name:	broker
SysV directives:	Start, Stop, Status, Restart, Proxy
Username:	dguser
Process name(s):	rbserver (many)
Process ID (PID) file(s):	/var/mon/RBserver.pid
Lock file(s):	No
Log file(s):	/var/tmp/RBserver.log
Configuration file(s):	/opt/edg/etc/rb.conf
LCFG object name:	none
Inbound ports:	none
Outbound ports:	7771
Other daemons contacted:	(many)
Link(s) to documentation:	(?)

Table 3.3: EDG Job Submission Daemon Information

Description:	Deamon to control the job submission process
Supplier:	Globus + EDG
Responsible WP:	WP1
SysV name:	jobsubmission
SysV directives:	Start, Stop, Status, Restart, Proxy
Username:	dguser
Process name(s):	jssparser (3), jssserver, condor-master, pr-daemon
Process ID (PID) file(s):	/var/run/(jssparser,jssserver,pr-daemon).pid
Lock file(s):	No
Log file(s):	In /var/tmp/ JSSserver.log, JSSparser.log CondorG.log, pr-daemon.log, jsscallback.log
Configuration file(s):	/opt/edg/etc/jss.conf
LCFG object name:	none
Inbound ports:	none
Outbound ports:	(?)
Other daemons contacted:	(?)
Link(s) to documentation:	(?)

Table 3.4: EDG Logging and Bookkeeping Server Daemon Information

Description:	Daemon to handle logging and bookkeeping
Supplier:	Globus + EDG
Responsible WP:	WP1
SysV name:	lbserver
SysV directives:	Start, Stop, Restart, Proxy, Status
Username:	root
Process name(s):	bkserver (min.3), ileventd (1)
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	Using LCFG: /var/obj/log/syslog Else: /var/log/messages
Configuration file(s):	none
LCFG object name:	none
Inbound ports:	none
Outbound ports:	7846
Other daemons contacted:	none
Link(s) to documentation:	none

Table 3.5: EDG Local Logger Daemon Information

Description:	Deamon for local logging (interlogger and dglogd)
Supplier:	Globus + EDG
Responsible WP:	WP1
SysV name:	locallogger
SysV directives:	Start, Stop, Restart, Proxy, Status
Username:	root
Process name(s):	dglogd (1), interlogger (2)
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	none
Configuration file(s):	none
LCFG object name:	none
Inbound ports:	none
Outbound ports:	15830
Other daemons contacted:	none
Link(s) to documentation:	none

Table 3.6: EDG Gridmapfile Update Daemon Information

Description:	Cron job to upgrade the /etc/grid-security/grid-mapfile used by GSI
Supplier:	Globus + EDG
Responsible WP:	WP6
SysV name:	None, started via cron service
SysV directives:	doesn't apply is started from /opt/edg/etc/cron
Username:	root
Process name(s):	mkgridmap (1 process) called by /opt/edg/etc/cron/mkgridmap-cron
Process ID (PID) file(s):	none (?)
Lock file(s):	none (?)
Log file(s):	none (?)
Configuration file(s):	EDG_LOCATION/etc/mkgridmap-cron.conf EDG_LOCATION/etc/mkgridmap.conf
LCFG object name:	none
Inbound ports:	none
Outbound ports:	389 (contacts a slapd)
Other daemons contacted:	LDAP servers
Link(s) to documentation:	none

Table 3.7: EDG CRL Update Daemon Information

Description:	Cron job to upgrade the Certificate Revocation List used by GSI
Supplier:	Globus + EDG
Responsible WP:	WP6
SysV name:	None, started via cron service
SysV directives:	doesn't apply, started from /opt/edg/etc/cron
Username:	root
Process name(s):	edg-fetch-crl (1 process) called by /opt/edg/etc/cron/edg-fetch-crl.cron
Process ID (PID) file(s):	none (?)
Lock file(s):	none (?)
Log file(s):	none (?)
Configuration file(s):	EDG_LOCATION/etc/edg-fetch-crl-cron.conf
LCFG object name:	none
Inbound ports:	none
Outbound ports:	80
Other daemons contacted:	WEB servers
Link(s) to documentation:	none

Table 3.8: MyProxy Daemon Information

Description:	Server to allow proxy renewal from RB.
Supplier:	NCSA
Responsible WP:	WP1 (use)/WP6 (pkg./config.)
SysV name:	myproxy
SysV directives:	start, stop
Username:	root
Process name(s):	myproxy-server (1)
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	/var/log/syslog /var/obj/log/syslog (LCFG)
Configuration file(s):	/opt/edg/edg-myproxy.conf
LCFG object name:	myproxy
Inbound ports:	7512
Outbound ports:	none
Other daemons contacted:	none
Link(s) to documentation:	http://www.ncsa.uiuc.edu/Divisions/ACES/MyProxy/

Known Limitations of the myproxy server 3.8: The configuration of the MyProxy server is intended only for the EDG usage. That is, proxy renewal is only allowed from Resource Broker machines. More complicated configurations are possible but need manual configuration.

Table 3.9: MDS Daemon Information

Description:	MDS Information System
Supplier:	Globus
Responsible WP:	WP3
SysV name:	globus-mds
SysV directives:	start, stop, restart, condrestart, status, reload
Username:	edginfo
Process name(s):	slapd (3–30)
Process ID (PID) file(s):	/var/tmp/edginfo-globus-mds.pid
Lock file(s):	/var/tmp/edginfo-globus-mds.lock
Log file(s):	/var/tmp/edginfo-globus-mds.log
Configuration file(s):	/etc/globus2.conf
LCFG object name:	globuscfg
Inbound ports:	2135/tcp
Outbound ports:	none
Other daemons contacted:	none
Link(s) to documentation:	http://www.globus.org/mds

Table 3.10: Replica Catalogue Daemon Information

Description:	LDAP Replica Catalogue (uses slapd)
Supplier:	WP2
Responsible WP:	WP2
SysV name:	edg-rc-server
SysV directives:	start, stop, restart, reload, condrestart
Username:	any
Process name(s):	slapd
Process ID (PID) file(s):	$EDG_LOCATION/edg-rc-server/var/rc-slapd.pid
Lock file(s):	/var/lock/subsys/edg-rc-server
Log file(s):	$EDG_LOCATION/edg-rc-server/var/edg-rc-server.log
Configuration file(s):	$EDG_LOCATION/edg-rc-server/etc/rc-slapd.conf
LCFG object name:	none
Inbound ports:	any set by configuration file
Outbound ports:	none
Other daemons contacted:	none
Link(s) to documentation:	http://cern.ch/GDMP/documentation.html (Testbed/RC links)

Table 3.11: GDMP Daemon Information

Description:	GDMP server for data set replication, mirroring and access to MSS
Supplier:	EDG, PPDG (VDT)
Responsible WP:	WP2
SysV name:	gdmp_server, started by inetd
SysV directives:	started by inetd
Username:	gdmp
Process name(s):	gdmp_server, started by inetd
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	/opt/edg/var/{VO}/gdmp_server_log.out
Configuration file(s):	/opt/edg/etc/gdmp.shared.conf (server) /opt/edg/etc/{VO}/gdmp.conf (VO conf. for SE) /opt/edg/etc/{VO}/gdmp.private.conf (RC conf.)
LCFG object name:	gdmp
Inbound ports:	2000 (control) other for parallel streams
Outbound ports:	2000
Other daemons contacted:	none
Link(s) to documentation:	http://cern.ch/GDMP/documentation.html

Table 3.12: GridFTP Daemon Information

Description:	GridFTP (file transfer protocol) Daemon
Supplier:	Globus
Responsible WP:	e.g. WP6
SysV name:	globus-gsi_wuftpd (1-10)
SysV directives:	start, stop, restart, condrestart, status
Username:	root
Process name(s):	in.ftpd
Process ID (PID) file(s):	/var/run/globus-ftpd.pid
Lock file(s):	/var/lock/subsys/globus-gsi_wuftpd
Log file(s):	/var/log/gsiwuftpd.log
Configuration file(s):	/etc/globus.conf
LCFG object name:	globusconf
Inbound ports:	2811/tcp
Outbound ports:	none
Other daemons contacted:	none
Link(s) to documentation:	http://www.globus.org/datagrid/gridftp.html

Table 3.13: UDPmon Daemon Information

Description:	UDP bandwidth measurement daemon
Supplier:	WP7
Responsible WP:	WP7
SysV name:	none (started from crontab)
SysV directives:	none
Username:	root
Process name(s):	udpmon_resp (1)
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	{prefix}/udpmon/data/udp_bw_resp_log-out
Configuration file(s):	none
LCFG object name:	none
Inbound ports:	14234/udp, 14233/udp
Outbound ports:	none
Other daemons contacted:	none
Link(s) to documentation:	none

Table 3.14: edg-netagent Daemon Information

Description:	WP7 netagent server
Supplier:	WP7
Responsible WP:	WP7
SysV name:	N/A
SysV directives:	N/A
Username:	nobody
Process name(s):	edg-netagent (3)
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	none
Configuration file(s):	{prefix}/etc/edg-netagent/current.conf
LCFG object name:	none
Inbound ports:	3002/tcp
Outbound ports:	none
Other daemons contacted:	none
Link(s) to documentation:	none

Table 3.15: edg-ftlog2rgma Daemon Information

Description:	The ftlog2rgma daemon publishes GridFTP logs into R-GMA.
Supplier:	EDG
Responsible WP:	WP7
SysV name:	edg-ftlog2rgmad
SysV directives:	start, stop, restart, condrestart, status
Username:	root
Process name(s):	edg-ftlog2rgma (1)
Process ID (PID) file(s):	/var/run/edg-ftlog2rgmad.pid
Lock file(s):	none
Log file(s):	/var/log/edg-ftlog2rgmad.log
Configuration file(s):	none
LCFG object name:	none
Inbound ports:	none
Outbound ports:	8080/tcp (RGMA)
Other daemons contacted:	none
Link(s) to documentation:	none

Table 3.16: edg-netmon-archiver Daemon Information

Description:	netmon-archiver provides RGMA Archiver for network monitoring info.
Supplier:	EDG
Responsible WP:	WP7
SysV name:	edg-netmon-archiverd
SysV directives:	start, stop, restart, condrestart, status
Username:	runs as root at the moment
Process name(s):	[java] (9)
Process ID (PID) file(s):	/var/run/edg-netmon-archiverd.pid
Lock file(s):	none
Log file(s):	/var/log/edg-netmon-archiverdlog
Configuration file(s):	/opt/edg/etc/edg-netmon-archiver.conf
LCFG object name:	none
Inbound ports:	none
Outbound ports:	none
Other daemons contacted:	edg-netmon2rgmad
Link(s) to documentation:	none

Table 3.17: edg-netmon-rgma-info Daemon Information

Description:	netmon-rgma-info publishes NetworkSE and NetworkCE tables
Supplier:	EDG
Responsible WP:	WP7
SysV name:	edg-netmon-rgma-infod
SysV directives:	start, stop, restart, condrestart, status
Username:	root
Process name(s):	edg-netmon-rgma-info (1)
Process ID (PID) file(s):	/var/run/edg-netmon-rgma-info.pid
Lock file(s):	none
Log file(s):	/var/log/edg-netmon-rgma-info.log
Configuration file(s):	/opt/edg/etc/edg-netmon-rgma-info.conf
LCFG object name:	none
Inbound ports:	none
Outbound ports:	none
Other daemons contacted:	edg-netmon2rgmad
Link(s) to documentation:	none

Table 3.18: edg-netmon2rgma Daemon Information

Description:	netmon-rgma provides daemon which handles the producers for publishing into RGMA
Supplier:	EDG
Responsible WP:	WP7
SysV name:	edg-netmon2rgmad
SysV directives:	start, stop, restart, condrestart, status
Username:	root
Process name(s):	edg-netmon2rgma (1)
Process ID (PID) file(s):	/var/run/edg-netmon2rgmad.pid
Lock file(s):	none
Log file(s):	/var/log/edg-netmon2rgmad.log
Configuration file(s):	none
LCFG object name:	none
Inbound ports:	none
Outbound ports:	8080/tcp (RGMA)
Other daemons contacted:	none
Link(s) to documentation:	none

Table 3.19: iperf Daemon Information

Description:	WP7 bandwidth measurement server
Supplier:	Univ. of Illinois, packaged by WP7
Responsible WP:	WP7
SysV name:	N/A (started from crontab)
SysV directives:	N/A
Username:	root
Process name(s):	iperf (3)
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	none
Configuration file(s):	none
LCFG object name:	netmon
Inbound ports:	5001/tcp
Outbound ports:	none
Other daemons contacted:	none
Link(s) to documentation:	http://dast.nlanr.net/Projects/Iperf/

Table 3.20: MSA (Monitoring Sensor Agent) Daemon Information

Description:	MSA Fabric Monitoring Sensor Agent
Supplier:	EDG
Responsible WP:	WP4
SysV name:	edg_fmon_agent
SysV directives:	start, stop, status, restart, reload
Username:	root
Process name(s):	edgfmonagent (1 process)
Process ID (PID) file(s):	EDG_LOCATION_VAR/fmon/edgfmonagent.pid
Lock file(s):	none (?)
Log file(s):	EDG_LOCATION_VAR/fmon/edgfmonagent.log
Configuration file(s):	/opt/edg/var/etc/edgfmonagent.conf
LCFG object name:	edglcfgfmonagent
Inbound ports:	12409/udp
Outbound ports:	
Other daemons contacted:	none
Link(s) to documentation:	http://wwwinfo.cern.ch/pdp/monitoring

Table 3.21: fmonserver (Monitoring Repository Collector) Daemon Information

Description:	fmonserver - Monitoring Repository Collector
Supplier:	EDG
Responsible WP:	WP4
SysV name:	edg_fmon_server
SysV directives:	start, stop, status, restart, reload
Username:	root
Process name(s):	edgfmonserver (1 process)
Process ID (PID) file(s):	none
Lock file(s):	none
Log file(s):	EDG_LOCATION_VAR/fmon/edgfmonserver.log
Configuration file(s):	/opt/edg/var/etc/edgfmonserver.conf
LCFG object name:	edglcfgfmonserver
Inbound ports:	12411/udp
Outbound ports:	
Other daemons contacted:	none
Link(s) to documentation:	http://wwwinfo.cern.ch/pdp/monitoring

Table 3.22: Daemon Ports and Machine Types

Service	Ports*	Open To	GRID		Virtual Organization						Site					
			IC	MDS	VOMS	MS-LDAP	PX	RB	BDII	RLS	UI	SE	CE	WN	NM	MON
httpd (apache)	80	world	–	–	√	–	–	–	–	–	–	√	–	–	√	√
MySQL	3306	machine	√	–	√	–	–	√	–	√	–	–	–	–	–	√
NFS (Network File System)	2049	site	–	–	–	–	–	–	–	–	–	√	√	√	–	–
ntpd (Network Time Protocol)	123/udp	ntp server(s)	√	√	√	√	√	√	√	√	√	√	√	√	√	√
OpenLDAP	389	world	–	–	–	√	–	–	–	–	–	–	–	–	–	–
openssh	22	world	√	√	√	√	√	√	√	√	√	√	√	√	√	√
portmap	111/udp & tcp	site	–	–	–	–	–	–	–	–	–	√	√	√	–	–
Globus & EDG Gatekeepers	2119	world	–	–	–	–	–	–	–	–	–	–	√	–	–	–
Globus Job Manager	20000–25000†	world	–	–	–	–	–	–	–	–	–	–	√	–	–	–
GridFTP	2811	world	–	–	–	–	–	–	–	–	–	√	√	–	–	–
MyProxy	7512	world	–	–	–	–	√	–	–	–	–	–	–	–	–	–
BDII (LDAP)	2170	world	–	–	–	–	–	–	√	–	–	–	–	–	–	–
CondorG	7771	world	–	–	–	–	–	√	–	–	–	–	–	–	–	–
GOUT (LDAP)	2169	world	–	–	–	–	–	–	–	–	–	–	–	–	–	√
GridFTP (EDG modified)	2811	world	–	–	–	–	–	√	–	–	–	–	–	–	–	–
Iperf	5001	world	–	–	–	–	–	–	–	–	–	–	–	–	√	–
LEMON (fab. mon. server)	12409/udp & tcp	site	–	–	–	–	–	–	–	–	–	–	–	–	–	√
LEMON (fab. mon. service)	12411	site	–	–	–	–	–	–	–	–	–	–	–	–	–	●
Logging & Bookkeeping Server	9000, 9001	world	–	–	–	–	–	√	–	–	–	–	–	–	–	–
locallogger (logd)	9002	world	–	–	–	–	–	√	–	–	–	–	√	–	–	–
MDS (LDAP)	2135	world	–	√	–	–	–	–	–	–	–	√	√	–	–	–
Mercury Monitor (local)	3570	gatekeeper	–	–	–	–	–	–	–	–	√	–	–	–	–	–
Mercury Monitor (site)	3570	world	–	–	–	–	–	–	–	–	–	–	√	–	–	–
Network Server	7772	world	–	–	–	–	–	√	–	–	–	–	–	–	–	–
pcpd	50500	world	–	–	–	–	–	–	–	–	–	–	–	–	√	–
RFIO	3147	world	–	–	–	–	–	–	–	–	–	√	–	–	–	–
R-GMA (tomcat)	8088, 8080, 8443§	world	√	–	–	–	–	–	–	–	–	–	–	–	–	√
RLS-RLI (tomcat)	8080, 8443§	world	–	–	–	–	–	–	–	–	–	–	–	–	–	–
RLS-LRC (tomcat)	8080, 9101-9120‡	world	–	–	–	–	–	–	–	√	–	–	–	–	–	–
RLS-RMC (tomcat)	8080, 9201-9220‡	world	–	–	–	–	–	–	–	√	–	–	–	–	–	–
RLS-ROS (tomcat)	8080, 9301-9320‡	world	–	–	–	–	–	–	–	–	–	–	–	–	–	√
RTCS (scheduler input)	6091	machine	–	–	–	–	–	–	–	–	–	–	●	–	–	–
RTCS (maintenance)	6092	machine	–	–	–	–	–	–	–	–	–	–	●	–	–	–
RTCS Scheduler (Maui)	42559	machine	–	–	–	–	–	–	–	–	–	–	●	–	–	–
SE/apache-ssl	6375	world	–	–	–	–	–	–	–	–	–	√	–	–	–	–
SE (tomcat)	8080, 8443§	world	–	–	–	–	–	–	–	–	–	√	–	–	–	–
UDPmon	14233/udp	world	–	–	–	–	–	–	–	–	–	–	–	–	√	–
VOMS	15000–15020	world	–	–	√	–	–	–	–	–	–	–	–	–	–	–
VOMS (httpd & mod_ssl)	443	world	–	–	√	–	–	–	–	–	–	–	–	–	–	–
VOMS (admin. client, tomcat)	8080, 8443§	world	–	–	√	–	–	–	–	–	–	–	–	–	–	–

*All are TCP ports unless otherwise indicated.
†Selectable range; required with GridFTP, Gatekeeper (no user logins).
§Secure port, ‡One secure port per supported VO.
● optional
√ required
– not required

3.2 Daemon Control

The majority of the daemons used in the testbed implement the SysV interface. The daemons which do, provide an appropriately-named script in the directory /etc/rc.d/init.d/. At a minimum, these scripts allow you to start and stop the service with the command:

```
/sbin/service daemon_name {start|stop}
```

or

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

where daemon_name is the name of the script in the above directory and only ”stop” or ”start” is given as an argument. The scripts may also support a ”status” argument which returns information about the daemon.

Additionally, some of the scripts also support the chkconfig interface which allows you to configure the daemon to start automatically at boot time with the following command:

```
/sbin/chkconfig  daemon_name on
```

where again daemon_name is the name of the script. This command can also be used to remove the daemon from the list of those started automatically by changing ”-add” to ”-del”.

4 Configuration by Machine Type

4.1 Introduction

The various computers involved in building a site of the DataGrid testbed fall into just a few functional categories. For a typical site this involves setting up a Gatekeeper node which acts as the portal for the site and as the front-end of the local batch system controlling a set of Worker nodes. Combined, this comprise a Computing Element (CE). Most sites operating a CE will provide some persistence storage with a Storage Element which is the interface to storage devices. To access the testbed for job submission a User Interface (UI) machine is required.

Some larger sites will offer additional services like the Job Submission Service which requires to install a Resource Broker (RB). If a RB is setup the site has to run a BDII information index server, too. This is a LDAP server that acts as an information index.

To allow users to run jobs with a livetime that exceeds the livetime of the proxy certificate a MyProxy machine that provides the proxy renewal service has to be setup.

Again, this is not necessary for the typical site.

Inside the testbed for each VO a replica catalog (RC) and a VO server has to be setup and maintained.

4.1 gives the list of different machine types and the services which must run on each.

Table 4.1: Machine Types and Necessary Services

Daemon	UI	IS	CE	WN	SE	RC	RB	MProxy	BDII
Globus or Edg Gatekeeper	–	–	XX	–	XX	–	–	–	–
Globus Rep. Cat.	–	–	–	–	–	XX	–	–	–
GSI-enabled FTPD	–	–	XX	–	XX	–	XX	–	–
Globus MDS	–	XX	XX	–	XX	–	–	–	–
Info-MDS	–	XX	XX	–	XX	–	–	–	–
Broker	–	–	–	–	–	–	XX	–	–
Job submission serv.	–	–	–	–	–	–	XX	–	–
Info. Index	–	–	–	–	–	–	–	–	XX
Logging & Bookkeeping	–	–	–	–	–	–	XX	–	–
Local Logger	–	–	XX	–	XX	–	XX	–	–
CRL Update	–	–	XX	–	XX	–	XX	–	–
Grid mapfile Update	–	–	XX	–	XX	–	XX	–	–
RFIO	–	–	–	–	XX	–	–	–	–
GDMP	–	–	–	–	XX	–	–	–	–
MyProxy	–	–	–	–	–	–	–	XX	–

Before a site is setup a few things common to all nodes should be mentioned. Please read, before you start the section about time synchronization 8.1. If you install your site using LCFG this service will be handled by the tool. In case you opt for manual configuration you have to install, configure and start the service on every node. In case you are using AFS make shure that your local AFS time server is in sync with the rest of the grid.

To get a better understanding of the security model used by EDG and globus read the section 8.2.

In case you need more information about a given service have a look at the sections in the Appendix.

4.2 Using LCFG

Before the steps needed to configure individual nodes are described, a walkthrough the main configuration file (site-cfg.h) is given. For details about using LCFG follow the references given in the introduction (1).

In several places of this guide verbatim text from configuration files is used. In some cases the original lines have been to long to be given here. In these case the lines have been split. Whenever this was done a \ character at the end of the line was used. In case you use the sample code given here, please join these lines again.

The example *site-cfg.h* file has been taken from CERN. To keep the information compact we assume that only two VOs are supported.

```
 /*
  site-cfg.h
  ==================================================
  SITE SPECIFIC CONFIGURATION
*/

/* COMMON LCFG DEFINITIONS ------------------------------------------------ */

#define LCFGSRV                 lxshare0371.cern.ch
#define URL_SERVER_CONFIG       http://lxshare0371.cern.ch
```

You have to set this to the full name of your LCFG server.

```
/*SOURCE TREE LOCATIONS ---------------------------------------------------- */

/* Define the root locations of the Globus tree and the EDG tree.   These
   are used in many configuration files and for setting the ld.so.conf
   libraries. NOTE: the underscore at the end of the define.  Used to avoid
   confusion with the GLOBUS_LOCATION and EDG_LOCATION tags in configuration
   files. */
#define GLOBUS_LOCATION_        /opt/globus
#define EDG_LOCATION_           /opt/edg

/* COMMON GRID DEFINITIONS ------------------------------------------------ */

/* CE AND SE HOST NAMES. These are defined here because they are used in
   some of the site definitions. */

/* ComputingElement hostname */
#define CE_HOSTNAME             lxshare0227.cern.ch
```

```
/* StorageElement hostname */
#define SE_HOSTNAME             lxshare0393.cern.ch
```

To handle multiple CEs and SEs is possible and requires some modifications in the configuration files. This is beyond the scope of this guide.

```
/* COMMON SITE DEFINITIONS ------------------------------------------------ */

#define LOCALDOMAIN             cern.ch
#define SITE_MAILROOT           SITE_MANAGERS_MAIL_ADDRESS@YOURSITE.ch
#define SITE_GATEWAYS           137.138.1.1
/* Allowed networks (useful for tcpwrappers) */
#define SITE_ALLOWED_NETWORKS   127.0.0.1, 137.138., 128.141.
#define SITE_NAMESERVERS        137.138.16.5 137.138.17.5
```

Please note that some lists are comma separated while others, like the SITE_NAMESERVERS, are separated by a single white space !

```
/* The netmask */
#define SITE_NETMASK            255.255.0.0
/* NTP server (domain and hostname) */
#define SITE_NTP_DOMAIN         cern.ch
#define SITE_NTP_HOSTNAME       ip-time-1
/* The time zone */
#define SITE_TIMEZONE           Europe/Paris
/* Site name */
#define SITE_NAME_              CERN-PRO-1-4
```

This name must be unique inside the whole grid! Make sure you coordinate your choice with the other site administrators.

```
/* Site EDG version */
#define SITE_EDG_VERSION        v1_4_3
/* Site installation date       year month day time */
#define SITE_INSTALLATION_DATE_ 20021118120000Z
/* Site distinguished name. */
#define SITE_DN_                \"dc=cern, dc=ch, o=Grid\"
```

You can find this information in the host certificate of your CE node.

```
/* All the WN (used by /etc/export configuration of /home NFS Mount
   e.g. testbed*.lnl.infn.it. Needed by ComputingElement.h) */
#define SITE_WN_HOSTS           lxshare*.cern.ch,tbed0*.cern.ch,adc*.cern.ch
/* All the SE hosts (comma separated list) */
#define SITE_SE_HOSTS_          SE_HOSTNAME
/* List (comma separated) of the Computing Element(s) of your site */
#define SITE_CE_HOSTS_          CE_HOSTNAME:2119/jobmanager-pbs-short,\
CE_HOSTNAME:2119/jobmanager-pbs-infinite
```

This is the list of CEs and their local resource managers. 2119 is the port of the gate keeper.

```
/* The default configuration of MDS is that there is a GRIS running on
   each functional node (CE, SE).  There is a single site-level GIIS
   running by default on the CE.  This site-level GIIS then registers to
   the top-level GIIS for the production or development testbed.  The
   details are handled via the globuscfg configuration object. */
/* Usually use a name like nikhefpro or nikhefdev for the production
   or development testbeds. */
#define SITE_GIIS               cern
#define SITE_GIIS_HOSTNAME      CE_HOSTNAME
/* These point to the next highest level in the MDS hierarchy.  Ask to
   find out the parameters for this.  At time of tagging these were:
   edgdev on lxshare0372.cern.ch for DEVELOPMENT Testbed
   edgpro on lxshare0373.cern.ch for PRODUCTION (Application) Testbed
   but DO ask to be sure.*/
#define TOP_GIIS                edgpro
#define TOP_GIIS_HOSTNAME       lxshare0373.cern.ch
```

For this information you should contact the integration team. Contact information is provided on the WP6 web page.

```
/* COMMON DEFAULT VALUES -------------------------------------------------- */
/* This defines the default location for the host certificates.  If
this is different for your site define the new value here.  If you
need to change it for the CE or SE separately, see below. */
#define SITE_DEF_HOST_CERT  /etc/grid-security-local/hostcert.pem
#define SITE_DEF_HOST_KEY   /etc/grid-security-local/hostkey.pem
#define SITE_DEF_GRIDMAP    /etc/grid-security/grid-mapfile
#define SITE_DEF_GRIDMAPDIR /etc/grid-security/gridmapdir/

/* DATA MGT PARAMETERS FOR SEVERAL NODE TYPES  ---------------------------- */

/* These variables define which VOs your site supports.  At least one must
   be defined.  For each line the RC and GDMP configurations will be done and
   on the SE a GDMP server will be configured.  It also will create 50
   accounts for each defined VO.

   You must define the associated password for each of the supported VOs.
   Contact the site administrators to obtain the passwords.
*/
#define SE_VO_ALICE
#define SE_GDMP_REP_CAT_ALICE_PWD  ALICE_PASSWORD

#define SE_VO_ATLAS
#define SE_GDMP_REP_CAT_ATLAS_PWD  ATLAS_PASSWORD
```

For this information contact the integration team or the VO managers.

```
/* COMPUTING ELEMENT DEFINITIONS ------------------------------------------ */

/* Subject of the certificate */
#define CE_CERT_SBJ             \"/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0227.cern.ch\"
/* Some site and host information (it goes in globus.conf)*/
#define CE_HOST_DN              \"hn=lxshare0227.cern.ch, dc=cern, dc=ch, o=Grid\"
/* Full path of the certificate */
#define CE_CERT_PATH            SITE_DEF_HOST_CERT
```

```
/* Full path of the secret key */
#define CE_SECKEY_PATH          SITE_DEF_HOST_KEY
/* System administrator e-mail */
#define CE_SYSADMIN             SITE_MAILROOT
/* Space separated job manager list (e.g. fork, pbs, lsf), part of globus.conf.
   NOTE: To support the standard globus commands (in particular the
         globus-job-get-output command) the fork job manager must be
         listed first! I.e. the fork job manager must be the default. */
#define CE_JOBMANAGERS          \"fork pbs\"
```

Note fork has not only to be the first in the list, but it has to be always in the list!

```
/* Batch system adopted by CE (this info goes in info-mds.conf */
#define CE_BATCHSYSTEM_         pbs
/* Binaries path of the batch system */
#define CE_BATCHSYSTEM_BIN_PATH /usr/pbs/bin
/* Local queue names */
#define CE_QUEUE_               short,infinite
/* List (comma separated no spaces) of StorageElement(s) close to this CE */
#define CE_CLOSE_SE_ID_         SE_HOSTNAME
/* Mount point(s) of the SE(s) close to this CE */
#define CE_CLOSE_SE_MOUNTPOINT  /flatfiles/SE00
```

More information on mount points will be given in 4.3.1.There the layout of the shared file system is explained in more detail.

```
/* Disk description */
#define CE_DISK_DESC            15GB-EIDE
/* CPU description */
#define CE_CPU_DESC             DUAL-PIII-800
/* CE InformationProviders: MinPhysMemory */
#define CE_IP_MINPHYSMEM        512
/* CE InformationProviders: MinLocalDiskSpace */
#define CE_IP_MINLOCDISK        2048
/* CE InformationProviders: NumSMPs */
#define CE_IP_NUMSMPS           26
/* CE InformationProviders: MinSPUProcessors */
#define CE_IP_MINSPUPROC        2
/* CE InformationProviders: MaxSPUProcessors */
#define CE_IP_MAXSPUPROC        2
/* CE InformationProviders: MaxSI00.
   See some examples of SpecInt at
   http://www.specbench.org/osg/cpu2000/results/cint2000.html */
#define CE_IP_MAXSI00           380
/* CE InformationProviders: MinSI00 */
#define CE_IP_MINSI00           380
/* CE InformationProviders: AverageSI00 */
#define CE_IP_AVRSI00           380
/* CE InformationProviders: AFSAvailable: */
#define CE_IP_AFS_AFSAVAILABLE  FALSE
/* CE InformationProviders: OutboundIP */
#define CE_IP_OUTBOUNDIP        TRUE
/* CE InformationProviders: InboundIP */
```

```
#define CE_IP_INBOUNDIP         FALSE
/* CE InformationProviders: RunTimeEnvironment (1) */
#define CE_IP_RUNTIMEENV1       ATLAS-3.2.1
/* CE InformationProviders: RunTimeEnvironment (2) */
#define CE_IP_RUNTIMEENV2       ALICE-3.07.01
/* CE InformationProviders: RunTimeEnvironment (10) */
/*#define CE_IP_RUNTIMEENV10      ! define it if you need! */
/*
   This must be defined for your CE; it indicates that your site is running
   but hasn't yet been certified.  Change this to EDG-CERTIFIED once your
   site has been tested by the ITeam. */
#define CE_IP_RUNTIMEENV15        EDG-TEST
/*#define CE_IP_RUNTIMEENV15      EDG-CERTIFIED */
```

By default 15 runtime environment variables can be defined. It is possible to add more by modifying the CE specific configuration file. It is important that you first set the EDG-TEST. For details about being certified for the edg testbed contact the integration team.

```
/* The mountpoint on the CE of the SE exported area via NFS */
#define CE_MOUNTPOINT_SE_AREA   CE_CLOSE_SE_MOUNTPOINT
/* Uncomment this below if you want to collect and publish
   data from a network monitor */
/* #define NETMON_HOST_ gppnm06.gridpp.rl.ac.uk */

/* STORAGE ELEMENT DEFINITIONS -------------------------------------------- */

/* Full path of the certificate */
#define SE_CERT_PATH            SITE_DEF_HOST_CERT
/* Full path of the secret key */
#define SE_SECKEY_PATH          SITE_DEF_HOST_KEY
/* Subject of the SE certificate */
#define SE_CERT_SBJ            \"/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0393.cern.ch\"
/* Some site and host information (it goes in globus.conf) */
#define SE_HOST_DN             \"hn=lxshare0393.cern.ch, dc=cern, dc=ch, o=Grid\"
/* System administrator e-mail */
#define SE_SYSADMIN            SITE_MAILROOT
/* List (comma separated without spaces) of ComputingElement(s) close to the SE. */
#define SE_CLOSE_CE_           SITE_CE_HOSTS_
/* The value of SE_SIZE in info-mds.conf */
#define SE_DISKSIZE             15
```

The SE_DISKSIZE and SE_FILESYSTEMS_ should be set in the node configuration file reflecting the actual available space and configuration.

```
/* comma separated list without spaces, values used in df to obtain freespace */
#define SE_FILESYSTEMS_         /dev/hda2
/* Disk description */
#define SE_DISK_DESC            15GB-EIDE
/* CPU description */
#define SE_CPU_DESC             DUAL-PIII-800
/* SE protocols */
#define SE_PROTOCOLS_           gridftp,rfio,file
```

Note that the file protocol is only available if you use a shared file system between the SE and the WNs.

```
/* SE protocols ports */
/* Note that although the IANA port for rfio is 3147, the software by
   default runs on 5001. */
#define SE_PROTOCOL_PORTS_      2811,5001,
/* GDMP area */
#define SE_GDMP_AREA            /flatfiles/SE00
/* List of the supported VO. Add/remove the VO name for each VO that you
   support/do not support in both of the following defines. */
#define SE_GDMP_VOS             alice,atlas
#define SE_VO_                  alice:SE_GDMP_AREA/alice,atlas:SE_GDMP_AREA/atlas

/* WORKER NODE DEFINITIONS ------------------------------------------------ */

/* The mountpoint on the WN of the SE exported area via NFS. It should be
   the same used for the SE */
#define WN_MOUNTPOINT_SE_AREA   CE_MOUNTPOINT_SE_AREA
/* USER INTERFACE DEFINITIONS --------------------------------------------- */

/* Resource broker */
#define UI_RESBROKER            lxshare0380.cern.ch
/* Logging and Bookkeeping URL */
#define UI_LOGBOOK              https://lxshare0380.cern.ch:7846
```

If you only want to install and configure a UI, then the UI_RESBROKER and UI_LOGBOOK are the only two defines you need to change.

4.3 Shared Filesystem Layout

As mentioned earlier several parts of the file system have to be shared between nodes on a EDG site. To help site-administrators to find a workable layout the CERN system is described. This should not be seen as a general blueprint, but as a working example. We included some comments on managing users.

This section should be first read before moving on to the following sections describing the installation of individual nodes. Many things will be unclear reading this the first time, but will become clearer later.

The current setup is based on 3 RAID disk servers using EIDE disks. Each server is configured as an NFS server exporting 5 100GB partitions.

All partitions are inserted in a common file system naming schema which follows the pattern: $/shift/ < server > / < disk >$, e.g.*/shift/lxshare072d/data02*

All client nodes (UI,CE,SE,WN) mount the needed partitions using the standard name as a mount point. Ad hoc links are then created on the nodes to point to these paths (examples of this later).

As we are managing several different testbeds, we added an extra path layer which specifies the testbed which is using a particular section of a file system e.g. */shift/lxshare072d/data02/site_pro-1.3* for EDG 1.3 production site.

For a given testbed, the following disk areas are located on the disk server (in parenthesis the nodes mounting the area):

1. User home directories, */home* (UI)
2. GRID security directory, */etc/grid-security* (CE,SE,WN)
3. VO users home directories, */home* (CE,SE,WN)

4. SE storage area(s), */flatfiles* (CE,SE,WN)

1. User home directories: this area is mounted on all UI nodes, independently of the testbed they belong to. In this way users have a unique working space. This unique home directory structure is associated to a unique user account system based on a NIS server (more on this later).On all hosts the link for this area is (note the absence of the "testbed" path component):

```
/home -> /shift/lxshare072d/data01/UIhome
```

2. GRID security directory: the main reason to share this directory tree is that the grid-mapfile and the CA CRL files are regularly updated. Having each host to do it independently increases the strain on the servers providing the update information and is prone to misalignments, e.g. a user can start a job on a CE but then the job cannot access the local SE as the user's certificate is not yet accepted by that node. Sharing this directory requires that:

- only one node should install the ca_< *site* > rpms, i.e. includes the security-rpm.h file in its rpm list
- only one node should activate the CRL and grid-mapfile update jobs

For the application testbed, updates take place on the SE node lxshare0393.

On all CE,SE,WN nodes belonging to the application testbed we have the link:

```
/etc/grid-security -> /shift/lxshare072d/data02/site_pro-1.3/grid-security
```

In some occasion this directory cannot be shared. At CERN this happens on RBs and on a special SE node, lxshare0384. In both cases this is due to the fact that the grid-mapfile had to be different from the standard one: on the RB all authorized certificates must be mapped to the dguser account (gridmap directory is also not needed here), while on the special SE we wanted to limit access to a predefined set of users. On these special nodes the grid-security directory is local, all ca_< *site* > rpms are installed, and the update cron jobs are executed.

3. VO users home directories: sharing of these directories between CE and WNs is mandatory (see XX). There is no real need to share this dir with the SE, but as a gatekeeper is also running on that node, this reduces the number of directories to keep under control. The corresponding link on the application testbed is:

```
/home -> /shift/lxshare072d/data02/site_pro-1.3/CEhome
```

4. SE storage area: sharing of this area between SE and CE/WNs allows the activation of the "file" access protocol on the SE. Due to the limitation in the partition size on CERN disk servers, this area spans more than one mounted partition. On each of the nodes sharing this area we created a local /flatfiles directory tree. Within this tree we created links to the actual disk partition. The structure of the directory tree must of course be rigorously identical on all nodes sharing it.

As an example, this is the content of the */flatfiles* on the application testbed at CERN:

```
[root@lxshare0393]# ls -l /flatfiles/SE00
alice -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/alice
atlas -> /shift/lxshare072d/data03/site_pro-1.3/flatfiles/SE00/atlas
biome -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/biome
cms -> /shift/lxshare072d/data05/site_pro-1.3/flatfiles/SE00/cms
dzero -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/dzero
eo -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/eo
flatfiles -> ..
```

```
iteam -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/iteam
lhcb -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/lhcb
tutor -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/tutor
wpsix -> /shift/lxshare072d/data04/site_pro-1.3/flatfiles/SE00/wpsix
```

From this we see that all VOs are sharing the same disk partition, */shift/lxshare072d/data04*, with the exception of atlas and cms, using */shift/lxshare072d/data03* and */shift/lxshare072d/data05* respectively (note: this layout was chosen as atlas and cms were planning some production tests). Also note the $flatfiles- >$ .. link: this must always be present to allow correct LFN-to-PFN mapping on all nodes.

4.3.1 LCFG based NFS setup

To configure LCFG to mount the correct disk partitions on each node, we commented out all NFS-export/-mount sections from the default configuration files and included a testbed-specific nfsmount configuration file to each of the nodes. For the application testbed this file is:

```
nfsmount-cfg.h
/*
 * Common nfsmount configuration file
 */
EXTRA(nfsmount.nfsmount) l072d01 l072d02 l072d03 l072d04 l072d05
nfsmount.nfsdetails_l072d01 /shift/lxshare072d/data01 edg-nfs00.cern.ch:\
/shift/lxshare072d/data01 rw,bg,intr,hard
nfsmount.nfsdetails_l072d02 /shift/lxshare072d/data02 edg-nfs00.cern.ch:\
/shift/lxshare072d/data02 rw,bg,intr,hard
nfsmount.nfsdetails_l072d03 /shift/lxshare072d/data03 edg-nfs00.cern.ch:\
/shift/lxshare072d/data03 rw,bg,intr,hard
nfsmount.nfsdetails_l072d04 /shift/lxshare072d/data04 edg-nfs00.cern.ch:\
/shift/lxshare072d/data04 rw,bg,intr,hard
nfsmount.nfsdetails_l072d05 /shift/lxshare072d/data05 edg-nfs00.cern.ch:\
/shift/lxshare072d/data05 rw,bg,intr,hard
```

Creation of the node specific links cannot, to our knowledge, be handled by LCFG, so we created node-type specific scripts to be executed on each node after installation.

4.3.2 NIS Server

As not all the users participating to the EDG project are CERN users, we set up a user account system independent from the standard CERN one which is based on AFS. To keep it simple, we used a standard NIS server installed on the disk server which also hosts the user home directory disk: this allows us to create users and home directories with a single command.

We then configured all UIs as NIS clients for the "edg-tb" NIS domain. This is only partially handled by LCFG. A full configuration requires the inclusion of a NIS client specific configuration file in the LCFG set up and then the execution of a script on each node.

The LCFG part is:

```
nisclient-cfg.h:

+auth.nsswitch ignore
+update.ypserver             edg-nfs00.cern.ch
EXTRA(boot.services)         nsswitch
EXTRA(boot.run)               nsswitch
EXTRA(profile.components)     nsswitch
+nsswitch.mods_passwd   compat
+nsswitch.mods_shadow   files nis
+nsswitch.mods_group    files nis
+nsswitch.mods_hosts    files dns [NOTFOUND=return]
+nsswitch.mods_netgroup files nis
```

This only creates the */etc/nsswitch.conf* and */etc/ypserv.conf* files.

The script to be executed on each node takes care of enabling and starting the ypbind server, also configuring the system **DOMAINNAME** variable:

```
/scripts/setup_enableNIS.sh:

#!/bin/bash
# Set the NISDOMAIN once and for all
domainname edg-tb
sed -e "s/^NISDOMAIN=.*$/NISDOMAIN=edg-tb/" /etc/sysconfig/network > /etc/sysconfig/network.new
mv -f /etc/sysconfig/network /etc/sysconfig/network.old
mv -f /etc/sysconfig/network.new /etc/sysconfig/network

# Add the NIS entries to passwd and group
#echo "ypserver edg-nfs00.cern.ch" >> /etc/yp.conf
echo "+::::::" >> /etc/passwd
echo "+:::" >> /etc/group

# all is ready: start the ypbind daemon
/etc/rc.d/init.d/ypbind start
/sbin/chkconfig ypbind on

# Once NIS client is up, groups exist and the auth object can be executed
cd /etc/obj
bin/runobj auth start
```

4.4 User Interface Machine

The User Interface Machine contains the client software necessary to communicate with the Resource Broker as described in the Users Guide document.

4.4.1 Manual Installation

The list of RPM packages to be installed for each version of EDG software can be viewed and downloaded at the following address: *http://datagrid.in2p3.fr/autobuild/rh6.2/rpmlist*

The RPMs for each EDG component (CE, SE, ...) are divided into several categories (CA, Globus, EDG, ...) this allows to install only the required components.

If you are upgrading a machine where a previous version of EDG is already installed it is strongly recommended to uninstall the EDG software.

To install the edg and globus software you need super-user privileges. All the commands listed below assume to be executed under 'root'.

1. Download and install the certification authorities packages */bin/rpm -ivh ca_**. You will end up with a set of files in the directory */etc/grid-sercurity/certificates* with the extension *.0, *.signing_policy and *.crl_url*.
2. Download and install the packages under the Globus category The files are installed under */opt/globus* and */opt/globus-24*.
3. Download and install the packages under the EDG category The files are installed under */opt/edg* and some of them also on */etc/rc.d/init.d* and */etc/profile.d*
4. Some of the packages in the External category are needed. You may want to select the ones needed or just install all of them.
5. Add the following lines to */etc/ld.so.conf*

   ```
   /opt/globus/lib
   /opt/edg/lib
   /opt/globus-24/lib
   ```

   Run */bin/ldconfig*
6. Copy the template file */opt/edg/etc/UI_ConfigEnv.cfg.template* to */opt/edg/etc/UI_ConfigEnv.cfg* and specify which Resource Broker you will use as you default one. You can visit the CERN Site Status page on *http://marianne.in2p3.fr* for knowing which is the RB currently being used for the applications testbed.
7. Configure GDMP for each VO your site plans to support. Currently edg as a whole supports atlas, alice, cms, lhcb, eo, biome, iteam, wpsix, dzero and tutor. Use the command */opt/edg/sbin/configure_gdmp_client* with the appropriate options.

These 5 steps are common to most nodes. Steps 6 and 7 are specific to a UI node.

4.4.2 LCFG Based Installation

Modify the site-cfg.h file in the source directory of your LCFG server Apart from general settings you have to change the defines that represent the resource broker and L&B server.:

```
/* Resource broker */
#define UI_RESBROKER            lxshare0380.cern.ch
/* Logging and Bookkeeping URL */
#define UI_LOGBOOK              https://lxshare0380.cern.ch:7846
```

Start the update or installation of you UI node as described in the LCFG guide. No additional manual intervention is needed.

4.4.3 Initial Test

A user may wish to customize the UI configuration file to use a different resource broker or to change the sandbox location, for example. This can be done by copying the standard UI configuration file, editing it appropriately, and then setting the environmental variable EDG_WL_UI_CONFIG_PATH to point to the new configuration file.

A rudimentary test of the user interface is to submit a "Hello World" example. Put the following into a file called `hello.jdl`:

```
Executable    = "/bin/echo";
Arguments     = "Hello";
StdOutput     = "hello.out";
StdError      = "hello.err";
OutputSandbox = {"hello.out","hello.err"};
Rank          = other.MaxCpuTime;
```

and submit this job with

```
dg-job-submit hello.jdl
```

The status of the job can be obtained with `dg-job-status` using the job identifier returned from the submit command. The output can be retrieved with `dg-job-get-output` again with the job identifier. The `hello.out` file should contain the word ”Hello”.

For a better introduction to using edg consult the EDG User Guide which can be found on WP6's web page.

4.5 Computing Element Configuration

A computing element consists of a gatekeeper and optionally a set of worker nodes joined by a local resource management system (batch system). If the computing element contains worker nodes, then the home areas of all of the accounts *must* be on a common shared file system with the gatekeeper node.

4.5.1 Manual Installation

Follow the steps 1-5 described for the UI machine, then:

1. Install the host certificate and private key respectively in the following places: */etc/grid-security/hostcert.pem* with permissions set to 0644 and */etc/grid-security/hostkey.pem* with permissions set to 0400.

2. Create the file */etc/sysconfig/globus* containing the following 2 lines:

   ```
   GLOBUS_LOCATION=/opt/globus-24
   GLOBUS_CONFIG=/etc/globus2.conf
   ```

3. Run the Globus initialization script:

   ```
   setenv GLOBUS_LOCATION /opt/globus-24
   $GLOBUS_LOCATION/sbin/globus-initialization.sh
   ```

4. Create and customize the following configuration files:

   ```
   /etc/globus.conf
   /etc/edg/info-mds.conf
   ```

   Examples for these files will be given later. File */etc/globus2.conf* is used for configuring the EDG information system on your site. Below you'll find the file on the CE host *ccgridli03.in2p3.fr* which is a GRIS and a GIIS for the site *cc-in2p3* which registers to the EDG applications testbed information system *edgpro* host *lxshare0373.cern.ch.* The user *edginfo* is used to run the LDAP daemons needed by the information system.

```
[common]
X509_USER_CERT=/etc/grid-security/hostcert.pem
X509_USER_KEY=/etc/grid-security/hostkey.pem
GRIDMAP=/etc/grid-security/grid-mapfile

[mds]
user=edginfo
[mds/gris/provider/gg]
provider=globus-gris

[mds/gris/provider/ggr]
provider=globus-gram-reporter

[mds/gris/provider/edg]

[mds/gris/registration/cc-in2p3]
regname=cc-in2p3
reghn=ccgridli03.in2p3.fr

[mds/giis/cc-in2p3]
name=cc-in2p3

[mds/giis/cc-in2p3/registration/edgpro]
regname=edgpro
reghn=lxshare0373.cern.ch

[gridftp]
```

5. Start the Globus/EDG services on the CE:

```
/sbin/chkconfig globus-gatekeeper on
/etc/rc.d/init.d/globus-gatekeeper start

/sbin/chkconfig globus-mds on
/etc/rc.d/init.d/globus-mds start

/sbin/chkconfig globus-gsincftp on
/etc/rc.d/init.d/globus-gsincftp start

/sbin/chkconfig localloger on
/etc/rc.d/init.d/localloger start
```

6. Increase some system parameters to improve EDG CE scalability and add these commands to rc.local to let them survive reboots. The following script will do the trick.

```
echo 480000 > /proc/sys/fs/inode-max
echo 120000 > /proc/sys/fs/file-max
cp -f /etc/rc.d/rc.local /etc/rc.d/rc.local.orig
cat >> /etc/rc.d/rc.local <<EOD

# Increase some system parameters to improve EDG CE scalability
if [ -f /proc/sys/fs/inode-max ]; then
    echo 480000 > /proc/sys/fs/inode-max
fi
```

```
if [ -f /proc/sys/fs/file-max ]; then
    echo 120000 > /proc/sys/fs/file-max
fi
EOD
```

7. Create, if not present, the */etc/grid-security-local* directory and copy there the host key/cert pair with name hostkey.pem and hostcert.pem.

8. If you are not using a pre-existing */etc/grid-security* area mounted from the NFS server, you must create the */etc/grid-security/gridmapdir* directory and here create one 0-length file for each of the users you created. This can be done using the command line:

```
touch `egrep "[a-z]+[0-9][0-9][0-9]"  /etc/passwd | cut -d ":" -f 1`
```

9. In case you want to use PBS follow the configuration steps given. lxshare0227.cern.ch is the CE and all other nodes mentioned are worker nodes.

   - Set the server name:

     ```
     echo "lxshare0227.cern.ch" > /usr/spool/PBS/server_name
     ```

   - Define the list of WNs in */usr/spool/PBS/server_priv/nodes*. The format is:

     ```
     lxshare0378.cern.ch np=2 edgpro
     ```

     “edgpro” is an arbitrary name which has been used to configure the server.”np” sets the number of concurrent jobs which can be run on the node.

   - Start PBS:

     ```
     /etc/rc.d/init.d/pbs stop
     /etc/rc.d/init.d/pbs start
     ```

     make the daemon startup persistent with:

     ```
     /sbin/chkconfig pbs on
     ```

   - Configure pbs server by loading the pbs_server.conf file which contains the configuration for PBS:

     ```
     /usr/pbs/bin/qmgr < /usr/spool/PBS/pbs_server.conf
     ```

     A sample configuration supporting only two queues is given here:

     ```
     #
     # Create queues and set their attributes.
     #
     #
     # Create and define queue short
     #
     create queue short
     set queue short queue_type = Execution
     set queue short resources_max.cput = 00:15:00
     set queue short resources_max.walltime = 02:00:00
     set queue short enabled = True
     set queue short started = True
     #
     # Create and define queue infinite
     #
     create queue infinite
     ```

```
set queue infinite queue_type = Execution
set queue infinite resources_max.cput = 72:00:00
set queue infinite resources_max.walltime = 240:00:00
set queue infinite enabled = True
set queue infinite started = True
#
# Set server attributes.
#
set server scheduling = True
set server acl_host_enable = False
set server managers = root@lxshare0227.cern.ch
set server operators = root@lxshare0227.cern.ch
set server default_queue = short
set server log_events = 511
set server mail_from = adm
set server query_other_jobs = True
set server scheduler_iteration = 600
set server default_node = edgpro
set server node_pack = False
```

Local Centre Authorization Service (LCAS)

The Local Centre Authorization Service (LCAS) handles authorization requests to the local computing fabric.

In this release the LCAS is a shared library, which is loaded dynamically by the globus gatekeeper. The gatekeeper has been slightly modified for this purpose and will from now on be referred to as edg-gatekeeper.

The authorization decision of the LCAS is based upon the user's certificate and the job specification in RSL (JDL) format. The certificate and RSL are passed to (plug-in) authorization modules, which grant or deny the access to the fabric. Three standard authorization modules are provided by default:

- A module that checks if the user is allowed on the fabric (without lcas the gridmap file is checked).
- A module that checks if the user should be banned from the fabric.
- A module that checks if the fabric is open at this time of the day for datagrid jobs.

All three modules get their information from simple configuration files: allowed_users.db, ban_users.db and timeslots.db, respectively.

For installation and configuration instructions on the edg-gatekeeper and LCAS modules, go to LCAS website[1].

Security

The Gatekeeper must have a valid host certificate and key installed in the `/etc/grid-security` directory. These are usually links to files in the `/etc/grid-security-local` directory.

The Gatekeeper must have all of the security RPMs installed. In addition, the daemon which updates the certificate revocation lists (see 8.2.2) and that which updates the grid mapfile (see Section 23) must also be running. An example mkgridmap configuration file can be found on the Testbed website[2]. The example maps users into pooled accounts based on membership in a virtual organization. These tasks are done by cron jobs.

[1] http://www.dutchgrid.nl/DataGrid/wp4/lcas/edg-lcas-1.0.3/

[2] http://marianne.in2p3.fr/datagrid/documentation/examples/

Configuring the GRAM and GRIS

Here is an example for a `/etc/globus.conf` file as it is used on the CERN application testbed. The node is lxshare0227.cern.ch.

```
GLOBUS_LOCATION=/opt/globus
GLOBUS_GATEKEEPER_SUBJECT="/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0227.cern.ch"
GLOBUS_HOST_DN="hn=lxshare0227.cern.ch, dc=cern, dc=ch, o=Grid"
GLOBUS_ORG_DN="dc=cern, dc=ch, o=Grid"
GLOBUS_GATEKEEPER_HOST="lxshare0227.cern.ch"
GATEKEEPER_PORT=2119
GATEKEEPER_LOG=/var/log/globus-gatekeeper.log
X509_CERT_DIR=/etc/grid-security/certificates
X509_GATEKEEPER_CERT=/etc/grid-security-local/hostcert.pem
X509_GATEKEEPER_KEY=/etc/grid-security-local/hostkey.pem
GRIDMAP=/etc/grid-security/grid-mapfile
GLOBUS_JOBMANAGERS="fork pbs"
GSIWUFTPPORT=2811
GSIWUFTPDLOG=/var/log/gsiwuftpd.log
GLOBUS_FLAVOR_NAME=gcc32dbg
GRID_INFO_EDG=yes
GRIDMAPDIR=/etc/grid-security/gridmapdir/
GRID_INFO_GRIS_REG_GIIS=cern
GRID_INFO_GRIS_REG_HOST=lxshare0227.cern.ch
GLOBUS_GATEKEEPER=/opt/edg/sbin/edg-gatekeeper
GLOBUS_GATEKEEPER_OPTIONS="-lcas_dir /opt/edg/etc/lcas -lcasmod_dir /opt/edg/lib/lcas/"
GLOBUS_GSIWUFTPD_UMASK=002
GRID_INFO_GRIS=yes
GRID_INFO_USER=edginfo
X509_GSIWUFTPD_CERT=/etc/grid-security-local/hostcert.pem
X509_GSIWUFTPD_KEY=/etc/grid-security-local/hostkey.pem
GLOBUS_GRAM_JOB_MANAGER_QSUB=/usr/pbs/bin/qsub
GLOBUS_GRAM_JOB_MANAGER_QDEL=/usr/pbs/bin/qdel
GLOBUS_GRAM_JOB_MANAGER_QSTAT=/usr/pbs/bin/qstat
GLOBUS_GRAM_JOB_MANAGER_MPIRUN=/usr/pbs/bin/qrun
GLOBUS_GRAM_JOB_MANAGER_QSELECT=/usr/pbs/bin/qselect
```

To configure the */etc/edg/info-mds.conf* start with what is present in */etc/edg/info-mds.conf.in.*

The example given is taken from the CERN CE lxshare0227.cern.ch Read the section about the site-cfg.h file and the part about installing via LCFG to get more information about the parameters.

```
WP3_DEPLOY=/opt/edg/info/mds
FTREE_INFO_PORT=2171
FTREE_DEBUG_LEVEL=0
SITE_DN=Mds-Vo-Name=local,o=Grid
SITE_INFO=yes
SITE_NAME=CERN-PRO-1-4
SITE_INSTALLATION_DATE=20021118120000Z
SITE_CPU_RESOURCE_DESCRIPTION=DUAL-PIII-800
SITE_DISK_RESOURCE_DESCRIPTION=15GB-EIDE
SITE_SYSADMIN_CONTACT=hep-project-grid-cern-testbed-managers@cern.ch
SITE_USER_SUPPORT_CONTACT=hep-project-grid-cern-testbed-managers@cern.ch
SITE_SECURITY_CONTACT=hep-project-grid-cern-testbed-managers@cern.ch
```

```
SITE_DATAGRID_VERSION=v1_4_3
SITE_SE_HOSTS=lxshare0393.cern.ch
SITE_CE_HOSTS=lxshare0227.cern.ch:2119/jobmanager-pbs-short,\
lxshare0227.cern.ch:2119/jobmanager-pbs-infinite
NETMON_PRESENT=no
NETMON_PINGER_HOST=lxshare0227.cern.ch
CE_PRESENT=yes
CE_HOST=lxshare0227.cern.ch
CE_BATCHSYSTEM=pbs
CE_CLUSTER_BATCH_SYSTEM_BIN_PATH=/usr/pbs/bin
CE_STATIC_LDIF=/opt/edg/info/mds/etc/ldif/ce-static.ldif
CE_QUEUE=medium,long,short,infinite
CE_CLOSE_SE_ID=lxshare0393.cern.ch
CE_CLOSE_SE_MOUNT_POINT=/flatfiles/SE00
GRID_INFO_USER=edginfo
SITE_NETMON_HOST=no
SITE_NETMON_HOSTS=none
```

Next copy the file `/opt/edg/info/mds/etc/ldif/ce-static.ldif.in` to `/opt/edg/info/mds/etc/ldif/ce-static.ldif` and modify it to reflect the local environment. SpecInt2000 benchmarks can be found at SPEC Website[3]. (For valid tags for RunTimeEnvironment see 23.) The file has been taken from the CE on lxshare0227.cern.ch. The nodes are dual 800MHz PIII nodes. The text on the right hand side is not part of the configuration. It has been put here to provide some description of the parameters.

```
Architecture:        intel   The architecture of the hosts composing the CE
OpSys:               RH 6.2  The operating system of the hosts composing the CE
MinPhysMemory:       512     Minimum value of the physical memory of any WN
MinLocalDiskSpace:   2048    The minimum local disk footprint
NumSMPs:             26      Number of SMP hosts
MinSPUProcessors:    2       The minimum number of SPU processors (for SMP hosts)
MaxSPUProcessors:    2       The Maximum number of SPU processors (for SMP hosts
AverageSI00:         380     Average of the SpecInt2000 benchmark of the WNs
MinSI00:             380     Minimum value of the SpecInt2000 benchmark of the WNs
MaxSI00:             380     Maximum value of the SpecInt2000 benchmark of the WNs
AFSAvailable:        FALSE   Defines if AFS is installed
OutboundIP:          TRUE    Defines if outbound connectivity is allowed
InboundIP:           FALSE   Defines if inbound connectivity is allowed
RunTimeEnvironment:  CMS-1.1.0
RunTimeEnvironment:  ATLAS-3.2.1
RunTimeEnvironment:  ALICE-3.07.01
RunTimeEnvironment:  LHCb-1.1.1
RunTimeEnvironment:  IDL-5.4
RunTimeEnvironment:  CERN-MSS
RunTimeEnvironment:  CMSIM-125
RunTimeEnvironment:  CERN-PRO-1-4
RunTimeEnvironment:  CMS-STRESSTEST-1.0.0
RunTimeEnvironment:  EDG-TEST
```

Once the globus-mds service is started, to see which information is published you can use:

```
ldapsearch -LLL -x -H \
ldap://<CE-hostname>:2135 -b "mds-vo-name=local,o=grid" "(objectClass=*)"
```

[3] http://www.specbench.org/osg/cpu2000/results/cint2000.html

4.5.2 LCFG Based Installation

After doing the appropriate configuration changes to *site-cfg.h* the file *ComputingElement-cfg.h* might need minor changes. These could be in the area of the LCAS object configuration and the parts that refer to the used NFS configuration. Check the *users.h* file and add or remove the virtual users appropriate for the VOs that you support.

Another configuration file that should be checked that it contains current information is *rc-cfg.h*. Make sure that the information about the replica catalogs given for the different supported VOs is correct. This information can be obtained from the VO managers or the Integration team.

If you plan to use PBS, include in the node specific configuration file *pbs-cfg.h* after the *ComputingElement-cfg.h*. This is needed due to dependencies. In case you plan to use your CE in addition as a WN then you have to replace *pbs-cfg.h* with *pbsexechost-cfg.h* and configure it as described in the section about setting up a WN.

Install the CE using LCFG. Then a few minor manual changes are required. Follow the steps described in the Manual Configuration section described in the items 6 to the end. Reboot the machine.

4.5.3 First Test

Check that the following services are running: pbs, globus-gsi_wuftpd, globus-gatekeeper, globus-mds and locallogger.

Check the information published by the node:

```
ldapsearch -LLL -x -H ldap://<CE-hostname>:2135 -b "mds-vo-name=local,o=grid" "(objectClass=*)"
```

Follow instructions given in the Users Guide and run from a UI node a globus-job-run command.

4.6 Worker Node Configuration

In a typical configuration all of the authorization for the worker node is handled by the associated gatekeeper. FTP transfers to and from are usually allowed but they must be initiated by the worker node as it does not run an FTP daemon.

As a consequence of this, the machine does not need to have a host certificate/key, grid-mapfile, or the security RPMs installed. On the other side, the */etc/grid-security/certificates* directory must exists to allow the WNs to verify user and host certificates.

Note: A Worker Node cannot be a Storage Element as well and cannot host a Resource Broker or the Logging and Bookkeeping services.

4.6.1 Manual Installation

Perform the steps 1-5 of the installation of a UI. Then configure the GDMP client in the same way it has been done on the UI.

4.6.2 LCFG Based Installation

Assuming that by now the *site-cfg.h* file has been configured, there is very little that you have to do. In case you use PBS, include the *pbsexechost-cfg.h* file and make sure that you set the parameters for the mount points for /home correctly.

Install the node(s). Check that the rc object has been run and set the access privileges for the $/opt/edg/etc/ < VO >$ directories correctly. They have to be **rwxr-xr-x** and the owner must be set to $root :< VOgrp >$.

4.7 Storage Element Configuration

4.7.1 General Comments

Access control to files is managed in the following way. On the SE group ID per VO has been created. All users belonging to a given VO have their certificates mapped in the grid-mapfile of the SE to a local user with the VOs group ID. The directory used by GDMP to replicate files is group writable by the VO group ID only and has a group sticky bit set (see the details further in these instructions). This prevents users belonging to other VOs to write/use the directory or files contained.

The storage element must also run a gatekeeper () and an FTP daemon (xref linkend="gsiftpd"/⟩). See the appropriate sections for the proper configuration of these daemons.

To run a storage element a host certificate is required.

If you want to provide the access protocol "file" you have to use a shared file system and make the storage area accessible from the WNs. This area is on most systems at the location */flatfiles* and contains directories for the various VOs.

In case you want to provide RFIO access install the castor-rfio and castor-rfio-devel RPMS. To use these make sure that the $PATH variable includes the path to the RFIO commands.

There is a script "rfiod.scripts" in the rfio sub-directory of the castor distribution that can be used to start/stop/restart/check the presence of rfiod (i.e. to start run rfiod.scripts start). This works for all Unix platforms.

Its not advisable to use inetd to start rfios as the requests can come very quickly and cause inetd to think the process is looping and refuse to start a new rfiod.

4.7.2 Manual Installation

1. Follow the steps 1-5 given in the UI section. Install the certificates as described in the CE section and follow the steps leading to running the Globus initialization scripts in this section.

2. The same configuration files that are needed for the CE are needed for an SE, however the contents of */etc/globus2.conf* differs. Here is, as an example, the file used on CERN's SE (lxshare0393).

```
[common]
GLOBUS_LOCATION=/opt/globus-24
globus_flavor_name=gcc32dbg
X509_USER_CERT=/etc/grid-security-local/hostcert.pem
X509_USER_KEY=/etc/grid-security-local/hostkey.pem
GRIDMAP=/etc/grid-security/grid-mapfile
GRIDMAPDIR=/etc/grid-security/gridmapdir/

[mds]
user=edginfo

[mds/gris/provider/gg]
provider=globus-gris

[mds/gris/provider/ggr]
provider=globus-gram-reporter

[mds/gris/provider/edg]

[mds/gris/registration/site]
```

```
regname=cern
reghn=lxshare0227.cern.ch

[gridftp]
```

3. Configure GDMP for each VO your site supports. Use the command **/opt/edg/sbin/configure_gdmp** with the appropriate options, consult information provided on the WP2 website. This will create a directory for each VO under */opt/edg/etc*, and you must modify the configuration files for each VO. For example, for the VO 'alice', you will have to modify */opt/edg/etc/alice/gdmp.conf* and */opt/edg/etc/alice/gdmp.private.conf*. Here is an example for */opt/edg/etc/alice/gdmp.conf*:

   ```
   GDMP_SHARED_CONF=/opt/edg/etc/gdmp.shared.conf
   GDMP_SERVICE_NAME=host/lxshare0393.cern.ch
   GDMP_VIRTUAL_ORG=alice
   GDMP_CONFIG_DIR=/opt/edg/etc/alice
   GDMP_VAR_DIR=/opt/edg/var/alice
   GDMP_TMP_DIR=/opt/edg/tmp/alice
   GDMP_GRID_MAPFILE=/opt/edg/etc/alice/grid-mapfile
   GDMP_SERVER_PROXY=/opt/edg/etc/gdmp_server.proxy
   GDMP_PRIVATE_CONF=/opt/edg/etc/alice/gdmp.private.conf
   GDMP_STORAGE_DIR=/flatfiles/SE00/alice
   GDMP_STAGE_FROM_MSS=/opt/edg/alice/bin/stage_from_mss.sh
   GDMP_STAGE_TO_MSS=/opt/edg/alice/bin/stage_to_mss.sh
   ```

   and for */opt/edg/etc/alice/gdmp.private.conf*:

   ```
   GDMP_REP_CAT_HOST=ldap://grid-vo.nikhef.nl:10489
   GDMP_REP_CAT_NAME=AliceReplicaCatalog
   GDMP_REP_CAT_MANAGER_CN=Manager
   GDMP_REP_CAT_MANAGER_PWD=THE PASSWORD
   GDMP_REP_CAT_CN=dc=eu-datagrid,dc=org
   GDMP_REP_CAT_FILE_COLL_NAME=Alice WP1 Repcat
   GDMP_REP_CAT_MANAGER_DN=cn=${GDMP_REP_CAT_MANAGER_CN},rc=${GDMP_REP_CAT_NAME},\
   ${GDMP_REP_CAT_CN}
   GDMP_REP_CAT_URL=${GDMP_REP_CAT_HOST}/rc=${GDMP_REP_CAT_NAME},${GDMP_REP_CAT_CN}
   GDMP_REP_CAT_FILE_COLL_URL=${GDMP_REP_CAT_HOST}/lc=${GDMP_REP_CAT_FILE_COLL_NAME},\
   rc=${GDMP_REP_CAT_NAME},${GDMP_REP_CAT_CN}
   GDMP_REP_CAT_OBJECTIVITY_COLL_URL=${GDMP_REP_CAT_HOST}/lc=${GDMP_REP_CAT_OBJYFILE_COLL_NAME},\
   rc=${GDMP_REP_CAT_NAME},${GDMP_REP_CAT_CN}
   ```

   Note that this file contains the password for the VO specific replica catalog which you can get from the VO manager or Integration team.

4. Start the Globus/EDG services on the CE:

   ```
   /sbin/chkconfig globus-gatekeeper
   /etc/rc.d/init.d/globus-gatekeeper start

   /sbin/chkconfig globus-mds
   /etc/rc.d/init.d/globus-mds start

   /sbin/chkconfig globus-gsincftp
   /etc/rc.d/init.d/globus-gsincftp start
   ```

Note: the GDMP server is started by 'inetd'.

There are now a few steps that the manual and LCFG based installation have in common.

- If you are not using a pre-existing */etc/grid-security* area mounted from the NFS server, you must create the */etc/grid-security/gridmapdir* directory and here create one 0-length file for each of the users you created. This can be done using the command line:

  ```
  touch ‘egrep "[a-z]+[0-9][0-9][0-9]"  /etc/passwd | cut -d ":" -f 1‘
  ```

- If you have installed the edg_query_vo_storage rpm, then you should create the corresponding configuration file. Copy the templates and then edit them following the instructions given in the files */opt/edg/etc/edg_query_vo_storage.conf.template* and */opt/edg/etc/edg_query_vo_storage-cron.conf.template*.

4.7.3 LCFG Based Installation

- Configure for your SE the file *StorageElement-cfg.h*. Configure the LCAS section of the file, update the VO specific sections and set the NFS parameters to reflect your setup.
- Install the node with LCFG.
- Follow the steps that have been mentioned in the previous section as being the same for manual and automated installation.
- Configure GDMP with: **/etc/obj/gdmp start**
- Reboot the node to get everything started

4.7.4 mkgridmap.conf

After the initial installation and configuration make sure that a correct */opt/edg/etc/mkgridmap.conf* file has been created.

Apart from the VO specific lines, giving the ldap addresses of the VOs, for the SE this file has to contain the special storage element VO.

To get some orientation have a look at the file used at CERN.

```
#### GROUP: group URI [lcluser]
#
# EDG Standard Virtual Organizations
group ldap://grid-vo.nikhef.nl/ou=testbed1,o=alice,dc=eu-datagrid,dc=org .alice
group ldap://grid-vo.nikhef.nl/ou=testbed1,o=atlas,dc=eu-datagrid,dc=org .atlas
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=cms,dc=eu-datagrid,dc=org .cms
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=lhcb,dc=eu-datagrid,dc=org .lhcb
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=biomedical,dc=eu-datagrid,dc=org .biome
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=earthob,dc=eu-datagrid,dc=org .eo
group ldap://marianne.in2p3.fr/ou=ITeam,o=testbed,dc=eu-datagrid,dc=org .iteam
group ldap://marianne.in2p3.fr/ou=wp6,o=testbed,dc=eu-datagrid,dc=org .wpsix
group ldap://grid-vo.nikhef.nl/ou=testbed1,o=dzero,dc=eu-datagrid,dc=org .dzero
group ldap://marianne.in2p3.fr/ou=EDGtutorial,o=testbed,dc=eu-datagrid,dc=org .tutor
#
# Other Virtual Organizations
#group ldap://grid-vo.cnaf.infn.it/ou=testbed1,o=infn,c=it .infngrid
```

```
#group ldap://vo.gridpp.ac.uk/ou=testbed,dc=gridpp,dc=ac,dc=uk .gridpp
#group ldap://babar-vo.gridpp.ac.uk/ou=babar,dc=gridpp,dc=ac,dc=uk .babar
#

# Following group is to get SE (GDMP) host certs ...
#group ldap://grid-vo.nikhef.nl/ou=devtb,o=gdmpservers,dc=eu-datagrid,dc=org gdmp
group ldap://grid-vo.nikhef.nl/ou=apptb,o=gdmpservers,dc=eu-datagrid,dc=org gdmp

#### Optional - DEFAULT LOCAL USER: default_lcluser lcluser
#default_lcluser .

#### Optional - AUTHORIZED VO: auth URI
auth ldap://grid-vo.nikhef.nl/ou=People,o=gdmpservers,dc=eu-datagrid,dc=org
auth ldap://marianne.in2p3.fr/ou=People,o=testbed,dc=eu-datagrid,dc=org

#### Optional - ACL: deny|allow pattern_to_match
#allow *INFN*

#### Optional - GRID-MAPFILE-LOCAL
#gmf_local /opt/edg/etc/grid-mapfile-local
```

The *grid-mapfile-local* file contains a list of certificates which will be included in addition to the items added during the periodic update of the file.

4.8 Replica Catalog

Here the differences between manual and LCFG based installation are marginal.

- Install a node using the RPMs given in the RC RPM list in your preferred way.
- Follow the instructions given by WP2 to configure the Replica Catalog. The Instructions are included in the **edg-rc-server-3.1.x** RPM and are installed at */opt/edg/edg-rc/server/README*

4.9 Resource Broker

For Testbed 1, the resource broker machine contains the resource broker itself, the job submission service, a logging and bookkeeping server. The information index has been moved to a different node and is replaced by the BDII, based on a standard LDAP server using the schemas previously used by the II. Each of these must be configured as well as some external software upon which these depend.

The resource broker machine must also be running an grid-ftp daemon.

For full functionality, `sendmail` must be available on the resource broker machine and must be in the path of the user running the various daemons.

Due to some limitations of the RB, multiple RBs in the application testbed have to be deployed. This means if you are a site running many UIs with active users you should consider setting up additional RBs.

4.9.1 Security

The resource broker must have a valid host certificate and key installed in the `/etc/grid-security` directory. In addition, copies of these files must be in the `.hostcert` subdirectory of the account running the resource broker daemons, usually dguser.

The resource broker must have all of the security RPMs installed. In addition, the daemon which updates the certificate revocation lists (see 8.2.2) and that which updates the grid mapfile (see 23) must also be running. An example mkgridmap configuration file can be found on the EDG documentation web page[4].

4.9.2 External Packages

CondorG

The resource broker relies on CondorG and the ClassAds from the Condor team. The RPMs for these packages must be installed and can be obtained from the EDG package repository.

CondorG runs several daemons under an unpriviledged account. You must create this account before installing CondorG. The recommended name is ”dguser”.

4.9.3 Installation and Configuration

First the procedures will be given that are unique to the manual part. After this the LCFG based procedure will be described and then the common, additional manual configuration steps are given.

Manual Installation

As with the other nodes, download and install the RPM lists. Follow the first 5 steps of the UI installation.

Then configure */etc/globus.conf* following the example given here:

```
GLOBUS_LOCATION=/opt/globus
GLOBUS_HOST_DN="hn=lxshare0227.cern.ch, dc=cern, dc=ch, o=Grid"
GLOBUS_ORG_DN="dc=cern, dc=ch, o=Grid"
GRIDMAP=/etc/grid-security/grid-mapfile
GRIDMAPDIR=/etc/grid-security/gridmapdir/
GSIWUFTPPORT=2811
GSIWUFTPDLOG=/var/log/gsiwuftpd.log
GLOBUS_FLAVOR_NAME=gcc32dbg
X509_GATEKEEPER_CERT=/etc/grid-security-local/hostcert.pem
X509_GATEKEEPER_KEY=/etc/grid-security-local/hostkey.pem
X509_GSIWUFTPD_CERT=/etc/grid-security-local/hostcert.pem
X509_GSIWUFTPD_KEY=/etc/grid-security-local/hostkey.pem
```

Create the required local users: **mysql, postgres** and **dguser**

On the RB there are several services that require access to a valid proxy. The proxies are generated by the SysV startup script with a default time of 24 hours. Add

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

to the `/etc/crontab`. In addition the gridmapfiles and the CRL files have to be updated on a regular basis. Add in addition:

```
53 1,7,13,19 * * * root /opt/edg/etc/cron/mkgridmap-cron
53 1,7,13,19 * * * root /opt/edg/etc/cron/edg-fetch-crl-cron
```

[4] http://marianne.in2p3.fr/datagrid/documentation/

LCFG based Installation

As always, make sure the *site-cfg.h* file reflects your site. Have a look at *ResourceBroker.h*. There shouldn't be any need to modify this file. Install the node.

Common Interactions

1. As user **root** increase some of the system limits to make the RB more robust. Execute the following lines:

```
# Increase some default system parameters for out greedy RB
echo 480000 > /proc/sys/fs/inode-max
echo 120000 > /proc/sys/fs/file-max
echo 1024 7999 > /proc/sys/net/ipv4/ip_local_port_range

# To make these modifications permanent, we add them to rc.local
cp -f /etc/rc.d/rc.local /etc/rc.d/rc.local.orig
cat >> /etc/rc.d/rc.local <<EOD

# Increase some system parameters to improve EDG RB scalability
if [ -f /proc/sys/fs/inode-max ]; then
    echo 480000 > /proc/sys/fs/inode-max
fi
if [ -f /proc/sys/fs/file-max ]; then
    echo 120000 > /proc/sys/fs/file-max
fi
if [ -f /proc/sys/net/ipv4/ip_local_port_range ]; then
    echo 1024 7999 > /proc/sys/net/ipv4/ip_local_port_range
fi
EOD
```

2. Create the directories for the host key and the host certificate and move them there. According to the given globus.conf, the paths should be */etc/grid-security-local/hostkey.pem* and */etc/grid-security-local/hostcert.pem*

3. Create links in */etc/grid-security*:

```
ln -s /etc/grid-security-local/hostkey.pem /etc/grid-security/hostkey.pem
ln -s /etc/grid-security-local/hostcert.pem /etc/grid-security/hostcert.pem
```

4. Copy the credentials to the dguser directory.

```
mkdir /home/dguser/.hostcert
cp /etc/grid-security-local/* /home/dguser/.hostcert/
chown -R dguser:dguser /home/dguser/.hostcert
```

5. Map all users to the user **dguser** to do this the configuration file for mkgridmap has to be changed.

```
mv /opt/edg/etc/mkgridmap.conf /opt/edg/etc/mkgridmap.conf.orig
cat > /opt/edg/etc/mkgridmap.conf.rb <<EOD
group ldap://marianne.in2p3.fr/ou=guidelines,o=testbed,dc=eu-datagrid,dc=org dguser
auth ldap://marianne.in2p3.fr/ou=People,o=testbed,dc=eu-datagrid,dc=org
gmf_local /opt/edg/etc/grid-mapfile-local
EOD
cp /opt/edg/etc/mkgridmap.conf.rb /opt/edg/etc/mkgridmap.conf
```

6. The mkgridmap script is run every 6 hours. To get an initial update start the process from the command line: **/opt/edg/etc/cron/mkgridmap-cron**

7. Configure CondorG. You have to be user **dguser** for some of the operations. The setup requires input from the user. All the defaults are correct and you have just to accept them.

   ```
   su dguser
   /opt/CondorG/setup.sh
   ```

8. Setup */home/dguser/.bashrc* and */home/dguser/workload_setup.sh*

   ```
   # .bashrc
   # User specific aliases and functions
   if [ -f ~/workload_setup.sh ]; then
     . ~/workload_setup.sh
   fi

   # Source global definitions
   if [ -f /etc/bashrc ]; then
           . /etc/bashrc
   fi
   ```

   and workload_setup.sh:

   ```
   # Point to the CondorG installation path and configuration file.
   CONDORG_INSTALL_PATH=/home/dguser/CondorG
   export CONDORG_INSTALL_PATH
   CONDOR_CONFIG=$CONDORG_INSTALL_PATH/etc/condor_config
   export CONDOR_CONFIG

   # Replica catalog API is needed by resource broker.
   GDMP_INSTALL_PATH=/opt/edg
   export GDMP_INSTALL_PATH

   # Setup the user and database area for the postgresql database.
   # This is used by the resource broker.
   PGSQL_USER=postgres
   export PGSQL_USER
   PGDATA=/opt/data
   export PGDATA
   PGSQL_INSTALL_PATH=/usr/bin/psql
   export PGSQL_INSTALL_PATH

   # Add paths to the shared library path.
   for p in \
       "${CONDORG_INSTALL_PATH}/lib" \
       "${GDMP_INSTALL_PATH}/lib"
   do
       if ! printenv LD_LIBRARY_PATH | grep -q "${p}"; then
           if [ -n "${LD_LIBRARY_PATH}" ]; then
               LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${p}"
           else
               LD_LIBRARY_PATH="${p}"
           fi
   ```

```
    fi
done
export LD_LIBRARY_PATH

# Add condor binaries to the path.
for p in \
    "$CONDORG_INSTALL_PATH/sbin" \
    "$CONDORG_INSTALL_PATH/bin" \
    "/usr/sbin"
do
    if ! printenv PATH | grep -q "${p}"; then
        PATH="${p}:${PATH}"
    fi
done
export PATH

# MUST add the libraries for the 2.95.2 run time libraries.
for p in \
    "/usr/local/lib"
do
    if ! printenv LD_LIBRARY_PATH | grep -q "${p}"; then
        if [ -n "${LD_LIBRARY_PATH}" ]; then
            LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${p}"
        else
            LD_LIBRARY_PATH="${p}"
        fi
    fi
done
export LD_LIBRARY_PATH
```

9. Configure */home/dguser/CondorG/etc/condor_config*. At the top of the file:

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

   (changing the hostname to the host of your resource broker) and modify the following parameters to have the given values:

   ```
   CRED_MIN_TIME_LEFT = 0
   GLOBUSRUN = \$(GLOBUS\_LOCATION)/bin/globusrun
   ```

   You may also wish to modify the CONDOR_ADMIN parameter to set the recipient of email to something other than the dguser account.

10. Return to the **root** account.

11. CondorG currently requires the Globus1 style signing policy file in which all of the policies appear in a single file. Until this is fixed by Condor, you must do the following:

    ```
    cat /etc/grid-security/certificates/*.signing_policy \
      > /etc/grid-security/certificates/ca-signing-policy.conf
    ```

12. Configure Postgres. First initialize the default DB area. Use the following lines:

```
mkdir /opt/data
chown postgres:postgres /opt/data
su postgres
initdb -D /opt/data
exit
```

13. Activate Postgresql. To make operation more simple add the following lines to */etc/rc.d/init.d/postgresql*:

```
# Use EDG data location
export PGDATA=/opt/data
```

in the *start()* section just before:

```
# Check for the PGDATA structure
```

Then change in the line:

```
su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl  -D \
 $PGDATA -p /usr/bin/postmaster start  > /dev/null 2>&1" < /dev/null
```

the output file from */dev/null* to

```
 /var/tmp/postgres.log 2>&1
```

As a last change add in the *stop()* section just before the su -l postgres

```
# Use EDG data location
export PGDATA=/opt/data
```

Now to start it:

```
/sbin/chkconfig postgresql on
/etc/rc.d/init.d/postgresql start
```

14. Give the privileges to create DBs to **dguser**.

```
su postgres
createuser <<EOD
dguser
y
n
EOD
exit
```

15. Since the Information index is not run on this node anymore, no instruction to configure it is given.

16. Configure the Resource Broker services. In the sample code lxshare0380 is the RB, lxshare0375 is the proxy server node and lxshare0225 is the BDII node. Copy */opt/edg/etc/rb.conf.template* to */opt/edg/etc/rb.conf.template* and edit it:

```
[
  MDS_contact = "lxshare0225.cern.ch";
  MDS_port = 2170;
  MDS_timeout = 60;
  MDS_gris_port = 2135;
  MDS_basedn = "mds-vo-name=local,o=grid";

  MDS_multi_attributes = {

"AuthorizedUser",
"RunTimeEnvironment",
"CloseCE"
  };

  LB_contact = "lxshare0380.cern.ch";
  LB_port = 7846;

  JSS_contact = "lxshare0380.cern.ch";
  JSS_client_port = 8881;
  JSS_server_port = 9991;

  JSS_backlog = 5;
  UI_backlog  = 5;

  UI_server_port   = 7771;

  RB_pool_size = 512;
  RB_notification_queue_size = 32;
  RB_purge_threshold = 600000;
  RB_cleanup_threshold = 3600;
  RB_sandbox_path = "/tmp";
  RB_logfile="/var/tmp/RBserver.log";
  RB_logfile_size=512000000;
  RB_logfile_level=7;
  RB_submission_retries=3;
  MyProxyServer="lxshare0375.cern.ch";
  SkipJobSubmission = false;
]
```

17. Now start the service

    ```
    /sbin/chkconfig broker on
    /etc/rc.d/init.d/broker start
    ```

18. To configure the Job Submission service edit */opt/edg/etc/jss.conf*. In case you use LCFG you just have to copy the template file.

    ```
    [
    Condor_submit_file_prefix   = "/var/tmp/CondorG.sub";
    Condor_log_file           = "/var/tmp/CondorG.log";
    Condor_stdoe_dir            = "/var/tmp";
    Job_wrapper_file_prefix     = "/var/tmp/Job_wrapper.sh";
    Database_name               = "template1";
    Database_table_name         = "condor_submit";
    ```

```
JSS_server_port             = 8881;
RB_client_port     = 9991;
Condor_log_file_size        = 64000;
]
```

19. Configure */opt/edg/etc/wl-jss_rb-env.sh*

```
mv /opt/edg/etc/wl-jss_rb-env.sh /opt/edg/etc/wl-jss_rb-env.sh.orig
cat /opt/edg/etc/wl-jss_rb-env.sh.orig | \
  sed -e "s/CONDOR_IDS=/CONDOR_IDS=\${CONDOR_IDS\:\-2002\.2002}/" \
  > /opt/edg/etc/wl-jss_rb-env.sh.rb
cp /opt/edg/etc/wl-jss_rb-env.sh.rb /opt/edg/etc/wl-jss_rb-env.sh
```

20. */var/tmp/CondorG.log* must exist and be owned by the dguser account before starting the job submission service.

```
touch /var/tmp/CondorG.log
chown dguser:dguser /var/tmp/CondorG.log
```

21. Start the Jobsubmission

```
/sbin/chkconfig jobsubmission on
/etc/rc.d/init.d/jobsubmission start
```

22. The logging and bookkeeping services keep track of the state of submitted jobs and record data useful for debugging problems. These services use a MySQL database to store this information; the database resides on the Resource Broker machine. The locallogger daemons must run on all gatekeeper and resource broker nodes. The lbserver daemons only need to run on the resource broker.

    On a Resource Broker a MySQL has to be run under a non privileged account. The following steps walk you through the required configuration. You have to choose a password for the server. In this example the password is **globus_admin**.

```
mkdir /var/lib/mysql
mkdir /var/lib/mysql/test
mkdir /var/lib/mysql/mysql
chown -R mysql /var/lib/mysql
/usr/bin/mysql_install_db
chown -R mysql /var/lib/mysql
chmod -R og-rw /var/lib/mysql/mysql
/sbin/chkconfig mysql on
/etc/rc.d/init.d/mysql start
```

23. Now configure the running MySQL server, first set the password:

```
/usr/bin/mysqladmin -u root password 'globus_admin'
```

    Then use it for the next commands to setup the default tables for logging and bookkeeping:

```
/usr/bin/mysqladmin -u root  -p create lbserver
/usr/bin/mysql -u root -p -e \
'grant create,drop,select,insert,update,delete on lbserver.* to lbserver@localhost'
/usr/bin/mysql -u lbserver lbserver < /opt/edg/etc/server.sql
```

To insure that MySQL is started, as required before logging and bookkeeping servers change the entry in */etc/rc.d/rc3.d*

```
mv /etc/rc.d/rc3.d/S90mysql /etc/rc.d/rc3.d/S85mysql
```

24. Now start the logging and bookkeeping services

```
/sbin/chkconfig lbserver on
/sbin/chkconfig locallogger on
/etc/rc.d/init.d/lbserver start
/etc/rc.d/init.d/locallogger start
```

25. As a final step reboot the machine.

Preliminary Tests

The tests here are very limited, but still useful.

To quickly check that the Postgres installation worked, you can create a dummy database as the user running the resource broker daemons:

```
su - dguser
createdb test
psql test
```

There should be no errors from these two commands.

A quick check to see if the server is responding is the following:

```
openssl s_client -connect \
lxshare0380.cern.ch:7846 -state -debug
```

This should respond verbosely with information about the SSL connection. Any error indicates a problem with the certificates. You will have to interrupt this command to get back to the command line.

4.10 MyProxy Server

For long-lived jobs there is the possibility that the job will outlive the validity of its proxy causing the job to fail. To avoid this, the workload management tools allow a proxy to be automatically renewed via a MyProxy server. The MyProxy server manages a long-lived proxy generated by a user and gives updated proxies to properly authenticated processes on behalf of the user.

The usual configuration is to have one MyProxy server per resource broker machine. The MyProxy server should run on a separate well-secured machine.

4.10.1 Security

The MyProxy must have a valid host certificate and key installed in the `/etc/grid-security` directory.

The MyProxy server must have all of the security RPMs installed. In addition, the daemon which updates the certificate revocation lists (see 8.2.2) must also be running.

4.10.2 Configuration

There is a single configuration file `/opt/edg/etc/edg-myproxy.conf` which should be filled with the subject names of associated resource brokers.

The SysV initialization script remakes the configuration file from the information in the `edg-myproxy.conf` and from the "signing policy" files in `/etc/grid-security/certificates`. This is done every time the daemon is started, so all changes are reflected in the running daemon when it is restarted.

4.10.3 Manual Configuration

Download and install the RPMs as described for the previous nodes. Go through the initial configuration steps. Don't forget the cron tab entries for the CRL update daemon. Add the subjects of the associated resource brokers to the `/opt/edg/etc/edg-myproxy.conf` file. At CERN for example a quite large number of nodes is supported:

```
/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0380.cern.ch
/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0383.cern.ch
/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0382.cern.ch
/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0381.cern.ch
/C=IT/O=INFN/OU=host/L=CNAF/CN=grid010g.cnaf.infn.it/Email=sitemanager@cnaf.infn.it
/C=IT/O=INFN/OU=host/L=CNAF/CN=grid004f.cnaf.infn.it/Email=sitemanager@cnaf.infn.it
/C=IT/O=INFN/OU=www server/L=Catania/CN=genius.ct.infn.it/\
Email=falzone@ct.infn.it,roberto.barbera@ct.infn.it
/C=IT/O=INFN/OU=User Interface/L=Catania/CN=grid008.ct.infn.it/\
Email=patrizia.belluomo@ct.infn.it
/C=IT/O=INFN/OU=www server/L=Catania/CN=grid009.ct.infn.it/\
Email=falzone@ct.infn.it
/C=IT/O=INFN/OU=gatekeeper/L=PD/CN=grid012.pd.infn.it/\
Email=Marco.Verlato@padova.infn.it
/C=IT/O=INFN/OU=datagrid-genius/L=Pisa/CN=genius.pi.infn.it/\
Email=livio.salconi@pi.infn.it
/C=IT/O=INFN/OU=GRID UI/L=CNAF Bologna/CN=genius.cnaf.infn.it/\
Email=stefano.zani@cnaf.infn.it
/C=IT/O=INFN/OU=gatekeeper/L=CA/CN=grid004.ca.infn.it/\
Email=daniele.mura@ca.infn.it
```

To start the server:

```
/sbin/chkconfig myproxy on
/etc/rc.d/init.d/myproxy start
```

4.10.4 LCFG based Installation

Configure the server by editing the part of the *Proxy-cfg.h* file that deals with the configuration of the myproxy object. Edit the line to contain subjects of all trusted RBs.

```
+myproxy.trusted /O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0380.cern.ch \
/O=Grid/O=CERN/OU=cern.ch/CN=host/lxshare0383.cern.ch
```

Copy the host certificate to the locations given in */etc/globus.conf*. Start the services as described in the previous section

4.11 BDII Node

The BDII node runs a standard open LDAP server to provide index information of the resources available. This information is updated by queries to the MDS system in regular intervals.

4.11.1 Manual Installation

Download the RPMs as before and install them on the target node.Follow the first 5 steps of the UI installation. Add to the line:

```
3,13,23,33,43,53 * * * * EDG_LOCATION_/etc/cron/bdii-cron 1>/dev/null 2>&1
```

to the */etc/crontab* file.

Follow the steps that are common with the LCFG based installation.

4.11.2 LCFG based Installation

Install a node using the *BDII-cfg.h* file. No changes are needed.

4.11.3 Common Configuration

- In */opt/edg/etc* copy the *bdii.conf.template* template file to *bdii.conf*
- Edit */opt/edg/etc/bdii.conf* for your site. This usually means changing only BDII_HOST, BDII_PASSWD, and BDII_PASSWD_PLAIN.

  To generate a new BDII_PASSWD(_PLAIN) pair you can use the */opt/openldap/sbin/slappasswd* command which works just like passwd for Unix.

  If you are installing the BDII node for a testbed different from the EDG application testbed, then you also need to change MDS_HOST which is the node highest in the MDS hierarchy.
- Copy the */opt/edg/etc/init.d/bdii* file to */etc/rc.d/init.d* (or create a soft link), activate the service with chkconfig, and start the slapd:

  ```
  cp /opt/edg/etc/init.d/bdii /etc/rc.d/init.d/bdii
  /sbin/chkconfig bdii on
  /etc/rc.d/init.d/bdii start
  ```

4.12 MDS Node

WP3 delivered Information Servers based on LDAP which has been deployed in the EDG testbed.

4.12.1 LCFG based Installation

After configuring *site-cfg.h* edit *MDS-cfg.h* to set the name of the top level giis and gris.

```
globuscfg.gris      alledg
globuscfg.giis      edgpro
```

No host cerificate is needed for this host. Install the node.

Since there is only a single Top-MDS node in EDG and this node has been setup using LCFG no information is given on manual installation.

System administrators should register their site with the appropriate GIIS at the next highest level.

4.13 Network Monitoring Node

Empty

5 Testing Your Installation

A set of tests, included in the release, help verify the correct installation and configuration of your site. The edg-site-certification package contains these tests. All of the tests are invoked in a similar manner:

```
edg-testbed-test BaseTest::ReverseDNS alpha.example.org
```

where this would invoke the Reverse DNS lookup test for the host alpha.example.org. An overview of the tests can be found in the directory `/opt/edg/share/doc/edg-site-certification*`. To run these tests the directory `/opt/edg/bin` must be in your path and PERLLIB must include `/opt/edg/lib/perl`. Both should be done automatically if the EDG profile scripts are properly installed.

The EDG Users' Guide contains a set of simple examples which are also useful in testing your site. It also gives a useful overview of the testbed from the user's perspective.

Further information can be obtained from the Testing Group's web page[1].

[1] http://marianne.in2p3.fr/datagrid/TestPlan/

6 Support

6.1 Contacts

The user's first point of contact for operational problems is the local site administrator. As such, you should try to answer questions from your local users and answer them if possible. System adminitrators and users are welcome and encouraged to use the bug-reporting facility. As a last resort, users may contact the Integration Team[1].

System administrators and encouraged to use routinely the bug-reporting system, Bugzilla[2], for flaws in the DataGrid software. For installation and configuration assistance, system administrators are first referred to the documentation but are welcome to contact the Integration Team[3] directly.

6.2 Website

The main WP6 website[4] contains documentation, contact information, the bug-reporting system, links to the source and packages repositories as well as links to other sites. This serves and the single point-of-access to information about the testbed activities.

[1] mailto:hep-proj-grid-integration-team@cern.ch
[2] http://marianne.in2p3.fr/datagrid/bugzilla/
[3] mailto:hep-proj-grid-integration-team@cern.ch
[4] http://marianne.in2p3.fr/

7 Information Systems Configuration

WP3 have delivered Information Servers based on LDAP, and information Services based on Relational Databases. In the testbeds only the LDAP based are deployed.

While the various components of the information systems can run as root, this is not recommended. Instead you should create an unpriviledged user for this.

The information services tree is rooted at the node `lxshare0373.cern.ch`. The other nodes of the information systems are:

- Currently empty list.

System administrators should register their site with the appropriate GIIS at the next highest level as has been explained in the sections about setting up CEs and SEs.

8 Appendix

In these sections we collect information that might deepen the understanding or might help configuring certain components. In general this information is not needed to setup a EDG testbed.

8.1 Time Synchronization

Time plays a vital role when checking the validity of certificates. Consequently it is vital that the DataGrid machines be clients of a reliable time server.

If the machines at your site are not already syncronized, then you may use the xntp3 package distributed with the other external packages used by EDG (See 1.4). This package implements the network time protocol (ntp)[1] and allows a machine to be a time client (as well as a time server).

If you use the xntp3 package, then configuring your machine as a time client is rather trivial. You must add at least one time server reference to the ntp configuration file `/etc/ntp.conf` and configure the machine to run the ntp daemon. The detailed steps are:

- Identify one or more independent time servers to use. If your site does not have a standard time server, then consult the list of public time servers[2].
- The ntp protocol uses the port 123 (tcp and udp); ensure that your firewall will not filter these packets.
- Set the clock to the time of one of the servers

  ```
  /usr/sbin/ntpdate ip-time-1.cern.ch
  ```

  replacing the CERN time server with the one you have chosen.
- Synchronize the hardware clock on the machine to the system clock.

  ```
  /sbin/hwclock --systohc
  ```

- For each time server, add an entry in the `/etc/ntp.conf` file like ”`server ip-time-1.cern.ch`” replacing the CERN time server with the one you have chosen. You may specify multiple servers.
- Force the ntp daemon to start automatically at boot time.

  ```
  /sbin/chkconfig xntpd on
  ```

- Start the ntp daemon

  ```
  /sbin/service xntpd start
  ```

  (or reboot the machine). You can check the status of the time server by using the ”`/usr/sbin/ntpq`” command.

[1] http://www.eecis.udel.edu/ ntp/

[2] http://www.eecis.udel.edu/ mills/ntp/servers.htm

The xntp3 package tries to be rather gentle to the system when readjusting the time. One result is that if the time is too far wrong (more than 1000s by default), then the daemon will simply refuse to reset the clock and will die. This is a common problem if you forget to use the `ntpdate` above.

It is extremely important that the hardware clock is synchronized to the system clock. If not, at the next boot an unsynchronized time will be reloaded and you risk having the time syncronization daemon stop.

If you have a large number of machines, you may wish to create a local time server. Refer to the local documentation of the xntp3 package for instructions.

8.2 Authentication, Authorization, and Security

8.2.1 Certificates

Cryptographic certificates are used to attest to the identity of an user or machine to the extent specified in the issuing certification authority's (CA) policy documents. Users accessing DataGrid resources must have a valid certificate; similarly, hosts offering services within the testbed must also have one.

The EDG-approved CAs have service areas which cover most of Europe and the United States. (Consult the current list[3] on the web.) If a user or site is not covered by an existing CA's service area, then one must either negociate with a CA to extend its service area or start a new CA.

It has been agreed that the CA operated in Lyon is responsible for users without access to a CA.

Installing User Certificates

To use the Globus security infrastructure you must have your certificate in PEM format. Follow the instructions below if you need to change a P12-formatted certificate into a PEM-formatted certificate. You should then place the two files "usercert.pem" and "userkey.pem" into a ".globus" directory in your home area. The file permissions for the userkey file should be 0700 for the other 755 is appropriate.

Optionally, you may place your certificate and key in a non-standard location. In this case you must define the two environmental variables X509_USER_CERT and X509_USER_KEY to point to your certificate and key, respectively.

8.2.2 Installing Host Certificates

Host certificate/key pairs should be installed into the directory `/etc/grid-security/`. The host key must be readable only by root (`chmod 0400 hostkey.pem`); the host certificate can be world readable (`chmod 0444 hostcert.pem`).

These certificates may be installed in non-standard locations by setting the values X509_GATEKEEPER_CERT and X509_GATEKEEPER_KEY to the fully-qualified location of the host certificate and key, respectively.

Changing Certificate Formats

Many of the certificate authorities deliver certificates through a web browser. To use these certificates with Globus, they must be exported from the browser and then reformatted for Globus. Exporting is browser-specific so you will need to follow the help provided with your browser. Once you have extracted the certificate you should have a file with a p12 extension. This is in the PKCS12 format; you will need to change this to PEM format. If the edg-utils package is installed on your machine, simply executing

[3] http://marianne.in2p3.fr/datagrid/ca/ca-table-ca.html

```
/opt/edg/bin/pkcs12-extract
```

will create appropriate certificate and key files and place them in the standard location. This is a convenience method for the following:

```
openssl pkcs12 -nocerts \
        -in cert.p12 \
        -out ~user/.globus/userkey.pem

openssl pkcs12 -clcerts -nokeys
        -in cert.p12
        -out ~user/.globus/usercert.pem
```

The first command gives you your private key; this file must be readable only by you. The second command gives your public certificate. The ” `user`” should be replaced by the path to your home area. The ”.globus” directory is standard place to put your certificates.

Popular browsers typically use certificates in PKCS12 format. Consequently you will need to modify the format of the PEM certificates used for Globus to use them within a browser. To change a certificate from PEM format into PKCS12 format (on a machine with edg-utils installed), just issue the following command:

```
/opt/edg/bin/grid-mk-pkcs12
```

Again, this is a convenience method for the following:

```
 openssl pkcs12 -export \
              -out file_name.p12 \
              -name "My certificate" \
              -inkey ~user/.globus/userkey.pem \
              -in ~user/.globus/usercert.pem
```

where file_name.p12 is the name of the PKCS12 certificate, and the ” `user`” in the last two lines should be replaced by the path to your home area. You must then import the certificate into your browser.

Updating Certificate Revocation Lists

Having current certificate revocation lists (CRLs) is an extremely important aspect of the security framework. These lists identify certificates which have been revoked because the user no longer uses them, or they have been compromised. The CRLs can be updated with the command `edg-fetch-crl`. There is an associated daemon (`edg-crl-upgraded`) which can be started automatically to retrieve the CRLs periodically. It can be manipulated like all SysV daemon scripts.

Note: if the CRLs are out-of-date, certificates from the associated CA will not be accepted.

Virtual Organizations

The current list of virtual organizations[4] can be found on the web. If you did not register with a virtual organization when you signed the EDG Usage Guidelines (or wish to change your VO membership), then you must contact the VO manager directly.

Note: With the Testbed 1 software, membership in more than one virtual organization is not supported. When grid mapfiles are generated the actual organization you will be associated with depends on the

[4] http://marianne.in2p3.fr/datagrid/vo/vo-table.html

order the virtual organizations are listed in a site's mkgridmap configuration file. There is no mechanism by which the user can indicate which virtual organization should be used.

If you really need different roles in the Testbed 1 context, you should request multiple certificates (with slightly different subject names) and register the different subject names with different virtual organizations.

VO-specific Software

Most of the virtual organization currently require that some VO-specific software is preinstalled at sites supporting that virtual organization. The list of VO-specific software is published into the information systems from the `/opt/edg/info/mds/etc/ldif/ce-static.ldif` file by setting one or more `RunTimeEnvironment` attributes.

The list of RPMs can be obtained from the edg repository.

Mapping Users to Local Accounts (grid-mapfile)

Grid users are given access to a site's resources based on a local unix account. The Globus system uses a grid mapfile to map a user's certificate subject into a local account. The grid mapfile is generated from information contained in various virtual organization (VO) membership lists and a local configuration file.

Individual Accounts The configuration file allows for three different strategies for creating the local user accounts, each with advantages and disadvantages. The first option is to create a unique local account for every grid user. This allows the environment for each user to be specifically tailored for that user and allows detailed accounting of resource usage through standard mechanisms. The disadvantage is that this involves a lot of maintainance by the system administrator and may involve a large number of accounts being created.

Shared Account The second option is for all members of a particular virtual organization to be mapped into a shared account. Administratively this is the easiest solution as it usually involves only setting up one account per virtual organization. However, all detailed accounting information is lost, detailed access control is more difficult, and there are possible resource conflicts between multiple users at the same site.

Pooled Accounts The third option is creating pooled accounts. It is similar to the last option but instead pools of identical accounts are created and at any given time only one user (identified by subject name) is using one account. For example, for the Atlas VO a site may create a pool of accounts atlas001, atlas002, etc. This has the advantages that the accounts are easier to maintain and allow detailed accounting. However, there is a need to specify a policy for local resources when a given user stops using a pooled account. (E.g. how long local files are maintained, will the user get the same account when she/he returns, etc.)

Configuration In addition to the configuration of the mkgridmap script (described below), the accounts and a `gridmapdir` must setup.

To create a pool of accounts, you must setup individual unix user accounts whose names have a common prefix and a numeric suffix. For example, "atlas001", "atlas002", etc. To map users into this pool the prefix must be specified preceeded with a dot, i.e. ".atlas".

In addition, a `gridmapdir` must be created; it's default location is `/etc/grid-security/gridmapdir`, but may be set to a different location in the `globus.conf` file. An empty file must exist in the `gridmapdir` for each pooled account; the name of the file must match exactly the account name including both prefix and numeric suffix.

The mapping between a subject name and an individual account is based on the time stamps of the account entries in the `gridmapdir` and additional files named according to the URL-encoded subject names of the users.

Note: this mapping is fixed until the subject name entry is deleted. Currently this isn't done automatically and if the account pool is exhausted, users will get the same error as if they were not authorized to use the resource.

One important aspect for using pooled accounts is that the grid-mapfile and the `/etc/grid-security/gridmapdir` directory must be shared between all of the nodes in a site. If this is not done, then it is possible that the mapping will be done inconsistently depending on how a given machine is accessed.

Generating Mapfile with mkgridmap The `mkgridmap` script generates a gridmap file based on user information in the LDAP servers of various virtual organizations.

The behaviour of the script can be highly customized via a configuration file which is located in `/opt/edg/etc/mkgridmap.conf`. In its simpliest form, it simply lists the appropriate virtual organizations, the accounts to map these users to, and an `auth` directive to check that the users have signed the EDG Usage Guidelines.

The following example file (appropriate for a computing element) maps users from the specified virtual organizations to pooled accounts with the given prefix.

```
group ldap://grid-vo.nikhef.nl/ou=testbed1,o=alice,dc=eu-datagrid,dc=org .alice
group ldap://grid-vo.nikhef.nl/ou=testbed1,o=atlas,dc=eu-datagrid,dc=org .atlas
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=cms,dc=eu-datagrid,dc=org .cms
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=lhcb,dc=eu-datagrid,dc=org .lhcb
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=biomedical,dc=eu-datagrid,dc=org .biome
group ldap://grid-vo.nikhef.nl/ou=tb1users,o=earthob,dc=eu-datagrid,dc=org .eo

group ldap://marianne.in2p3.fr/ou=ITeam,o=testbed,dc=eu-datagrid,dc=org .iteam
group ldap://marianne.in2p3.fr/ou=wp6,o=testbed,dc=eu-datagrid,dc=org .wpsix

auth ldap://marianne.in2p3.fr/ou=People,o=testbed,dc=eu-datagrid,dc=org
```

This also checks the generated list of users against those who have signed the EDG Usage Guidelines. An example appropriate for a resource broker

```
group ldap://marianne.in2p3.fr/ou=guidelines,o=testbed,dc=eu-datagrid,dc=org dguser
auth ldap://marianne.in2p3.fr/ou=People,o=testbed,dc=eu-datagrid,dc=org
```

checks only the group of users who have signed the EDG Usage Guidelines and maps them into the user which runs the broker daemons.

Firewalls, Ports, and Site Security

Table 8.1 lists those ports used by various parts of the testbed software. Temporary ports used by Globus can be restricted to a particular range. Nearly all services can be configured to run on non-standard ports, if necessary.

There is at least one additional port needed for a two-phase commit job submission. This port has not yet been identified; in the meantime, opening all ports above 1024 will work.

Table 8.1: Ports Used by Various Services

Port	Service
80	HTTP server for Network Monitoring
123	Network Time Protocol
2119	Globus Gatekeeper
2135	MDS info port
2169	FTree info port
2170	Information Index
2171	FTree info port
2811	GSI ftp server
3147	RFIO
7771	Resource Broker
7846	Logging & Bookkeeping
8080	Tomcat Server (R-GMA, SpitFire)
8881	Job Sub. Service (client)
9991	Job Sub. Service (server)

GSI and Kerberos (AFS)

The client and server programs `gsiklog` and `gsiklogd` allow you to obtain an AFS token by presenting a Grid proxy rather than a Kerberos password.

This software has been produced by Doug Engert[5] of Argonne, with some testing and bug fixes by Helmut Heller[6] and Andrea Parrini[7].

The source code is available from Argonne National Laboratory[8] and we have produced Linux RPM's built with the Testbed 1 Globus2.0 distribution available from the EDG software repository[9].

For the client, installation from RPM is very straightforward, with no post-install configuration if the machine is already running as an AFS client. (`gsiklog` uses the existing AFS configuration files of the afsd cache daemon.)

Once configured, AFS tokens can be acquired in a `gsiklogd`-enabled cell by simply using the grid-proxy-init and then `gsiklog` commands. (`gsiklog -help` lists additional options, including specifying the remote AFS username and remote cell.)

The `gsiklogd` daemon can most easily be installed on an existing AFS authentification server, as it needs access to the Kerberos key `/usr/afs/etc/KeyFile` for its cell.

It must also be provided with a Grid key and certificate pair in `/etc/grid-security` called `afskey.pem` and `afscert.pem`, and the distinguished name must end CN=afs/CELL where CELL is the AFS cell name.

Finally, a file `/etc/grid-security/afsgrid-mapfile` must exist, with the same format as a gatekeeper grid-mapfile, but specifying local AFS usernames rather than unix usernames.

The daemon supports and SysV interface and can be started, stopped, and set to autostart in the customary way (see 3.2).

[5] mailto:deengert@anl.gov

[6] Helmut.Heller@lrz-muenchen.de

[7] mailto:aprarini@tiscalinet.it

[8] ftp://achilles.ctd.anl.gov/pub/DEE/

[9] http://datagrid.in2p3.fr/distribution/globus/gsiklog/

Table 8.2: Spec File Changes

	architecture	Directory Name
Linux	linux	LINUX
Solaris 2.6, gcc	solaris-2.6-gcc	SUN4SOL2
SGI Irix 6.x, cc	irix-6-cc	SGI64

8.3 Monitoring

8.3.1 Network Monitoring

8.3.2 Application Monitoring with GRM/PROVE

For more information on GRM see GRM - Grid Application Monitor Users Manual. For more information on PROVE, see PROVE-Visualisation tool for Grid Applications.

For linux, installing the RPM does all necessary configuration.

For other operating systems, the following must be done. Replace the terms 'linux' and 'LINUX' in the "grm.spec" and "prove.spec" files with appropriate terms from 8.2. The 'linux' term signifies the architecture. 'LINUX' is the name of the (sub)directory that will contain the binary files for linux.

A configuration file with the same name ('linux.def') should be present in the conf/ directory of the source. In the conf/ file there are also configuration files for the irix and solaris operating systems as examples (irix-6-cc.def and solaris-2.6-gcc.def).

8.4 Installing Application Software

The software specific to various applications is available from the EDG package repository[10]. You should install all of the application software necessary to support the users authorized to use your site.

When installing application software be sure that you update the RunTimeEnvironment flags in `/opt/edg/info/mds/etc/ldi` and restart the information systems. This will publish via the information systems that you have installed the given set of software.

8.5 Ftree and Globus MDS Information Services and Information Providers

8.5.1 Introduction

In order for the Job broker to find resouces on which to run a job, and storage elements on which to store data, an information provider needs to be setup. The GIIS, or Grid Index Information Service, is the type of information provider used to locate resources in Testbed-1. The GIIS is based on LDAP.

MDS is the LDAP based information provider which is part of Globus. WP3 has written it's own LDAP type of information provider, where the backend is cached in memory. This was written because performance tests with Globus MDS indicated that the performance of MDS was not adequate. The LDAP based information provider provided by WP3 is called ftree, and it is integrated with OpenLDAP2, not with MDS.

[10] http://datagrid.in2p3.fr/pkgs/raw/applications/index.html

WP3 has also delivered schema files, which define the information to be displayed by both ftree and MDS. The same schema files and information providers are used by both MDS and ftree. This will allow comparative tests between ftree and MDS to take place, while providing the same information.

For further information on LDAP and MDS deployment, along with a description of the schema files see 'MDS Deployment - Testbed 1.'

These documents are available in the documentation directory on the WP6 website for testbed-1 under documentation.

The Configuration instructions for the LDAP based information providers which follow apply regardless of whether Globus MDS or ftree is used. In some places they are slightly different, and this will be indicated.

8.5.2 Installation and Configuration

The following instructions explain how to configure the information providers. The installation procedure is largely carried out by installing the appropriate RPMs for the type of machine (site cache/GIIS, SE (storage element) CE (computing element) or netmon (network monitor)). In addition to the Globus RPMs, an RPM needs to be installed for the ftree information service. An RPM, edg-info-main-*.rpm is provided to help configure the information providers and three RPMs are provided to install the information provider scripts, the ones to install are dependent on machine type. Following the installation of the RPMs copy `/etc/edg/info-mds.conf.in` to `/etc/edg/info-mds.conf`. Edit `info-mds.conf`, the variables prefixed with a hash (#) must be edited and the hash removed.

8.6 Base Installation - all machines (site cache/GIIS, SE and CE)

Install

```
openldap-ftree-*.rpm
edginfo-main-*.rpm
```

Common settings for all configurations:

For all configuration, set the values in `/etc/globus.conf` to

```
GRIDMAP=/etc/grid-security/grid-mapfile
GATE_KEEPER_PORT=2119
GLOBUS_LOCATION=/opt/globus/
#GRID_INFO_USER= - This should NOT be root, this should be set to a non privileged user
GRID_INFO_GRIS=yes
GRID_INFO_EDG=yes
```

and in `/etc/edg/info-mds.conf` to

```
WP3_DEPLOY=/opt/edg/info/mds - The directory in which WP3
                               software is installed. If it is installed using
                               the RPMs this does not need to be changed
FTREE_INFO_PORT=2171 - The port number for the ftree information server
FTREE_DEBUG_LEVEL=0 - The debug level for ftree,
                      useful settings are 255 and 256
SITE_DN=Mds-Vo-name=local,o=grid - This should not contain
                    any spaces and should end in o=grid, if
                    left blank it will default to the hosts domain
                    components, dc=...,dc= For use with MDS2 use
                    Mds-Vo-name=local,o=grid
```

Site cache/GIIS Installation

Set the following variables within `/etc/globus.conf`

```
#GRID_INFO_GIIS_1=ral - The site name
#GRID_INFO_REG_GIIS=uk - The country
#GRID_INFO__REG_HOST=gppmds.gridpp.rl.ac.uk - The country host
```

Set the following variables within `/etc/edg/info-mds.conf`

```
SITE_INFO=yes
NETMON_PRESENT=no
CE_PRESENT=no
SE_PRESENT=no
#SITE_NAME=RAL - The site name
#SITE_INSTALLATION_DATE=20011115123410Z - This is in the format yyyymmddhhmmssZ
SITE_SYSADMIN_CONTACT=grid.sysadmin@hostname
SITE_USER_SUPPORT_CONTACT=grid.support@hostname
SITE_SECURITY_CONTACT=grid.security@hostname
SITE_DATAGRID_VERSION=1
#SITE_SE_HOSTS=gppse01.gridpp.rl.ac.uk,gppse02.gridpp.rl.ac.uk -
               This is a comma separated list with no
               spaces of the host names of the SEs
#SITE_CE_HOSTS=gppa.gridpp.rl.ac.uk - This is a comma separated
               list with no spaces of the host names of
               the CEs
#SITE_NETMON_HOST=gppnet.gridpp.rl.ac.uk - This is host name
               of the network monitor information provider
```

Network Monitor Information Provider Installation

Install

```
edg-info-netmon-*.i386.rpm
```

Set the following variables within `/etc/globus.conf`

```
#GRID_INFO_GIIS_1=netmon - The GIIS name
#GRID_INFO_REG_GIIS=ral - The site name
#GRID_INFO__REG_HOST=gppmds.gridpp.rl.ac.uk - The site host
```

Set the following variables within `/etc/edg/info-mds.conf`

```
SITE_INFO=no
NETMON_PRESENT=yes
CE_PRESENT=no
SE_PRESENT=no
#NETMON_PINGER_HOST=network.rl.ac.uk - This is the machine on
                    which the edg-pinger-*.i386.rpm is installed
```

Storage Element Information Provider Installation

Install

```
edg-info-se-*.i386.rpm
perl-Filesys-DiskFree-*.rpm
```

Set the following variables within /etc/globus.conf

```
#GRID_INFO_GIIS_1=se - The GIIS name
#GRID_INFO_REG_GIIS=ral - The site name
#GRID_INFO__REG_HOST=gppmds.gridpp.rl.ac.uk - The site host
```

Set the following variables within /etc/edg/info-mds.conf

```
SITE_INFO=no
NETMON_PRESENT=no
CE_PRESENT=no
SE_PRESENT=yes
#SE_ID=gppse01.gridpp.rl.ac.uk - This may be set manually, if left
       blank it will default to the local hostname
#SE_SIZE=500 - The size of the storage element in MB
SE_CONTACT=grid.support@hostname
SE_TYPE=disk
#SE_FILESYSTEMS=/dev/hda2,/dev/hda4 - This is a comma separated
                list with no spaces, these values are
                used with df to obtain the free space of the SE
#SE_CLOSE_CE=gppa.gridpp.rl.ac.uk - This is a comma separated
             list with no spaces, the values are the host
             names of the close computing elements
SE_PROTOCOLS=gridftp,rfio,file - This is a comma separated
             list with no spaces, the values are the protocols
             supported by the storage element
SE_PROTOCOL_PORTS=2811,3147, - This is a comma separated list
                  with no spaces, these values must relate
                  to the corresponding SE_PROTOCOLS
```

Computing Element Information Provider Installation

Install

```
CEInformationProviders-*.i386.rpm
```

Set the following variables within /etc/globus.conf

```
GRID_INFO_GIIS_1=ce - The GIIS name
GRID_INFO_REG_GIIS=ral - The site name
GRID_INFO__REG_HOST=gppmds.gridpp.rl.ac.uk - The site host
```

Set the following variables within /etc/edg/info-mds.conf

```
SITE_INFO=no
NETMON_PRESENT=no
SE_PRESENT=no
CE_PRESENT=yes
#CE_HOST=gppa.gridpp.rl.ac.uk - This may be set manually,
         if left blank it will default to the local hostname
#CE_BATCHSYSTEM=pbs - Supported systems are pbs and lsf,
                bqs will be added shortly
#CE_CLUSTER_BATCH_SYSTEM_BIN_PATH=/usr/pbs/bin -
              This is the path to the directory containing the
              queue management commands
#CE_QUEUE=short,long - This is a comma separated list
          with no spaces of the queue names of the computing
          element
#CE_CLOSE_SE_ID=gppse01.gridpp.rl.ac.uk, gppse02.gridpp.rl.ac.uk,gppse03.gridpp.rl.ac.uk
               - This is a comma separated list with no spaces of the
                 names of close storage elements
#CE_CLOSE_SE_MOUNT_POINT=usr/atlas,,usr/cms -
               This is a comma separated list with no spaces of the
               mount points of Close Storage Elements,
               these values must relate to the corresponding CLOSE_SE_ID's
```

The `CEInformationProviders-*.rpm` also installs the file `/opt/edg/info/mds/etc/ldif/ce-static.ldif.in`. This file has to be copied to `/opt/edg/info/mds/etc/ldif/ce-static.ldif`, the contents need to be changed to reflect the computing element environment. If required each queue can be customised using individualised static ldif files. If ce-static-queuename.ldif exists then this will be used in place of the ce-static.ldif.

Starting the server

The servers can be started and stopped via SysV scripts named `edginfo-mds` and `globus-mds` and can be set to autostart with the `chkconfig` command (see see 3.2).

Setting up a virtual organisation or country information service.

Install the `edg-info-main` RPM. There are 3 files of interest

```
etc/info-vo.conf
etc/rc.d/init.d/edginfo-vo
opt/edg/info/mds/etc/testbed1-vo.ldif
```

The only file that should need editing is the `testbed1-vo.ldif` this needs to contain entries for the sites within the vo/country an example entry is given for RAL

The ftree vo/country server is now ready to roll: /etc/rc.d/init.d/edginfo-vo start .

Setup for multiple site servers

There is a requirement for some sites to have more than one server running. Where a site has two or more sets of resources that are used by different VOs, a server will have to be run for each VO. If one or more VO shares the resources then only one server is required. Hence the need to have a .conf separate from `globus.conf`. A copy of the info.conf will be required for each server, as will copies of the edginfo and the contents of the `/opt/edg/info/mds` directory.

To set up another server a copies of the wp3-testbed-mds directory (e.g. wp3-testbed-mds-atlas), the edginfo (e.g. edginfo-atlas), and the info.conf (e.g. info-atlas.conf) file will have to be made.

The value for the WP3_DEPLOY in info-atlas.conf will have to be set to wp3-testbed-mds-atlas; the value for INFO_CONFIG in edginfo will have to be set to info-atlas.conf.

Finally, any other site- or VO-specific values will also have to be set in info-atlas.conf.

8.7 R-GMA

The Relational database based information provider has also been written by WP3. This is known as R-GMA, short for Relational Grid Monitoring Architecture. It is again possible to display information using the same schema as that displayed using LDAP and MDS. Again, the performance of this is being tested for comparison with the LDAP and MDS approach.

For further information on R-GMA see 'R-GMA Relational Information Monitoring and Management System User Guide'

8.7.1 Installation and configuration of R-GMA - Initial Setup

The R-GMA package consists of seven RPMs and depends on a number of external packages. These components can be obtained from the package repository[11]. Each RPM is described below.

- `rgma-api-java-*.rpm` contains the full set of Java APIs that application programmers can use to access the services offered by R-GMA. This also includes the APIs that R-GMA components use internally.
- `rgma-api-cpp-*.rpm` contains the corresponding C++ API. The internally used APIs are not yet included.
- `rgma-servlets-*.rpm` contains the six servlets that make up the core architecture of R-GMA.
- `rgma-tools-*.rpm` The RegistryServlet and SchemaServlet use a database to persistently store information about producers and the type of information they produce. This RPM contains the scripts to initialize these databases.
- `rgma-doc-*.rpm` contains the documentation. This includes the R-GMA guide, the architecture document, html-documentation of all the APIs (Java and C++) and the demos and sensors.
- `rgma-sensors-*.rpm` This currently contains only one sensor which should be run for each resource that runs a GRIS to republish the information into R-GMA. This is currently done by polling a GRIS and producing a table with the obtained information. In the future we envisage to publish directly information using the primary information providers.
- `rgma-demo-*.rpm` This contains two demos showing how to use the R-GMA components.

The external packages on which R-GMA depends are listed below. They can also be found from the above mentioned website.

- Java: You should use jdk1.3 from Sun or IBM. Both have been used to test and develop R-GMA. You should ensure that java is included in your path, for example if you have installed the edg java rpm then your path needs to include `/usr/java/jdk1.3.1_01/bin`.

[11] http://marianne.in2p3.fr/datagrid/testbed1/repositories/pkg-repository.html

- MySQL: This only applies to you if you want to run your own RegistryServlet and SchemaServlet. In this case you must install and run MySQL. The root user should not have a password so that the scripts in the tools rpm work. If you need tighter security you need to edit those scripts. In the future this aspect of the installation will be improved.

- Tomcat: You will need to run version 4.0. Since the Spitfire component of the WP2 software also needs tomcat, you will have this anyway. However you need to edit the server.xml file in order to tell tomcat from where to load the servlets that make up the R-GMA core. Look for the set of lines that read:

  ```
  <!-- Tomcat Root Context -->
  <!--
      <Context path="" docBase="ROOT" debug="0"/>
  -->
  ```

  below that add the following set of lines:

  ```
  <!-- R-GMA servlets directory -->

  <Context path="/R-GMA/ProducerServlet"
   docBase="/opt/edg/info/servlets/ProducerServlet" debug="0" reloadable="true"/>
  <Context path="/R-GMA/DBProducerServlet"
   docBase="/opt/edg/info/servlets/DBProducerServlet" debug="0" reloadable="true"/>
  <Context path="/R-GMA/ConsumerServlet"
   docBase="/opt/edg/info/servlets/ConsumerServlet" debug="0" reloadable="true"/>
  <Context path="/R-GMA/ArchiverServlet"
  docBase="/opt/edg/info/servlets/ArchiverServlet" debug="0" reloadable="true"/>
  </programlisting>
  To run the demo or when you need to run a RegistryServlet and a
  SchemaServlet for a VO also add:
  <programlisting>
  <Context path="/R-GMA/SchemaServlet"
   docBase="/opt/edg/info/servlets/SchemaServlet" debug="0" reloadable="true"/>
  <Context path="/R-GMA/RegistryServlet"
   docBase="/opt/edg/info/servlets/RegistryServlet" debug="0" reloadable="true"/>
  ```

  The first part ensures that applications can use the produce- (persistently when using the Database-Producer) consume- and archive- services of the respective servlets. See the next section for more informtion. To be able to stream information from producers to consumers the attribute allowChunking of the Connector has to be set to false as in the following:

  ```
  <Connector className="org.apache.catalina.connector.http.HttpConnector"
  port="8080" minProcessors="5" maxProcessors="75" allowChunking="false"
  enableLookups="true" redirectPort="8443" acceptCount="10" debug="0"
  connectionTimeout="60000"/>
  ```

- Ant: Ant is needed to build the demos and the sensors. Make sure ant is in your PATH. For example if you have installed the edg ant rpm then your path needs to include `/usr/bin`.

- libwww: In order to use the C++ API you will need the libwww libraries. Install the rpm w3c-libwww.rpm.

- R-GMA: The R-GMA software is found under `/opt/edg/info`. There exists a file called `release-setup.sh` that sets a number of environment variables. If you use the external packages as provided on the

WP6 website, you should not need to edit this file. Otherwise you need to customize it. Once release-setup.sh is customized it needs to be executed before one can run the demo. Move the `log4j.props`, file into the directory specified by the variable `home` in `release-setup.sh` and customize it if you want to see debug messages telling you what is going on.

8.7.2 Configuring R-GMA for a Virtual Organization

To set up R-GMA for a virtual organization one has to run one RegistryServlet and one SchemaServlet. These make use of a database to store information about producers (RegistryServlet) and tables (SchemaServlet). These Servlets will initially be run at RAL and the URLs of those Servlets will be made available elsewhere. In the future each virtual organisation will run their own RegistryServlet and SchemaServlet.

To be able to produce information one has to run at least one ProducerServlet, however many producers can use the same ProducerServlet to publish data for them. In the same way one has to run at least one ConsumerServlet to be able to consume data that has been published by a producer. For every servlet the web.xml file describing the web-application has to be configured and for Consumer, Producer, DataBase-Produer and Archiver a properties file has to be configured. Each API class needs to know the location of the respective servlet which services it. The properties files are located in `$RGMA_HOME` and currently have to be copied into the home directory of the user running the application code that uses the API class in order to be found. The scripts to run the demos and the sensors do this automatically for you. If you are running Tomcat on the same machine as the application code that uses the producer/consumer/archiver APIs the default values for the ServletLocations suffice. The idea behind this setup is, that one can run a sensor that publishes information on each node of a cluster and have one ProducerServlet running on a head node to handle all the requests from consumers.

Each servlet has a number of init parameters that are set at the beginning of the servlet life cycle and are now discussed in turn. The sections about the SchemaServlet and RegistryServlet and Tools are only relevant if you need to build your own VO. The web.xml files have to be configured before Tomcat is started up since they are read at startup time only.

ProducerServlet

`registryServletLocation` is the URL of the RegistryServlet. The ProducerServlet has to be able to contact the Registry to register Producers.

DBProducerServlet

`registryServletLocation` is the URL of the RegistryServlet. The DBProducerServlet has to be able to contact the Registry to register DataBaseProducers.

ConsumerServlet

`registryServletLocation` is the URL of the RegistryServlet. The ConsumerServlet has to be able to contact the Registry to find out about Producers.

SchemaServlet

`schemaDatabaseLocation` is a JDBC URL for the location of the Schema database, see the documentation of your database for more information. The default setting is for a mysql database running on localhost. It probably makes sense to run the database on the same host as the SchemaServlet, but it is not mandatory.

`schemaDatabaseUserName` is the database user name of the schema database. The default is schema.

`schemaDatabasePassword` is the clear text password for the above user. The default is info.

RegistryServlet

`registryDatabaseLocation` is a JDBC URL for the location of the Registry database, see the documentation of your database for more information. The default setting is for a mysql database running on localhost. It probably makes sense to run the database on the same host as the RegistryServlet, but it is not mandatory.

`registryDatabaseUserName` is the database user name of the registry database. The default is registry.

`registryDatabasePassword` is the clear text password for the above user. The default is info.

`schemaServletLocation` is the URL of the SchemaServlet. The default is to run the SchemaServlet on the same host as the RegistryServlet, in which case the same database can hold both the registry and schema database.

Tools

To populate the Schema database with a set of known tables and to bring the registry database into a clean state with no registered producers the build file in `$RGMA_HOME/tools/dbases` has to be run. Since soft-state registration is currently not yet implemented the Registry can get into an inconsistent state. The administration of the registry database will be moved into the RegistryServlet in a future release.

Demos

The `/opt/edg/info/demo` directory contains two demos to illustrate the use of R-GMA. To run the demos just install the RPMs and don't configure the servlets or properties files. Also include the RegistryServlet and SchemaServlet in the server.xml file. Run a MySQL server with no password for the root user. Starting up tomcat makes sure all these services are available. In order to run the demo you need to change the permissions on the files named run in SimpleDemo and SimpleDemo/etc to be executable. E.g.

```
 find . -name run -exec chmod 0755
{} \;
```

Each subdirectory of the demo directory contains a script called run that takes the name of the respective demo `SimpleDemo` or `ClusterLoad` as an argument. The `README` in each subdirectory explains briefly what is happening.

8.7.3 Installation and configuration of R-GMA - Avaliable Sensors

In this section the available sensors that have been implemented using the R-GMA approach are discussed, with the emphasis upon how the sensors are used.

MDS Producer Sensor

The purpose of the MDS Producer sensor is to publish all the information available from a Globus GRIS server or in fact from any LDAP server into the R-GMA and to permit a consumer to access this information via the normal R-GMA approach. Each Site that runs a GRIS/LDAP server should run a MDSProducer.

The Globus GRIS server publishes information about the status of the Grid and its components, such as available CPU nodes, available service types and the status of batch queues. The server is implemented using the LDAP protocol, with the Grid information stored in a hierarchical LDAP directory structure. Each piece of information is associated with an attribute, with the permitted attributes being defined and grouped by an LDAP schema or 'object class'. The context of the information is given by its position within the directory structure.

There are currently 6 schema defined in the Globus2 release:

```
globusBenchmarkInformation
globusNetworkInterface
globusQueue
globusServiceJobManager
globusSoftware
grADSoftware
```

Furthermore EDG publishes information according to a number of objectclasses which are republished into the following set of R-GMA tables:

```
NetMonHostLoss
NetMonHostRTT
NetMonHostThroughput
NetMonLossPacketSize
NetMonRTTPacketSize
NetMonThroughputBufferSize
NetMonTooliperfER
NetMonToolpingER
SiteInfo
StorageElement
StorageElementProtocol
StorageElementStatus
```

For the Globus GRIS here is exactly one table in R-GMA for each of the objectclasses. Since each schema consists of a number of attributes, these attributes form the column names of the relational table. An additional column is added to each table, giving the LDAP distinguished name (DN), or the context, of the entry. The way the EDG LDAP schemas are used is more complicated especially for the networking information, but there is a correspondence between a table and a certain combination of Objectclasses. R-GMA cannot currently republish information about the FileElement Objectclass because this information is not permanently held in the LDAP server but dynamically created requiring the knowledge of a local filename. The MDSProducer is completely generic and only assumes knowledge about the names of the objectclasses.

The MDS Producer is implemented in JAVA, in the class `MDSProducer`. The class is supplied with a properties file which points it to a particular LDAP server, and contains a list of table names and corresponding search filters to publish. The properties file, `MDSProducer.props`, consists of 5 properties in the standard JAVA key=value format with each property taking a new line:

- `schemaServletLocation` The MDSProducer needs to contact the SchemaServlet to inquire about the column names that correspond to attributes of entries in the DIT.
- `mdsBindHost` The host upon which the LDAP server is running.
- `mdsBindPort` The integer TCP port number on which the server is running.If not specified the default of 2171 is used.
- `mdsBindDn` The base DN from which to start the LDAP search.
- `tableNames` A comma separated list of the SQL table names in which to publish the information. The order must correspond to the order given in the searchFilter property.

- `searchFilters` A comma separated list of LDAP search filters. Each search filter picks out particular entries in the DIT. A subset of the attributes corresponds to column names of the corresponding table (maintained by the SchemaServle) to be published.

The MDS Producer will then request all entries of each of the specified search filters, starting the search at the given base DN. The information from each of the entries found is then published, along with the DN of the entry, in the appropriate table.

The MDS Producer is likely to run nearby the LDAP server that it is polling, possibly upon the same machine, although this does not have to be the case since the LDAP servers are polled using the standard LDAP wire protocol. Currently there will be one MDS Producer for every GRIS server. It would be easy to implement a system to provide aggregate information about one or more sites. This would involve a simple Consumer-Producer model where the Consumer side subscribes to all the site MDSProducers and then publishes the aggregate information in some suitable format.

The MDSProducer class includes a main() method that runs the pollGRIS method of the MDSProducer class in an infinite loop. The time between subsequent polls is given as a commandline argument in milliseconds. There is currently a bug, which we don't understand, when accessing the ComputingElement objectclass which prevents us from republishing this information.

Running the MDSProducer

We assume that a ProducerServlet is deployed and properly configured (registryServletLocation points to your VO's RegistryServlet), tomcat is up and running, the MDSProducer.props file points to an LDAP server that runs the EDG information provider scripts, the property schemaServletLocation points to the VO's SchemaServlet and the file" run" in `/opt/edg/info/sensors` is executable. Now run the command “run MDSProducer 10000” which starts the MDSProducer and polls the LDAP server every 10000 milliseconds.

8.8 GSI FTP Daemon

A GSI-enabled daemon must run on any node which needs to serve its local file system to remote users via GridFTP (i.e. via the client `globus_url_copy` which uses gridftp as the transport protocol). This includes the gatekeeper, resource broker, and storage element nodes.

8.8.1 Security

Incoming requests are authorized via the grid-mapfile mechanism. Consequently, machines running the ftp daemon must have a full security installation. That is the machine must have a host certificate and key installed, a grid-mapfile, and all of the security RPMs which contain the Certificate Authority certificates and Certificate Revocation List URLs installed. The daemons which update the grid-mapfile (23) and CRLs (8.2.2) should also be running.

8.8.2 Configuration

The FTP daemon is configured via the `/etc/globus.conf` file. 8.3 lists the relevant parameters, their default values, and their descriptions.

8.8.3 Control

This daemon is controlled via a standard init.d-style script which support the start, stop, restart, and status directives. (See 3.2 for more details.)

Table 8.3: FTP Daemon Parameters

Parameter	Default	Description
GLOBUS_LOCATION	/opt/globus	Installation root of Globus software.
GLOBUS_GSIWUFTPD_PORT	2811	Port to use for GSI-enabled FTP.
GLOBUS_GSIWUFTPD_LOG	/var/log/globus-gsi_wuftpd.log	Location of log file.
X509_GSIWUFTPD_CERT	/etc/grid-security/hostcert.pem	Location of host certificate.
X509_GSIWUFTPD_KEY	/etc/grid-security/hostkey.pem	Location of host key.
GRID_GSIWUFTPD_USER	root	User to run FTP daemon.
GLOBUS_GSIWUFTPD_OPTIONS	unspecified	Additional FTP daemon options.
GLOBUS_TCP_PORT_RANGE	unspecified	Range of TCP ports (e.g. ”30000,31000”)
GLOBUS_UDP_PORT_RANGE	unspecified	Range of UDP ports (as above)