NCSR "DEMOKRITOS"

Institute of Nuclear Physics

UNIVERSITY OF ATHENS

Department of Informatics
& Telecommunications

crossGrid

Cluster Installation Guide in CrossGrid

LCFGng for Dummies

Cluster Installation in CrossGrid

LCFGng for Dummies   (v1.0)

Vangelis Floros
University of Athens – Dept. of Informatics & Telecommunications
(floros@di.uoa.gr)

Christos Markou
NCSR "Demokritos"
(cmarkou@inp.demokritos.gr)

Nikos Mastroyiannopoulos
NCSR "Demokritos"
(nmastro@inp.demokritos.gr)

Table of Contents

Chapter

1

Introduction

T he purpose of this guide is to provide a systematic set of instructions on the procedure of setting up a Grid cluster using the LCFGng system. Initially developed in the University of Edinburgh, LCFGng has been adopted by many European Grid projects to facilitate the installation and administration of large-scale inter-cluster Testbeds.

The clusters build around the Globus[1] Toolkit which currently is the de facto standard software infrastructure for providing Grid services. There are also numerous of other software packages working together, all installed and administered using LCFGng. Finally the OS platform that is used is Red Hat Linux 7.3, on Intel x86 architecture

Who should read this guide

This document is targeted towards Grid Cluster Administrators, who want to set up a cluster from scratch, without prior knowledge of Grid concepts and principles. This guide is the accumulation of a year of experience setting-up, upgrading and administering LCFG controlled Grid cluster installations, in the context of "CrossGrid"[2] IST Project.

At the present time, there are numerous help documents on the Net (primarily in European DataGrid's and LCG project's web sites), describing the process. The problem with these documents is that they don't cover all issues, require a good background knowledge regarding Linux and cluster administration, and, even worse, they sometimes contain erroneous and outdated material. Thus our effort was twofold; on one hand to provide an integrated view of all this material, filtering out what works and what doesn't, and on the other hand to document our own experience and the problems that emerged during the process.

---

[1] http://www.globus.org

[2] http://www.crossgrid.org/

We must emphasize that the cluster to be set up is not a stand-alone site, rather a part of a larger inter-cluster testbed. This means that there are also several nodes that perform shared services, which are not site-specific, but, rather, shared by the testbed as a whole. This document is not going to get into the information, which has to do with these shared services.

The Software

This updated version of the guide marks the transition of Crossgrid's testbed from EDG 1.4.X to LCG-1 grid software bundle. This is a major upgrade since there are lot's of differences between these to releases. LCG-1 is based on LCFGng and Red Hat Linux 7.3 whereas EDG 1.4.X uses the old LCFG on RH 6.2 systems. Thus a major impact of this transition is that all clusters have to be setup again from scratch. Although at first this sounds like a great effort (and to a point it is) LCG-1 proves to be much more stable and straightforward to install.

Currently the main repository for Crossgrid software, which also mirrors EDG and LCG software, is Gridportal (http://gridportal.fzk.de) hosted at FZK in Karlsruhe, Germany. The intended cluster administrator should be able to find most of the software packages referenced in this guide there.

Other Sources

This document primarily is based on two documents: "LCG1 Installation notes" [5] and "EDG LCFGng Server Installation using RedHat 7.3" [6]. Moreover configuration details for joining the CrossGrid testbed are available from LIP's site hosted in Lisboa, Portugal, at http://www.lip.pt/computing/projects/crossgrid/crossgrid-pro. For any other questions concerning LCG-1 and CrossGrid software you should use the CrossGrid helpdesk facility at http://cg1.ific.uv.es/hlpdesk/.

Acknowledgments

The authors would like to thank all the CrossGrid participants for they comments and support in preparing this document. Especially we would like to thank, Ariel Garcia, Marcus Hardt, Rafael Marco, Jorge Gomes, Mario David, Jan Astalos and Jesus Marko.

The Players

For setting up a minimum Grid cluster one needs at least 5 PCs. Without getting into many details, a typical site consists of the following types of nodes:

Computing Element

This is the main entry to a local site. It hosts the Globus Gatekeeper Grid service, which together with the Worker Node provides computational resources to the user. This node handles the interaction with the rest of the Grid environment

accepting jobs, dispatching them for execution and finally returning the output. This provides a uniform interface to the computational resources it manages.

Note: Formally a CE is defined to be the ensemble of a Gatekeeper, one or more Worker Nodes and a batch system. In this guide, we use the term CE to declare only the node running the gatekeeper.

Worker Node

These nodes lay behind a Gatekeeper (Computing Element) and are managed via a local batch system (namely PBS the Portable Batch System). User-defined computational tasks are actually performed on these nodes. The end-user software must, therefore, be installed on them. These nodes do not run any Globus daemons, but do have a few Globus APIs[1] for accessing grid services and information. A cluster will typically employ many WNs, in contrast to the other node roles where typically only one PC per role is required.

Storage Element

A Storage Element provides uniform access to large storage spaces. The storage element may control large disk arrays, mass storage systems and the like. This element hides the details of the backend storage systems and provides a uniform interface to the Grid user.

User Interface

This node provides necessary client software, which allows the end-user to utilize the CrossGrid testbed. Users log into this machine to query for Grid resources, submit jobs and retrieve their output, stage data files to remote storage areas and generally utilize the available testbed services.

LCFGng server

This is the host where the cluster is orchestrated by using the LCFGng[2] suite, together with various other tools and services like NFS[3] and DHCP[4]. The whole installation of the cluster begins on this server. The set up of the software has to be done manually and all the information that the other nodes require to configure themselves is retrieved by this server. Moreover, this is the machine that has to be kept safe and "clean", throughout the whole procedure and after completing setting up the site, and that is why some security issues have to be taken into account.

Pre-Installation Issues

To start a site within the Grid concept, the minimum hardware requirements are:

---

[1] Application Programming Interface

[2] http://www.lcfg.org

[3] Network File System

[4] Dynamic Host Configuration Protocol

- 5 PCs (one for each role described above) – preferably with high speed CPU (> 1.5GHz), typical disk space (> 40 Gbytes) and fair amount of main memory (at least 512 MB).

- 2 monitors – one is for the LCFG server and one for testing the installation in each node, although after setting up the site, one needs none.

- 1 keyboard and 1 mouse

When you are selecting PCs for an LCG/EDG cluster consider the following:

o Select Fast Ethernet network cards with proven Linux compatibility. This is a most important issue. You have to make sure with the vendor, that the network cards shipped within the PCs provide 100% support for the Linux kernel included in Red Hat 7.3. Also it is useful that the network cards support the PXE protocol which, as we will discuss later, let's you boot and install a new PC without a boot-disk but instead fetching everything from the local network.

o Do not install a demanding video adapter (8 Mb of display memory is more than enough), at most you will have to use a text console to audit a node's behavior.

o A floppy disk drive is required in every host if you are going to install them using LCFGng boot disk.

o Don't spend money on DVD-ROM drivers, CD Recorders and things like that. They are useless for GRID clusters. It's better to install some extra memory instead.

Network card MAC Address

Before starting the whole procedure, you have to get the MAC[1] addresses of the Ethernet cards. These addresses are going to be needed later on, when installing the DHCP service. Usually, the MAC addresses are written onto the cards or can be found in some feature fields of the BIOS of the system.

Network Access

The machines, which are going to be used in the site, have to have network access. A lot of the software that is going to be used will be downloaded from the Web or via FTP. Of course, there must be network connection between the nodes.

Naming

One should decide on the names of the nodes. This is a crucial first step since a lot depend on that. There are three obvious ways of naming the machines. One is to

---

[1] Media Access Control Address is based on IEEE 802 standards

name each node by using its own functionality incorporated in the name, e.g. for worker node *wnode* or *worknode* or *workern* etc. The other convention is to just give every node a number naming e.g. *n001*, *n002* or *cg001*, *cg002* etc. The third option, and the one that we used, is to give a specific name to the LCFGng server, e.g. *lcfg* and then name the other nodes of the cluster using an increasing number sequence, e.g. cg001, cg002, etc. This has the advantage of making software changes to all the nodes easier by writing less software commands and with less probability of making mistakes during configuration. In addition, if you want, after the cluster installation, to write shell scripts to automate some processes, you will appreciate the uniform naming schema of your nodes.

IPs and DNS

After dealing with the naming schema of your machines, you must reserve a number of IP addresses for them. Usually, this is something that has to be decided in cooperation with the network administrator of your organization. It is important to have in mind the possible growth of the number of nodes in the cluster in the future. Before start doing anything, it is better to ensure that the names and relative IPs of your hosts have been properly declared in your site's DNS[1] server table information and have been publicly distributed. The *nslookup* program can be handy for that purpose.

In addition, it is advisable when asking for IP addresses to try to get a contiguous block of addresses, especially if you plan to add more nodes in the future. This will make the firewall configuration easier. If there is shortage of available IPs in a site (something very common nowadays), it is possible to have the WNs without direct network connectivity by utilizing NAT[2].

The following table summarizes the information you have to have ready in hand before starting the setup of the cluster: If you are not certain about any of the values you have to consult your local network administrator.

Description	Example
IP address of Gateway	195.134.67.1
IP addresses of one ore more DNS servers	195.134.65.3
Network mask	255.255.255.0
Network broadcast address	195.134.67.0

---

[1] Domain Name System

[2] Network Address Translation (NAT) is a technique used to map a set of private IP addresses within an internal network to another set of external IP addresses on a public network.

NTP time server hostname	ntp1.sp.se
Hostname and IP address of LCFGng server	cglcfg, 195.134.67.170
Hostname and IP addresses of LCFG of cluster nodes	cgnode00, cgnode01, … 195.134.67.171, 195.134.67.172, …
MAC addresses of cluster nodes	00:80:48:15:86:91, 00:80:4b:12:32:10, …

Chapter

2

LCFGng Server Installation

S etting up an LCFGng is a time-consuming process, comprised by many steps which have to be followed carefully to avoid any problems. The process has proven to be trouble free more or less but there is always a chance that you miss something or some sort of incompatibility may arise unexpectedly to annoy the administrators. Keep in mind that, most of the time, the installation and administration of one's cluster will be performed through the LCFGng server, so one has to make sure that everything works properly in this machine.

The objective of the procedure described is to have at the end a machine running Red Hat Linux 7.3 (with the required OS updates) having installed the server side of LCFGng, a DHCP service, an Apache web server, NFS, and loaded with the necessary LCG-1/Crossgrid packages that will be installed in the cluster nodes.

Installing Red Hat Linux 7.3

Red Hat 7.3 comprises of three CDs. In case there is no CD available, one has to get access to the version of Red Hat from a web server which mirrors Red Hat Linux distributions[1]. If you follow the instructions only CDs 1 and 2 will be needed.

System boot

Start the Red Hat 7.3 installation by inserting the first CD in the drive. The installation uses a graphical installer named Anaconda When the system boots the user is presented with various installation choices. Select *Custom install* in order to have full control over the packages that will be installed.

---

[1] For mirror sites check http://www.redhat.com/download/mirror.html

Hard Disk Partitions

The first decision one has to make is the partitioning of the hard disk. We prefer to keep things simple here and create three Linux native partitions and a Linux swap partition. It is advisable to keep the Linux swap partition as large as the physical memory of the system. The other three partitions are:

/boot, which does not have to be more than 32Mb. 16MB would do fine for most cases[1]. This is the directory where the kernel is stored, and where the boot loader seeks for it when the system boots up. It must be the first partition in the system.

/opt, which is the partition where all the Grid software will be stored, and from where it will be retrieved later on, for the installation on the cluster nodes. This has to be as large as possible. We suggest reserving 70% of one's total hard disk space for this.

/, which is the root partition and its size is of no real interest, though it is preferable not to be small. We suggest something between 5GB and 30% of your hard disk's space. This is where the local applications are stored, so some future requirements should be taken into consideration.

Of course, you have the freedom to create as many partitions as you like, with variable sizes, but the above configuration is usually sufficient for most of the cases.

Network Configuration

Here you have to fill the configuration of the network IPs, that is:

IP Address:

Netmask:

Network:

Broadcast:

Hostname:

Gateway:

Primary DNS:

Secondary DNS:

The *IP address* is the address of the LCFGng server on the network and *hostname* is of course the name you have chosen for your host. All this information have been gathered during pre-installation..

---

[1] Resent Linux releases like RH9.0 require a minimum of 75MB for /boot. RH 7.3 does not pose such requirement

Time Zone selection

You must tick the box *system clock uses UTC* and choose your time zone. In the case of NCSR "Demokritos" that is Europe/Athens. The system clock must be in UTC time because that is what LCFGng system uses. Grid software (especially Globus) requires all node clocks to be synchronized. By using UTC for the hardware clock and an NTP server[1], as we will see in a later paragraph, one may have machines synchronized, irrespective of their world location. If in a site there is no NTP server configured, you can use one of the publicly available servers around the world[2].

Account configuration

Choose a root password. For security reasons do not create any other accounts, and of course don't let people use the machine for other purposes.

Firewall Setup

Enable firewall at Medium Security. Select eth0 as trusted device and enable DHCP, SSH and WWW. This way you will block all the traffic from TCP ports 0-1023 except those used by the above applications. Note that the above settings provide a basic level of security which unfortunately is not enough and has to be improved later on according to local site network security policies.

Package Selection

Select the following package groups:

1. Printer support
2. Classic X Window System
3. X Window System
4. GNOME
5. Network Support
6. Messaging and Web Tools
7. NFS File Server
8. Web Server
9. Utilities
10. Software Development

In addition to the above package groups you have to include also the RPM of the DHCP server. To do so click on *Select individual packages* navigate into the System Environment/Daemons category and select dhcp.

---

[1] It is a server which has access to time synchronized by atomic clocks and can distribute this info over the web, via a specific process

[2] For more information check: http://www.eecis.udel.edu/~mills/ntp/servers.html

The installation proceeds by coping and installing all the above selected packages. This will take a while depending on the machine's speed.

X Window System Configuration

Before the system is ready to boot for the first time the X Window System together with the monitor and the display card has to be configured. Make sure that you choose the appropriate drivers for the display hardware in order to be able to work in a graphical environment.

The system is then booted with a clean Linux installation. Then proceed as follows.

LCFGng and Services Setup

Set up SSH

SSH (Secure SHell) is a client/server application used for remote access of a system. Due to it's security characteristics is dominating over other similar applications like telnet or rlogin. The required SSH programs and libraries have already been installed in the LCFGng server. All you have to do is to make the ssh daemon start automatically whenever the system boots. To do this issue the following commands as root:

```
$ chkconfig --level 2345 sshd on
$ service sshd start
```

The ssh server daemon now is configured and running. You can use the ssh command to access the server from the net.

Set up NTP

As already discussed NTP is a protocol for time synchronization over the net. An NTP server has to be defined from which the LCFGng while acquire accurate time information. To do this you have to edit **/etc/ntp.conf** file. Find the following line and replace <YourNTPServer> with the hostname of a valid NTP server (e.g. ntp1.sp.se):

```
.
.
.
server  <YourNTPServer>
.
.
.
```

Now you have to setup the NTP client daemon to start automatically every time the system boots. First manually synchronize with the server:

```
$ /usr/sbin/ntpdate <YourNTPServer>
```

then issue:

```
$ chkconfig --level 2345 ntpd on
$ service ntpd start
```

The NTP client daemon will periodically connect with the NTP server, retrieve time information and adjust the system clock accordingly if needed.

Populate hosts.allow and hosts.deny files

To strengthen the security of the LCFGng server it is advisable to restrict access to local TCP services only to those machines that require them, such as the cluster nodes or your personal workstation. This can be done through two files namely **/etc/hosts.allow** and **/etc/hosts.deny**. File hosts.allow should contain all the machine IPs that can access the services. An example could be:

```
#
# hosts.allow   This file describes the names of the hosts which are
#           allowed to use the local INET services, as decided
#           by the '/usr/sbin/tcpd' server.
#
ALL: 195.134.67.
ALL: 195.134.66.
ALL: 195.134.65.
```

The above permits access to all TCP services from all machines that belong to .67, .66 and .65 subdomains. Individual IPs can be declared accordingly (eg. ALL: 195.134.69.118).

File hosts.deny is simpler since it just denies access to anyone else:

```
#
# hosts.deny   This file describes the names of the hosts which are
#           *not* allowed to use the local INET services, as decided
#           by the '/usr/sbin/tcpd' server.
#

ALL:ALL
```

Edit the two files accordingly.

Install edg-populate-serverng and updaterpms-static-server packages

From this step forward the server will start shaping up as an LCFGng server. There are two packages that have to be installed:

- updaterpms-static-server
- edg-populate-serverng

You can find the packages at EDGs site in IN2P3, France. The URLs are respectively:

- http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/packages/external/RPMS/RH73, and
- http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/packages/autobuild/i386-rh7.3/RPMS

If you follow to the above links you will be navigated into two directories which contain lots of different RPMs. You have to find and download the latest version for the above files. The moment we are writing this guide the latest versions are:

- **1.0.2-1** for updaterpms-static-server, and
- **1.0.23-edg1** for edg-populate-serverng

Download the files in root's directory and install them using the rpm tool:

```
$ rpm -ivh updaterpms-static-server-1.0.2-1.i386.rpm
$ rpm -ivh edg-populate-serverng-1.0.23-edg1.noarch.rpm
```

Download the machine profiles and RPM lists

The heart of the LCFGng systems are two set of files: The first set called **rpmlists** defines which packages have to be installed in an LCFGng client machine according to the role that this machine will have in the cluster (e.g. CE, SE etc). The second set called **source** files contains configuration directives for the software that is installed in each machine, including the operating system. Each machine controlled by LCFGng has a **profile** file, which includes the required source files according to the specific software that is installed in the particular node. Again the role of the machine implicitly sways the contents of a profile.

Both sets are consisted of numerous files which have already been prepared by the LCG, EDG and CrossGrid development teams. They are provided through LCG's and CrossGrid's CVS systems. Each release of LCG software bundle is characterized by a version number, also known as a tag in CVS jargon. For instance LCG-1 version 1.1.3 is tagged LCG1-1_1_3. CVS uses tags to distinguish files between different versions. Without getting deep into the details of the CVS system, the steps one can follow to retrieve the initial rpmlists and source files are as follows:

The files will be downloaded under /opt/local/linux/7.3 directory.

```
$ mkdir /opt/local/linux/7.3/cvs
$ cd /opt/local/linux/7.3/cvs
$ export CVS_RSH=`which ssh`
$ export CVSROOT=:ext:anoncvs@gridportal.fzk.de:/data/cvs/crossgrid
$ cvs -z3 co crossgrid/wp4/sites
```

The above will retrieve the latest tag of LCG-1, EDG and CrossGrid configuration files. Moreover it also retrieves configuration files for all CrossGrid sites (at least of those who use CVS for version control). If you change to crossgrid/wp4/sites you will see the following subdirectories:

```
$ cd crossgrid/wp4/sites/
$ ls -al
total 76
drwxr-xr-x   19 root    root        4096 Dec 15 15:09 .
drwxr-xr-x   4 root     root        4096 Dec 15 15:09 ..
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 auth
drwxr-xr-x   6 root     root        4096 Dec 15 15:09 common
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 csic
drwxr-xr-x   2 root     root        4096 Dec 15 15:09 CVS
drwxr-xr-x   6 root     root        4096 Dec 15 15:09 cyfronet
drwxr-xr-x   8 root     root        4096 Dec 15 15:09 demo
drwxr-xr-x   8 root     root        4096 Dec 15 15:09 fzk
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 icm
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 ins
drwxr-xr-x   5 root     root        4096 Dec 15 15:09 lip
drwxr-xr-x   5 root     root        4096 Dec 15 15:09 psnc
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 sas
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 tcd
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 uab
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 ucy
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 usc
drwxr-xr-x   3 root     root        4096 Dec 15 15:09 uva
```

There is one sub-directory for each site. If you belong to the CrossGrid testbed probably your site is already listed. It's a good idea to use it to keep control of you own configuration files. If you are not familiar with CVS you should check GridPortal's site at http://gridportal.fzk.de for general instructions.

From the above directories the one which is of general interest is the sub-directory named **common**:

```
$ cd common
$ ls -al
total 24
drwxr-xr-x    6 root     root        4096 Dec 15 15:09 .
drwxr-xr-x   19 root     root        4096 Dec 15 15:09 ..
drwxr-xr-x    8 root     root        4096 Dec 15 15:09 cg
drwxr-xr-x    2 root     root        4096 Dec 15 15:09 CVS
drwxr-xr-x   10 root     root        4096 Dec 15 15:09 edg
drwxr-xr-x    7 root     root        4096 Dec 15 15:09 lcg1
```

We will refer to this directory from now on as the **common directory**. In the common directory you can find rpmlists and source files for the LCG-1 and EDG software bundles (under the directories **lcg1** and **edg** respectively), and customized rpmlists and source files of CrossGrid that extend LCG-1 configuration files (under **cg** directory).

Directory lcg1 has the following tree structure:

```
lcg1/
|-- BDII
|-- rpmlist
|-- source
`-- updaterep
```

The machine profiles are in the lcg1/source subdirectory, the RPM lists in the lcg1/rpmlist subdirectory. Directory lcg1/updaterep contains some useful scripts. You can ignore BDII if you are installing only the four core machine roles (CE, SE, WN and UI)

Similar is the structure of cg directory:

```
cg
|-- ng_rpmlist
|-- ng_source
|-- rpmlist
|-- source
`-- updaterep
```

From the above you can ignore rmplist and source directory since they refer to the previous version of LCFG. The directories ng_rpmlist and ng_source are the LCFGng specific ones. In this step we will need the rpmlist templates that reside under cg/ng_rpmlist/templates directory. Copy them to /opt/local/linux/7.3/rpmcfg:

```
$ cp cg/ng_rpmlist/templates/* /opt/local/linux/7.3/rpmcfg
```

All these files have a .template filename suffix. You should rename them accordingly and omit this suffix e.g.

```
$ cd /opt/local/linux/7.3/rpmcfg
$ mv LCFGng-rpm.template LCFGng-rpm
...
```

Edit LCFG-rpm and set the LCG1_RPMDIR and CG_RPMDIR macros as follows:

```
#define LCG1_RPMDIR \
</opt/local/linux/7.3/cvs/crossgrid/wp4/sites/common/lcg1/rpmlist
#define CG_RPMDIR \
</opt/local/linux/7.3/cvs/crossgrid/wp4/sites/common/cg/ng_rpmlist
#define LOCAL_RPMDIR </opt/local/linux/7.3/rpmcfg
```

Normally you shouldn't have to bother much with the other templates although there are some exceptions. We will elaborate on that in the next chapter. For the time being rename them and leave them as is.

Finally make a symbolic link named /export pointing to /opt

```
$ ln –s  /opt /export
```

Populate the RPM repository with updateRPMcache

Two files downloaded in edg/updaterep and cg/updaterep directories respectively are of importance in this step: updateRPMcache.pl and updaterep.conf.

updateRPMcache.pl is a Perl script used to populate the RPM repository in the LCFGng server, downloading only those RPMs which are necessary to satisfy the machine configuration files (profiles) passed on the command line. The RPM repository is just a local directory in the server which resides under /opt/local/linux/7.3/RPMS.

The command options of the script are:

```
updateRPMCache.pl [--help] [--clean, --noclean] [--days=2.5] <mach.
config...>

This command updates the LCFG RPM cache using information in the given
machine configuration files.  The following options are supported:

--help      print out this usage statement

--clean     remove obsolete rpms older than given number of days
--noclean   print number of obsolete rpms but do not remove (default)

--days=2.0  limit in days for removing obsolete rpms (default=2)

--base=dir  directory to use for base of repository; default is the
            value in /etc/updaterep.conf.

--rpmcfg    treat the machine configuration file(s) as rpmcfg file(s)
rather
            than a node configuration file.

--conf=file use the given configuration file rather than the default
            /etc/updaterep.conf

Notes:
  1) The time-limit works from the last modification time.  This script
     will touch the needed rpms everytime it is run.

  2) Wildcarded rpm names will not be found in the repository.
```

Note: The updateRPMcache.pl script that LCG-1 currently distributes does not support all of the above options, and especially the –-rpmcfg option which is very important in order to download the RPMs required for the LCFGng server installation. That's why we recommend to use the latest version of the script distributed with EDG 2.X in common/edg/updaterep.

The URLs from which the script will download the files are listed in cg/updaterep/updaterep.conf. This file should be copied to directory /etc. :

```
$ cp cg/updaterep/updaterep.conf /etc/updaterep.conf
```

Now you are ready to download the RPMs needed to install the LCFGng server. cd to edg/updaterep directory and issue:

```
$ ./updateRPMCache.pl -rpmcfg \
/opt/local/linux/7.3/rpmcfg/ngserver73-rpm
$ ./updateRPMCache.pl --rpmcfg \
/opt/local/linux/7.3/rpmcfg/nginstallroot73-rpm
$ ./updateRPMCache.pl --rpmcfg \
/opt/local/linux/7.3/rpmcfg/LCFGng-rpm
```

As we will se in the later chapters updateRPMCache.pl will also be used for downloading the RPM packages for the client machines, further populating the repository.

Build nginstallroot file system for 7.3

The nginstallroot file system is the directory tree that will be copied in every client node when this will be configured for the first time. The LCFGng server hosts this directory tree and exports it via NFS to all clients. To build the installroot required by 7.3 installations run this script:

```
$ /opt/local/linux/utils/create-nginstallroot-rh73
```

Please note that during rpm installation process you can see some error messages like "Cannot create temp file" or "execution of %post scriptlet failed", do not worry about them.

The nginstallroot is placed in /opt/local/linux/nginstallroot/7.3. Go to the /opt/local/linux/nginstallroot/7.3/etc and copy the file installparams.default to installparams

```
$ cd /opt/local/linux/nginstallroot/7.3/etc
$ cp installparams.default installparams
```

The default installparams file assumes a US keyboard and the CET time zone. You have to edit it if it doesn't apply to your case

Run lcfgng_server_update

To make sure that all needed LCFGng rpms are installed on your server, you should use the lcfgng_server_update.pl script, also located in common/lcg1/updaterep. This script will report which rpms are missing or have the wrong version and will create a shell script which you can then use to fix the server configuration. Run it in the following way:

```
$ ./lcfgng_server_update.pl \
/opt/local/linux/7.3/cvs/crossgrid/wp4/sites/common/\
lcg1/rpmlist/lcfgng-common-rpm.h \
/tmp/lcfgng_server_update_script.sh
$ /tmp/lcfgng_server_update_script.sh
```

once more

```
$ ./lcfgng_server_update.pl \
/opt/local/linux/7.3/cvs/crossgrid/wp4/sites/common/\
lcg1/rpmlist/lcfgng-server-rpm.h \
/tmp/lcfgng_server_update_script.sh
$ /tmp/lcfgng_server_update_script.sh
```

and again

```
$ ./lcfgng_server_update.pl \
/opt/local/linux/7.3/cvs/crossgrid/wp4/sites/common/\
cg/ng_rpmlist/lcfgng-server-CG-rpm.h \
/tmp/lcfgng_server_update_script.sh
```

Setup the DHCP server

Dynamic Host Configuration Protocol (DHCP) is the protocol used, among other things, to provide network nodes with an IP address during their installation. Thus a DHCP server daemon has to be setup in the LCFGng server. The daemon is configured through the /etc/dhcpd.conf file. You can find an example of this file in /etc/dhcpd.conf.ngexample. Below is an example from UoA site:

```
# /etc/dhcpd.conf for dhcpd 2.0

# distribute an IP address only if the nodes is known
deny unknown-clients;
# the server will not reply to the unknown clients; in this way
# it is possible to have a second DHCP server
allow booting;
allow bootp;
not authoritative;
option domain-name "di.uoa.gr";

# These 3 lines are needed for the installation via PXE
option dhcp-class-identifier "PXEClient";
filename "pxelinux.0";

subnet 195.134.67.0 netmask 255.255.255.0 {

    option routers 195.134.67.1;
    option domain-name-servers 195.134.65.3;

    host cgnode00 {
        hardware ethernet 00:80:48:15:35:44;
        fixed-address 195.134.67.171;
        option user-class "http://cglcfg.di.uoa.gr/profiles/";
    }

    host cgnode01 {
        hardware ethernet 00:80:48:15:35:a7;
        fixed-address 195.134.67.172;
        option user-class "http://cglcfg.di.uoa.gr/profiles/";
    }

    host cgnode02 {
```

```
        hardware ethernet 00:80:48:15:26:ce;
        fixed-address 195.134.67.173;
        option user-class "http://cglcfg.di.uoa.gr/profiles/";
    }

    host cgnode03 {
        hardware ethernet 00:80:48:15:15:41;
        fixed-address 195.134.67.174;
        option user-class "http://cglcfg.di.uoa.gr/profiles/";
    }

    host cgnode04 {
        hardware ethernet 00:80:48:45:85:b9;
        fixed-address 195.134.67.175;
        option user-class "http://cglcfg.di.uoa.gr/profiles/";
    }
} # end subnet 195.134.67.0
```

Update this file as needed by your local network configuration adding all your nodes. For each node you have to first to define the hostname after the host keyword (e.g. cgnode00). The **option user-class** contains the URL of the LCFGng server. Change it to your server's one. The **hardware ethernet** is the MAC address of the client node. In the **fixed-address** line enter the IP address that you want this client node to be assigned.

Note for CrossGrid sites upgrading from EDG 1.4.X. You can use the /etc/dhcpd.conf from your old LCFG server but not as it is. Consult the example and make the appropriate modifications.

Finally start and configure DHCP daemon to load automatically every time the system boots:

```
$ service dhcpd start
$ chkconfig --level 2345 dhcpd on
```

Setup the NFS server

As already mentioned the LCFGng server exports the nginstallroot tree to all the client nodes using the NFS protocol. Also the repository under /opt/local/linux/7.3 containing all the RPMs is exported in order for the clients to find and install the packages they need.

For NFS configuration cp the file /etc/exports.ngexample73 to /etc/exports

```
$ cp /etc/exports.ngexample73 /etc/exports
```

Make sure that the file contains at least the following lines

```
/opt/local/linux/7.3                    *(ro,no_root_squash)
/opt/local/linux/nginstallroot/7.3  *(ro,no_root_squash)
```

Finally start and configure the NFS daemons (two actually) to load automatically every time the system boots:

```
/sbin/service nfs start
/sbin/service nfslock start
/sbin/chkconfig --level 345 nfs on
/sbin/chkconfig --level 345 nfslock on
```

Set up HTTP server

A Web server has to be setup in the LCFGng server since LCFGng uses the HTTP protocol to propagate profile information to the client nodes. The Apache web server has already been installed in the system. All you have to do is to configure it to be used in the LCFGng context.

Configuration information for the Apache HTTP daemon (httpd) is kept in file `/etc/httpd/conf/httpd.conf`. In the same directory you can find a file named `/etc/httpd/conf/httpd.conf.ngexamples73` which has already been prepared with most of the required settings. So the first step is to copy this file to the working one:

```
$ cd /etc/httpd/conf
$ cp httpd.conf.ngexamples73 httpd.conf
```

For security reasons you have to restrict access to the Web Server only to those hosts that are considered trusted. The restrictions that have been applied with hosts.allow and hosts.deny are not sufficient since Apache has its own access policies. Edit httpd.conf and change lines 495 and 760 from "Allow all" to "Allow <YourDomain>" where <YourDomain> your domain is the your local network domain name. This restricts access of web folders /var/obj/conf/server/web and /var/obj/conf/server/web/install only to hosts belonging to this domain. For example in our case the file looks like:

```
.
.
<Directory "/var/obj/conf/server/web">
.
.
    Order allow,deny
    Allow from di.uoa.gr
</Directory>
.
.
<Directory "/var/obj/conf/server/web/install/">
      AllowOverride None
#     Options None
      Options ExecCGI
      Order allow,deny
      Allow from di.uoa.gr
</Directory>
<Files "install.cgi">
      AuthUserFile /etc/httpd/.htpasswd
      AuthType BASIC
```

```
        AuthName "User and Password needed"
        require user lcfgng
</Files>
.
.
```

Notice the commands included between <Files..> and </Files> tags. These instruct the httpd daemon that install.cgi should be accessed and executed only by user **lcfgng**. This is not an Operating System user but a Web Server specific user whose login and password (in an encrypted form) is listed in file /etc/httpd/.htpasswd. In order to create the password file you need to issue the following command:

```
$ htpasswd -c /etc/httpd/.htpasswd lcfgng
```

The program will ask for a password for the newly created user. The script install.cgi as we will later see is used to define over the Web, which cluster nodes should be installed using PXE.

Finally start and enable the web server:

```
/sbin/service httpd start
/sbin/chkconfig --level 345 httpd on
```

Completing installation

Prepare Installation Boot disk

If you are going to use boot disk for LCFGng client installation you have to download the appropriate boot image and create the disk. First download the image file:

```
$ wget http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-
lcfg/packages/external/diskettes/RH73/bootdisk_rh73_04122002.img
```

Then insert a new diskette in the floppy drive and issue the command:

```
$ dd if=bootdisk_rh73_22112002.img of=/dev/fd0 bs=1024
```

Prepare PXE

New nodes can be installed without a boot disk using PXE (Pre-Execution Environment). Note that in order to be able to take advantage of this option the network card of the node has to support PXE. The installation is done with **pxelinux** an implementation of the PXE specification for Linux. The first step is to put the pxelinux loader (provided by the package syslinux, already installed with installpackages-ngserver73) in the /tftpboot directory :

```
$ cp /usr/lib/syslinux/pxelinux.0 /tftpboot/
```

TFTP is the protocol used by PXE capable Network Cards to download the
pxelinux.0 loader and then by this loader to download its configuration file
(from /tftpboot/pxelinux.cfg) and a kernel (from /tftpboot/kernel). To
enable it modify /etc/xinetd.d/tftp by changing the "disable" parameter to
"no" and reload the xinetd configuration:

```
$ vi /etc/xinetd.d/tftp
$ /sbin/service xinetd restart
```

Chapter

3

Cluster definition with LCFGng

LCFGng is designed to handle automated installation and configuration in a very diverse and changing cluster environment. The LCFG server will keep centrally all the information for cluster description, the profiles of client computers and the control of installation and update of required software.

LCFGng Architecture

In this section we will briefly present the inner details of the LCFGng system. Figure 1 shows the overall architecture of the system:
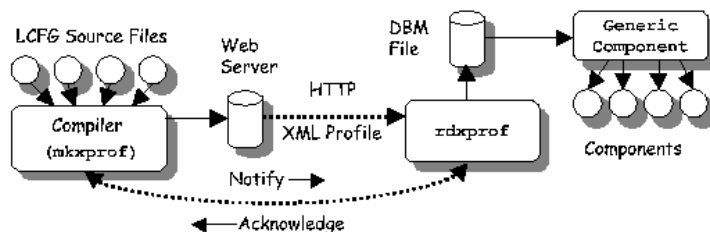

Figure 1 - LCFG architecture

The notion of rpmlists and source files has already discussed in the previous chapter when we retrieved them from the LCG-1 CVS. The source files are compiled by a tool named mkxprof (make xml profile) which produces for each client node an XML file with all the profile details that were defined in the source files. These XML files (one for each node in the cluster) are published in the local Apache Web Server in a well known web directory where the nodes can find and retrieve them.

Note: The tool mkxprof comes in two flavors: as a daemon service (mkxprofd) and as a standalone program (mkxprof). The daemon mkxprof searches for source profiles in a standard directory in the server, namely /var/obj/conf/server/source. Every time that is reloaded it looks in the above directory for node profiles and

compiles them all. By using the standalone mkxprof you can compile source profiles anywhere on the disk, by passing the full-path name of one or more profiles as a parameter to the program. It is advisable that these two approaches are not combined together. Thus you should decide which solution (daemon or standalone) suits you better and stick to it.

A daemon process named rdxprof (read xml profile) runs in every client node and periodically checks for profile updates from the LCFGng web server. If an updated profile is available rdxprof fetches it over HTTP and calls the LCFGng components that handle the details of applying the new profile changes in the machine.

The LCFGng components are Perl scripts that reside under /etc/obj directory in each client node. There exists one component for each software package or operating system facility that is centrally controlled by LCFGng. For instance there is an *auth* component that handles local users and groups, an *nfs* component that handles nfs mounts, a *globus* component that handles Globus configuration etc. Profile updates can be done also manually by running rdxprof by hand in a client node and then calling the appropriate component. This is especially useful as we will see when new software packages have to be installed or updated.

Finally client nodes acknowledge the update of the profile back to the LCFGng server, when the update process has completed.

RPM Lists

RPM lists are files that list the rpm packages that have to be installed in client nodes. The files have already been downloaded and copied to directory /opt/local/linux/7.3/rpmcfg during the setup of LCFGng server. If you take a look at the contents of the directory you will see numerous lists, one for each role a machine can have in a LCG-1/CrossGrid cluster.:

```
$ cd /opt/local/linux/7.3/rpmcfg
$ ll
total 68
-rw-r--r--    1 root     root          327 Jan 16 11:29 BDII-rpm
-rw-r--r--    1 root     root          616 Jan 16 13:38 CE-rpm
-rw-r--r--    1 root     root          465 Jan 16 14:28 cfgdir-rpm.h
-rw-r--r--    1 root     root          463 Jan 16 11:29 DW-rpm
-rw-r--r--    1 root     root          339 Jan 16 11:29 IC-rpm
-rw-r--r--    1 root     root         1266 Jan 15 12:17 LCFGng-rpm
-rw-r--r--    1 root     root          264 Jan 15 12:17 local-rpm.h
-rw-r--r--    1 root     root          324 Jan 16 11:29 MDS-rpm
-rw-r--r--    1 root     root          336 Jan 16 11:29 MON-rpm
-rw-r--r--    1 root     root          336 Jan 16 11:29 NM-rpm
-rw-r--r--    1 root     root          324 Jan 16 11:29 PX-rpm
-rw-r--r--    1 root     root          333 Jan 16 11:29 RB-rpm
-rw-r--r--    1 root     root          343 Jan 16 11:29 RLS-rpm
-rw-r--r--    1 root     root          363 Jan 16 11:29 SECLASSIC-rpm
-rw-r--r--    1 root     root          456 Jan 16 11:29 UI-rpm
-rw-r--r--    1 root     root          327 Jan 16 11:29 VOMS-rpm
-rw-r--r--    1 root     root          329 Jan 16 11:29 WN-rpm
```

CE-rpm lists packages for the Computing Element, SE-rpm for the Storage element etc. If you look inside the rpmlists we will notice that they are C-header-like files with similar command syntax.

Each LCG-1/CrossGrid version has its own set of rpm lists. This means that every time you upgrade to a new release a new set of rpm lists have to be downloaded. We will come back to cluster upgrade in Chapter 5. The good news is that these lists are already prepared each time by the LCG-1 development team and the CrossGrid integration team. There are two files that you may have to edit yourself, namely `cfgdir-rpm.h` and `local-rpm.h`.

cfg-rpm.h

File cfgdir-rpm.h contains the definition of the paths where LCG1 rpmlists and CG rpmlists respectively are located in the file system:

```
/*
 * Define the directory from which all XXX-rpm files will be read.
 *
 * Please note the strange way quotes are used... only the opening quote has
 * to be put here, the closing one goes in the file where the variable is used.
 */

#define LCG1_RPMDIR </export/local/linux/7.3/cvs/crossgrid/wp4/sites/common/lcg1/rpmlist
#define CG_RPMDIR </export/local/linux/7.3/cvs/crossgrid/wp4/sites/common/cg/ng_rpmlist
#define LOCAL_RPMDIR </export/local/linux/7.3/rpmcfg
```

LOCAL_RPMDIR macro points to the directory where cfgdir-rpm.h and the rest of the local files are located. Change these definitions only if you have downloaded and installed these files in different directories. Otherwise leave the default values.

local-rpm.h

File local-rpm.h is used for definition of any other of packages that you would like to install and are not included in the LCG-1/CrossGrid bundle, or are included and you don't want them to be installed. One such package is the package of the local CA which is specific for each site. For instance in Greece the local CA is HellasGridCA operated by Aristotle University of Thessaloniki. It is up to the administators to contact with the appropriate CA and get the RPM packages with local security settings. In our case the package is named ca_HellasGrid-local-0.1-1.noarch.rpm. The rpm has to be copied in directory /opt/local/linux/7.3/RPMS/security and the following line has to be added in local-rpm.h:

```
ca_HellasGrid-local-0.1-1
```

Administrators in other locations will have to find the appropriate CA get in contact with the responsible and get the appropriate package for their needs.

Reversely if you don't want to trust a specific CA and thus disallow access to people with credentials signed by it you may add the following line in local-rpm.h:

```
-ca_HellasGrid-0.19-1
```

This will uninstall the ca_HellasGrid rpms from the cluster nodes.

Source Files

An initial set of LCG-1 specific source files has already been downloaded in directory common/lcg1/source during the LCFGng server installation. Note that source files differentiate not only between LCG-1 releases but also between different Grid Testbeds. This means that when the LCG-1 bundle is adopted in a different Testbed context (like CrossGrid) lots of things have to be modified. There are also good news here since the Test & Validation team of CrossGrid is making the appropriate modifications to support the required Virtual Organizations (VOs) and applications that are being developed by CrossGrid's various Work Packages. Thus you can find the CrossGrid specific source files under common/cg/ng_source. Still there are a few settings that are unique among sites and have to be fixed by local administrators before the setup of the cluster.

To aid the configuration of a new cluster a number of "template" files has been prepared and have been placed in common/cg/ng_source/templates directory. These files are:

- local-cfg.h.template

- cfgdir-cfg.h.template

- private-cfg.h.template

- site-cfg.h.template

- redhat73-local-cfg.h.template

- ClusterWorkerNode-cfg.h.template

- Users-UI-cfg.h.template

- Template node profiles in CE.template, WN.template, UI.template and WN.template.

At this point you should decide where your local source files would reside. If you follow the daemon mkxprof approach copy all template files to /var/obj/conf/server/source. Go to the common directory and issue:

```
$ cp cg/ng_source/template/* /var/obj/conf/server/source
```

Answer yes to the questions weather to overwrite existing files after the second command. Then go to local sources directory and rename the files by omitting the .template suffix:

```
$ cd /var/obj/conf/server/source
$ mv cfgdir-cfg.h.template cfgdir-cfg.h
```

```
$ mv ClusterWorkerNode-cfg.h.template ClusterWorkerNode-cfg.h
$ mv local-cfg.h.template local-cfg.h
$ mv private-cfg.h.template private-cfg.h
$ mv redhat73-local-cfg.h.template redhat73-local-cfg.h
$ mv site-cfg.h.template site-cfg.h
$ mv Users-UI-cfg.h.template Users-UI-cfg.h
```

Configuring cfgdir-cfg.h

Start by configuring file cfgdir-cfg.h. This contatins the paths where all the source files are spread in the LCFGng server. By default it contains the following:

```
#define LCG1_CFGDIR "/root/crossgrid/wp4/sites/common/lcg1/source
#define CG_CFGDIR "/root/crossgrid/wp4/sites/common/cg/ng_source
#define LOCAL_CFGDIR "./
```

LCG1_CFGDIR and CG_CFGDIR point to the directory where LCG-1 and CrossGrid source files respectively are kept. LOCAL_CFGDIR points to the directory of the local configuration files. Value "./ means that all files reside in the same (current directory). You can use this definition if you want for instance to keep separately the source profiles from the rest .h local source files.

Configuring site-cfg.h

File site-cfg.h contains site-wide specific information. This is the file that has to be extensively modified to your site's custom needs. The table below summarizes the lines that have to be changed. For example values check the values that already exist in the file.

Value	Explanation
`#define CE_HOSTNAME`	Hostname of the Computing Element
`#define SE_HOSTNAME`	Hostname of the Storage Element
`#define SITE_LOCALDOMAIN`	The local domain name of your site
`#define SITE_MAILROOT`	Email of the site administrator
`#define SITE_GATEWAYS`	Gateway hosts of your local network
`#define SITE_ALLOWD_NETWORKS`	IPs and IP subdomains that are allowed to access the services
`#define SITE_NAMESERVER`	Comma separated list of IPs of Site Nameservers

`#define SITE_NETMASK`	Netmask of your local network
`#define SITE_NETWORK`	IP address of local network
`#define SITE_BROADCAST`	Broadcast address of local network
`#define SITE_NTP_HOSTNAME`	Hostname of you NPT server
`#define SITE_TIMEZONE`	Local site time zone
`#define SITE_NAME_`	A name for your cluster. Choose something distinctive.
`#define SITE_EDG_VERSION`	Enter the LCG-1 tag currently installed in your site
`#define SITE_INSTALLATION_DATE_`	The Year, Month, Day, Hour, Minute and Second.that the site was installed in the format YYYYMMDDHHmmSSZ
`#define SITE_LCFG_SERVER`	The hostname of the LCFGng server
`#define SITE_WN_HOSTS`	The hostnames of cluster Worker Nodes in a wildcarded format. This is used by NFS for exporting directories between WNs and other nodes.
`#define SITE_GIIS`	A unique word that identifies your site. It is used by Testbed Information Indexed Servicies.
`#define CE_IP_*`	Various definitions for the characteristics of the CE machine. Edit accordingly to the best of your knowledge.

Keep in mind that the above modifications are the minimal that has to be made. If special configuration is applied in a site the administrator is responsible for finding the appropriate changes that have to be made.

Configuring private-cfg.h

There is only one line included in this file.

```
+auth.rootpwd y3SesdDaa5k
```

It defines in an encrypted form (using 3DES cryptographic algorithm) the root password for all cluster machines. You can produce your own password using perl and issuing the following command by replacing *PassWord* with the desired passphrase and `Sl` with any other two character string.

```
$ perl –e 'printf crypt("PassWord","Sl")."\n"'
```

The `Sl` is the so-called *salt* of encryption. Now the encrypted password that is produced can be inserted into the above line replacing the existing value.

Configuring redhat73-local-cfg.h

This file actually contains amendments to definitions included in file common/lcg1/source/redhat73-cfg.h. The former contains operating system specific definitions. It is general enough to support most common hardware configurations and basic Linux services but it is very likely that you will have to make proper modifications if you have special hardware not recognized during the setup of a machine, or you want an OS service to behave differently. These modifications should be inserted in redhat73-local-cfg.h.

One option you should confirm and modify accordingly is the module (driver) of the network card that is installed in the machines. It is defined in redhat73-local-cfg.h at line:

```
+hardware.mod_eth0              alias eth0 8139too
```

8139too is the module for network interfaces that are based on the Realtek chipset. Modify it according to what your network cards support. If not all machines contain the same cards you can insert the module that is supported by most of them, and specify a module per machine in the relevant profile as we will explain later on.

If you are not sure of what module corresponds to your card unfortunately there is no straightforward way to find it. You can search the Internet if you know brand and model information.

This file also contains two macro definitions namely LOCAL_RPMCFG and LCG1_RPMG. These are used by client nodes the first time they are installed and each time they are upgrading. You must change the LOCAL_RPMCFG variable to point to your site's rpmlist folder, and the LCG1_RPMCFG variable to point to the LCG rpmlist directory. Note that LCG1_RPMCFG must be a relative path from your LOCAL_RPMCFG directory. For instance if you have followed all the instructions till now the above macros should be set to the following:

```
#define LOCAL_RPMCFG /export/local/linux/7.3/rpmcfg
#define LCG1_RPMCFG ../cvs/crossgrid/wp4/sites/common/lcg1/rpmlist
```

Configuring the WN shared definitions

Since it is expected to have many Worker Nodes in a cluster it is convenient to keep most common settings in a single file which can be included by each WN profile. This is the purpose of ClusterWorkerNode-cfg.h file. This file has already been prepared for CrossGrid. Any definitions shared among all WNs can be included in this file.

Configuring UI users

Accounts for UI machine users are configured and controlled centrally in the LCFGng server using file Users-UI-cfg.h. You have to edit it to create the initial user accounts as follows:

First define a group and a groupID for the users :

```
EXTRA(auth.groups)                    xgrid
auth.groupgid_grid                    2010
```

xgrid is the group name and 2010 the ID. Make sure that the ID is unique among other UI groups.

Then provide a space separated list of usernames:

```
EXTRA(auth.users)         john mary
```

The above will create two users with login names john and mary respectively. Then enter details for each account:

```
+auth.userpwd_john               c3fx2l1I0qISY
+auth.usercomment_ john          John Smith
+auth.userhome_ john             /home/john
+auth.usergroup_ john            xgrid
+auth.useruid_ john              1501
+auth.usershell_ john            /bin/tcsh
```

Notice how the +auth definitions are explicitly related with user john. The password is given in an encrypted form, produced using the command

```
$ perl –e 'printf crypt("PassWord","Sl")."\n"'
```

as when defining the root password in private-cfg.h. At auth.usercomment provide the full name of the user. This is important and needed when the user will later on try to request a grid certificate from his account. Make sure that the auth.useruid is unique for every user.

When later you decide to create a new account for your UI node you have to edit again this file and add similar details for the new users.

Profile Configuration

Each node in the cluster has a separate profile file which contains configuration details for this node. The file is named after the hostname of the corresponding node. For example in our site we have the following profile files:

```
$ ll cg*
-rw-r--r--    1 root      root           682 Nov  4 15:25 cgnode00
-rw-r--r--    1 root      root           252 Nov  4 15:25 cgnode01
-rw-r--r--    1 root      root           524 Nov  4 15:25 cgnode02
-rw-r--r--    1 root      root           252 Nov  4 15:25 cgnode03
-rw-r--r--    1 root      root           786 Nov  4 15:25 cgnode04
```

cgnode00 corresponds to the Computing Element, cgnode01 to the SE, cgnode01 and cgnode03 are Worker Nodes, cgnode02 is the User Interface and cgnode04 is the Storage Element.

Template profiles have been prepared for these four basic roles (and also for any other role currently defined in CrossGrid), named as CE-template, SE-template, WN-template and UI-template. The first step is to rename each template profile to the corresponding hostname. e.g:

```
$ mv CE-template <YourCEhostname>
```

Where <YourCEhostname> is the hostname of your Computing Element etc..

Then apply the following changes:
- In all profiles change the definition **#define HOSTNAME <Hostname>** and replace <Hostname> with the hostname of the node.
- If you have more Worker Nodes or when you later want to add a new WN in the cluster copy an existing WN profile to the new name of the node and change the hostname in the #define HOSTNAME line.
- If a particular node has a different network card than the one declared in redhat73-local-cfg.h add the following line after the inclusion of this file:

```
/* Linux default resources */
#include LOCAL_CFGDIR/redhat73-local-cfg.h"
+hardware.mod_eth0                    alias eth0 <module>
```

   By replacing <module> with the appropriate network card module name

Profile compilation

You are ready know to compile the sources and produce the XML profiles. Start and enable mkxprof daemon on the LCFGng server with the command:

```
$ service mkxprofd start
$ chkconfig --level 345 mkxprofd on
```

This will run mkxprof as daemon but without building the profiles. For building the XML files profiles you have to reload it:

```
$ service mkxprofd reload
```

You have to run this command every time you change the configuration files and you want to deploy them to the clients. In LCFGng the result of the XML profiles creation is available from the LCFGng web server:

```
http://<LCFGServer>/status/
```

You should check this page for compilation errors of the profiles. If there are no errors you can proceed with the installation of the nodes otherwise check the compilation logs to find the cause of the problem and make the appropriate modifications in the source profile files.

The web page should look like this:
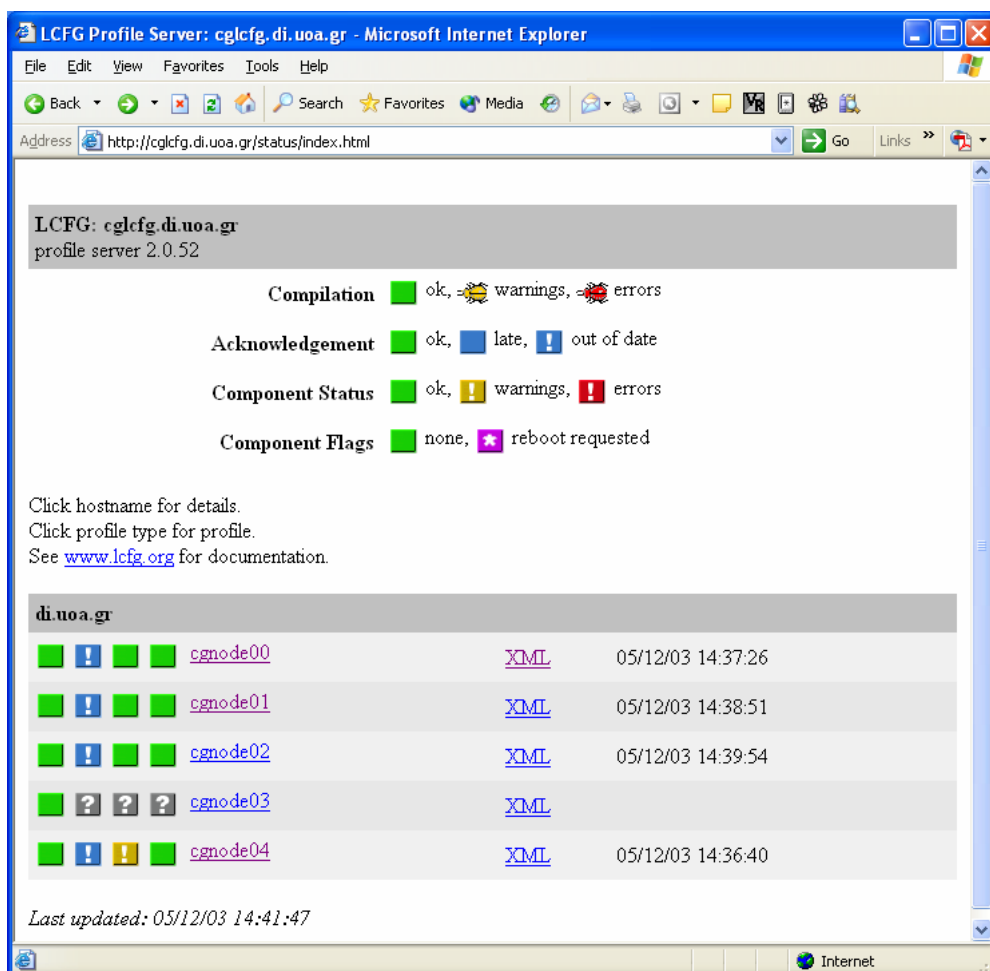
From here you have access to node status information, compilation error logs and the contents of compiled XML profiles.

The actual profile files in XML are stored in

```
/var/obj/conf/server/web/profiles/<domainname>/<hostname>.xml
```

Where domainname is the cluster network domain and hostname the name of each node.

Populating the RPM Repository

Once all the client profiles are compiled, you can populate the RPM repository with the client RPMs using updateRPMcache.pl.

To download the packages you run the script passing as parameter the node profile's full path. This will download packages required for a specific node. Wildcards may be used to get all packages for all nodes at once:

```
$ cd common/edg/updaterep
$ ./updateRPMcache.pl /var/obj/conf/server/source/grid*
```

In the above example client machines are called grid01, grid02 and grid04.

Note: It is important to stress that this procedure requires a couple of hours or more to complete. Approximately 2GB of files are going to be downloaded from GridPortal's LCG-1 mirror site and if your site doesn't have a very fast connection with FZK (where GridPortal is hosted) it will take more or less 4-5 hours to get all packages. So after running the above you may consider going out for a walk/coffee/lunch/sleep ☺

Chapter

4

Node Installation and Configuration

A fter successfully compiling the profiles and populating the repository you can finally move to the last part of the cluster installation: the actual installation and configuration of cluster machines. s we mentioned in a previous paragraph, before the installation of the nodes of the cluster via LCFG can start, their names and IPs must be registered in the DNS server of the site. This is important because each host looks up for its hostname and IP in the DNS table and if it is not there, the installation cannot continue.

In the next paragraphs, we try to keep the per-element procedure of configuring, for two main reasons. First, we find appropriate to have an installation procedure that will start and finish on the same host, independently of the middleware installed. Besides that, the per-element configuration has the advantage of gathering all the information of one host; also, the various bugs of the installation and configuration are more easily identified.

Defining the Boot Method

You have two options on how to boot the nodes for the first time: one is to use the LCFGng bootdisk and the other PXE. If a machine doesn't have a floppy disk the later method is inevitable.

Configure LCFGng

The boot method of every node is configured in the LCFGng server using a web interface. Use your favorite web browser and go to the following URL:

```
http://<LCFGngServer>/install/install.cgi
```

The browser will prompt for a username and a password. These refer to the **lcfgng** web user that we created when we were setting up the Apache Web server. After you enter the login and the password you are presented with the following screen:
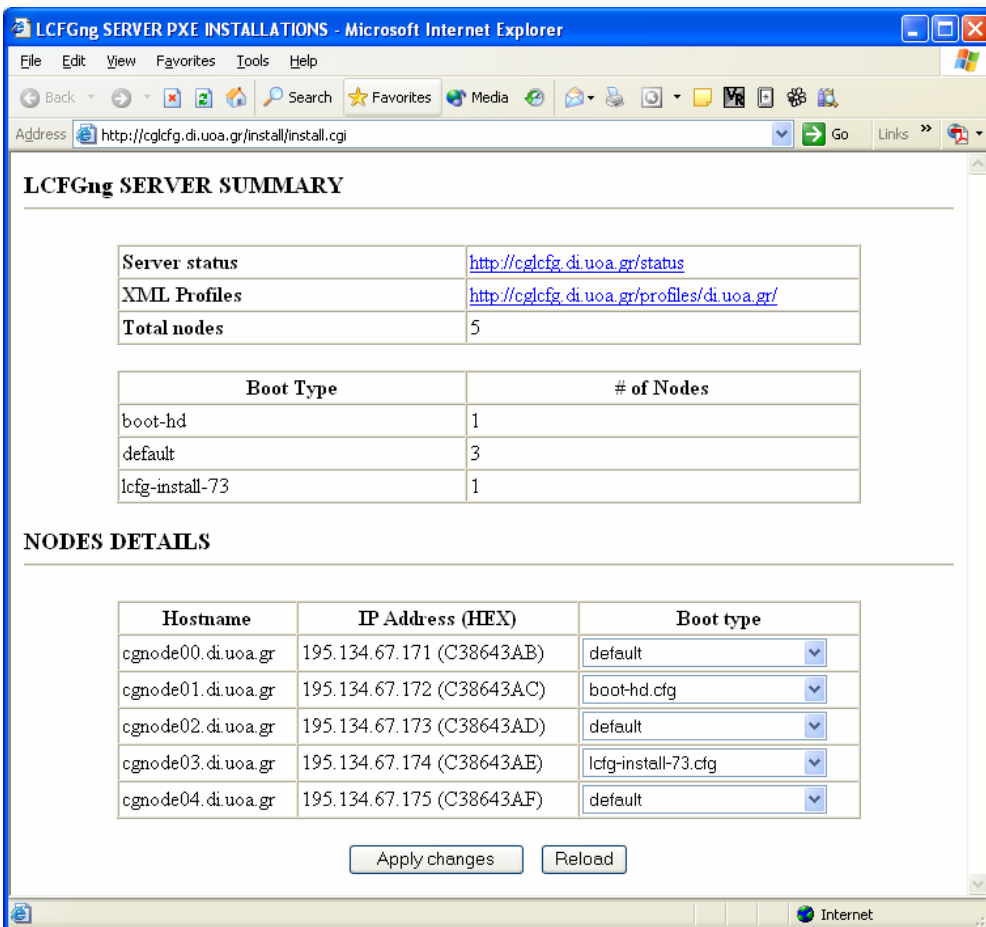
You can see a list of the client nodes and a drop down list of the boot type for each one of them. The available options are:

1. **boot-hd.cfg**, boots from the Hard Disk

2. **default**, same as boot-hd.cfg

3. **lcfg-install-73.cfg**, boots from the network and installs Red Hat 7.3. Before starting the installation the system will ask for confirmation from the administrator

4. **lcfg-install-62.cfg,** boots from the network and installs Red Hat 6.2. Before starting the installation the system will ask for confirmation from the administrator. Not applicable in our case

5. **lcfg-install-nointeract-73.cfg,** boots from the network and installs Red Hat 7.3 without asking for confirmation

6. **lcfg-install-nointeract-62.cfg,** boots from the network and installs Red Hat 6.2 without asking for confirmation. Not applicable in our case

Choose the desired option among 1, 2, 3 and 5. Press Apply Changes to to activate them.

Configure the client node

In each node you are going to install you have to define in the BIOS settings the preferable boot sequence, that is in which order the system should check for bootable devices. Turn on the machine and enter the BIOS setup (usually by pressing DEL). If this node will boot from diskette check that the first boot device is the floppy disk. If PXE is going to be used select Network for the first boot device. Save the changes and reboot.

In the case of PXE you will probably have to setup also the Network Card BIOS. For Realtek cards you can enter the BIOS by pressing Shift-F10 during the machine boot. In the configuration screen there are three options: enable PXE, disable PXE or to use system BIOS settings. Either the first or the last option should work fine.

Computing Element (CE)

The installation starts with the CE since it is the heart of the local Grid.

System Installation

If you are using the boot disk insert it in the floppy drive and turn-on the machine. The system boots from the disk and presents a menu (Grub boot loader options actually) for the boot method to use, interactive or non-interactive. Both methods are similar; in the first case the system will ask for confirmation before starting the installation (and thus the destruction of the existing system in the hard disk – if any). The first thing that the installation process does is to get in contact with the DCHP server and retrieve IP information. After that it proceeds with the download of the profile from the LCFGng server and checks what packages it will install and where it to find them. Next it mounts the exported directories of LCFGng server with NFS and starts the installation of the packages.

PXE installation is similar. The system starts and seeks for a host in the network to service it. LCFGng server responds and provides the new PC with the boot kernel. The rest of the procedure is identical with the bootdisk one.

The installation in all machines is done in two phases: In the first phase Linux is installed with all required core packages. After this the system reboots and continues with the installation of Grid software. During this first reboot the system will search for devices that haven't been installed automatically and will try to configure them. Be selective with which devices you are going to enable. For instance if a sound card is found don't activate it!

The procedure takes about 30' in a Pentium 4 machine.

When the system boots for the second time you proceed with the fine tuning and manual configuration of sub-systems that are not setup automatically.

Globus Gatekeeper Certificates

For the Globus Gatekeeper service to run, a certificate has to be requested by the designated Certification Authority and installed in the Computing Element. For this to happen one has to generate first a public-private key pair and a certificate request file. The procedure is as follows:

Login to the CE and at the shell prompt enter the following command:

```
$ grid-cert-request –host <CE-FQDN> \
-key /opt/globus/etc/<CEHostname>-computingelement.key \
-cert /opt/globus/etc/<CEHostname>-computingelement.cert \
-req /opt/globus/etc/<CEHostname>-computingelement.req
```

At the above command, replace <CE-FQDN> with the Fully Qualified Domain Name of your own Computing Element (e.g. cgnode00.di.uoa.gr), and <CEHostname> with its hostname (e.g. cgnode00). Grid-cert-request creates three files in /opt/globus/etc:

- **<CE-Hostname>-computingelement.key**, which contains the private key of the gatekeeper.

- **<CE-Hostname>-computingelement.req**, which contains the public key bundled with additional information, in a so called PKCS#10 format, appropriate for requesting a certificate from a CA.

- **<CEHostname>-computingelement.cert**, which is zero-sized and essentially provides a placeholder for the certificate that will be issued by the CA.

Next simply email the request file (<CE-Hostname>-computingelement.req) to you're the site/VO designated Certification Authority. Probably for this a formal procedure has to be followed, which is defined in the CA Certificate Practice Statement. One has to consult his/her CA responsible for the exact requirements.

The certificate will usually be sent to you, from the CA, by email as an attached file. This file must be saved in /opt/globus/etc with the name <CE-Hostname>-computingelement.cert, overwriting the existing zero-sized file.

Finally, start the gatekeeper:

```
$ service globus-gatekeeper start
```

The Computing Element is now ready to accept remote job submissions.

PBS Configuration

PBS (Portable Batch System) is the software responsible for distribution of the jobs, that are submitted to the Computing Element, to all the Worker Nodes. LCFGng can't handle the PBS configuration so it's necessary to manually customize the system.

Login as root in CE and edit the configuration file /var/spool/pbs/pbs_server.conf according to the following example:

```
#
# Create queues and set their attributes.
#
#
# Create and define queue workq
#

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
set queue short max_queuable = 4
set queue short max_running = 2

create queue long
set queue long queue_type = Execution
set queue long resources_max.cput = 12:00:00
set queue long resources_max.walltime = 24:00:00
set queue long enabled = True
set queue long started = True
set queue long max_queuable = 4
set queue long max_running = 2

create queue infinite
set queue infinite queue_type = Execution
set queue infinite resources_max.cput = 48:00:00
set queue infinite resources_max.walltime = 72:00:00
set queue infinite enabled = True
set queue infinite started = True
set queue infinite max_queuable = 4
set queue infinite max_running = 2

#
# Set server attributes.
#
set server scheduling = True
set server acl_host_enable = False
set server default_queue = short
set server managers = root@<CEhostname>
set server operators = root@<CEhostname>
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 = cgpro
set server node_pack = False # important for job distribution!
```

Replace <CEhostname> with the full qualified name of your CE (e.g. cgnode00.di.uoa.gr). Note that queues short, long, and infinite are those defined in the site-cfg.h file and the time limits are those in use at the original LCG-1 testbed. Feel free to add/remove/modify them to your liking but do not forget to modify site-cfg.h accordingly.

The variables max_queuable and max_running define the maximum number of jobs that can be queued and run respectively simultanously by the specific queue. It is advisable that

Edit file /var/spool/pbs/server_priv/nodes to add the list of WorkerNodes you plan to use. For instance the following applies to the University of Athens:

```
cgnode01.di.uoa.gr        np=2    cgpro
cgnode03.di.uoa.gr        np=2    cgpro
```

where np=2 gives the number of job slots (usually equal to #CPUs) available on the node, and cgpro is the group name as defined in the default_node parameter in pbs_server.conf.

Make a backup of the nodes file:

```
$ cp /var/spool/pbs/server_priv/nodes /var/spool/pbs
```

Create the queues and "feed" the queue manager with the pbs server configuration file:

```
$ pbs_server -t create
$ qmgr < /var/spool/pbs/pbs_server.conf
```

Note: Check the file /var/spool/pbs/server_name. This file should include the hostname of your PBS server (the CE hostname).

Finally restart the pbs daemon (restore also the `nodes` file because it keeps deleting after the feed!!):

```
$ cp /var/spool/pbs/nodes /usr/spool/pbs/server_priv
$ /etc/rc.d/init.d/pbs_server restart
```

Publishing WN information from the CE

When submitting a job, Grid users are supposed to state, using a Job Description Language (JDL), the minimal hardware resources (memory, scratch disk space, CPU time) required to run the job. These requirements are matched by the Resource Broker with the information on LDAP servers to select a set of available CEs where the job can run. For this schema to work, each CE must publish some information about the hardware configuration of the WNs connected to it. This

means that site administrators must collect information about WNs available at the site and insert it in the information published by the local CE.

The procedure to do this is the following:

- choose a WN which is "representative" of your batch system (see below for a definition of "representative") and make sure that the chosen node is fully installed and configured. In particular, check if all expected NFS partitions are correctly mounted.

- on the chosen WN run the following script as root, saving the output to a file.

```
#!/bin/bash
echo -n 'hostname: '
host `hostname -f` | sed -e 's/ has address.*//'
echo "Dummy: `uname -a`"
echo "OS_release: `uname -r`"
echo "OS_version: `uname -v`"
cat /proc/cpuinfo /proc/meminfo /proc/mounts
df
```

- copy the obtained file to /opt/edg/var/info/edg-scl-desc.txt on your CE, replacing any pre-existing version.

- restart the local site MDS server on the CE with

```
$ /etc/rc.d/init.d/globus-mds restart
```

We come now to the issue of "representative WN". In general, WNs are added to a batch system at different times and with heterogeneous hardware configurations. All these WNs often end up being part of a single queue, so that when an LCG job is sent to the batch system, there is no way to ask for a specific hardware configuration (note: LSF and other batch systems offer ways to do this but the current version of the Globus gatekeeper is not able to take advantage of this possibility). This means that the site manager has to choose a single WN as "representative" of the whole batch cluster. In general it is recommended that this node is chosen among the "least powerful" ones, to avoid sending jobs with heavy hardware requirements to under-spec nodes.

Storage Element (SE)

System Installation

The procedure of installing the system in the Storage Element is exactly the same as the one followed for the CE. It will take approximately 30' in a Pentium 4 machine. When the system boots for the second time you proceed with the fine tuning and manual configuration of sub-systems that are not setup automatically.

Globus Gatekeeper Certificates

As with the Computing Element and since the Storage Element E hosts a Globus Gatekeeper, one has to request and install a digital certificate for the service. Currently there is a controversy regarding to the presence of the gatekeeper in the Storage Elements. You may choose not to activate it but there are situation that it's useful to use it for SE testing and profiling purposes. In any case it is a good idea to have the certificates ready regardless

The procedure is similar to the one for the CE:

Login to the SE and at the shell prompt enter the following command:

```
$ grid-cert-request –host <SE-FQDN> \
-key /opt/globus/etc/<SEHostname>-storageelement.key \
-cert /opt/globus/etc/<SEHostname>-storageelement.cert \
-req /opt/globus/etc/<SEHostname>-storageelement.req
```

At the above command replace <SE-FQDN> with the Fully Qualified Domain Name of the Storage Element (e.g. cgnode04.di.uoa.gr), and <SEHostname> with its hostname (e.g. cgnode04). Grid-cert-request will create three files in /opt/globus/etc:

- **<SE-Hostname>-storageelement.key**, which contains the private key of the gatekeeper.

- **<SE-Hostname>-storageelement.req**, which contains the public key bundled with additional information, in a so called PKCS#10 format, appropriate for requesting a certificate from a CA.

- **<SE-Hostname>-storageelement.cert**, which is zero-sized and essentially provides a placeholder for the certificate that will be issued by the CA.

Then email the request file (<SE-Hostname>-computingelement.req) to the site/VO designated Certification Authority. When the certificate has been received from the CA it has to be copied in /opt/globus/etc with the name <SE-Hostname>-storageelement.cert, overwriting the existing zero-sized file.

Finally start the gatekeeper:

```
$ service globus-gatekeeper start
```

Worker Node (WN)

Install the Worker Nodes in the same manner as the Computing Element. The process will take a little longer for the WNs because lot's of libraries and application software of various experiments has to be installed. In a Pentium 4 machine it will take approximately 50' minutes to complete the installation.

Fine Tuning

The default allowed maximum number of open file on a RedHat node is only 26213. This number might be too small if users submit file-hungry jobs so you may want to increase it on your WNs. for instance to set this parameter to 256000 you can use this command:

```
$ echo 256000 > /proc/sys/fs/file-max
```

You can make this setting reboot-proof by adding the following code at the end of your /etc/rc.d/rc.local file:

```
# Increase max number of open files
if [ -f /proc/sys/fs/file-max ]; then
    echo 256000 > /proc/sys/fs/file-max
fi
```

User Interface (UI)

Install the User Interface in the same manner as the other nodes. In a Pentium 4 machine it will take approximately 30' minutes to complete the installation. When the system completes installation it is ready for use. No further configuration is required

Chapter

5

Regular Upgrades

T he end of the Installation and Configuration procedure marks the beginning of the next long-term cycle, which is the daily base maintenance and upgrading of a Grid cluster. Here LCFGng demonstrates its power as a cluster maintenance tool. In general the task of maintaining a cluster up-to-date and trouble free, proves to be much easier than the installation of the cluster per se. Most of the tasks are carried out centralized from the LCFGng server and there are only a few occasions that one has to actually login to a specific host to perform an action.

The LCG-1 development team releases regularly new versions of the LCG-1 software bundle with bug fixes and enhancements. CrossGrid usually adopts immediately the new release and may also provide its own releases based on the progress and requirements of the internal developed applications. It is advisable to keep your cluster up-to-date by following each new release and upgrading your cluster. The upgrade procedure consists of the following steps:

First you have to retrieve a new set of rpmlists and source files from the CVS. Go to the directory under which you keep the CVS tree (the one in which you have issued cvs checkout at the beginning of installation). Then issue the commands:

```
$ export CVS_RSH=`which ssh`
$ export CVSROOT=:ext:anoncvs@gridportal.fzk.de:/data/cvs/crossgrid
$ cvs update crossgrid/wp4/sites
```

This will update in the Crossgrid directory all the files that have changed since the previous release. Make a note of these files. In directory common/lcg1 you can find a text file named lcg1-notes.txt. This file contains important instructions for updating your cluster. Actually many portions of the manual originate from this document.

Copy all the updated rpmlists (both from common/lcg1/rpmlist and common/cg/ng_rpmlist in /opt/local/linux/7.3/rpmcfg.

Read lcg1-notes.txt to find specific instructions on what needs to be changed in source files. Normally changes should be made only to site-cfg.h file. Especially look at Appendix E which usually contains detailed upgrade instructions. Apply the changes to the source files.

Download any new required RPM package with updateRPMcache.pl:

```
$ cd common/edg/updaterep
$ ./updateRPMcache.pl /var/obj/conf/server/source/grid*
```

Finally generate the new profiles:

```
$ service mkxprofd reload
```

Remember to check http://<LCFGServer>/status/ for any possible errors. The new profiles will propagate to the client nodes in a couple of minutes. If there are new packages that have to be installed this will also be done automatically after 5-10 minutes.

To force an update of a node, login to that specific node and issue the commands:

```
$ rdxprof
$ /etc/obj/updaterpms run
```

updaterpms will check the rpm list from the LCFG server and install all new packages listed.

References

1) "The Anatomy of the GRID: Enabling Scalable Virtual Organizations" Ian Foster, Carl Kesselman, Steven Tuecke (to appear: Intl. J. Supercomputer Applications, 2001)

2) "Large scale Linux Configurations with LCFG" Paul Anderson, Alastair Scobie

3) "LCFG: The Next Generation" Paul Anderson, Alastair Scobie

4) "Getting started with LCFG" Paul Anderson Jessie Paterson DICE Computing Environment Project

5) "LCG-1 Installation notes", Emanuele Leonardi (Available from LCG1 CVS. Updated regularly with every new LCG-1 release)

6) "EDG LCFGng Server Installation using RedHat 7.3", C. Aiftimiei, E. Ferro, C. Loomis, 22 September 2003 http://datagrid.in2p3.fr/distribution/datagrid/wp4/edg-lcfg/documentation/lcfgng-server73.html

7) "WP4 LCFG FAQ" Massimo Biasotto, Andrea Chierici, Enrico Ferro, Marco Serra

8) "LCFG Site Server Installation Guide" Mohammad Jaudet, Julian Blake, Jan Iven

9) "First Time LCFG Server Installation: red Hat 6.2 with Updates" Julian Blake

10) "CrossGRID services at LIP" http://www.lip.pt/computing/projects/crossgrid/crossgrid-services/

11) "The CrossGrid Test and Validation testbed" http://www.lip.pt/computing/projects/crossgrid/crossgrid-tv/services.htm

12) "The CrossGrid Production testbed" http://www.lip.pt/computing/projects/crossgrid/crossgrid-pro/