

Simplified deployment of a LCG cluster via LCFG-UML

Ariel García*, Marcus Hardt†, Forschungszentrum Karlsruhe, IWR,
P.O. Box 3640, D 76021 Karlsruhe, Germany
George Tsouloupas‡, High Performance Computing systems Laboratory,
Univ. of Cyprus, P.O. Box 20537, CY 1678 Nicosia, Cyprus

Abstract

We present a User Mode Linux image preconfigured as an LCFG cluster installation and configuration server. This server is required for setting up and managing LHC Computing Grid (LCG) type of clusters, if a full manual installation is to be avoided. The aim of this development is twofold. First, to avoid most of the overhead associated with the setup of such a cluster management server, lowering the entry barrier for sites who want to do a first LCG-grid cluster deployment. The second objective is to avoid investing dedicated hardware resources in this management service.

INTRODUCTION

The clusters using European DataGrid (EDG) [1]/LHC Computing Grid (LCG) [2] middleware are usually managed by means of the “LCFG” software [3] originally developed by the Univ. of Edinburgh and extended by DataGrid. By default, the available documentation assumes you will be using a dedicated machine for running that software. It requires to setup several services, which is a time-consuming task. However, apart from generic functionality (NFS serving), this server needs only very little processing power. Therefore the idea behind setting up a User Mode Linux (UML) [4] LCFG server is to avoid wasting hardware resources, and, at the same time, to make it easier for a new site to join an EDG/LCG Grid, bypassing part of the LCFG server installation.

An LCFG server consists basically of the following services:

- profile compilation
- profile publishing (via http/apache webserver)
- package repository (NFS export)
- additional services: dhcp server, tftp server for pxe boot, name server, time server

Most of these pieces can be easily “boxed” in a User Mode Linux server. The exception is the NFS package repository, which should be kept outside the UML server simply because of space (and potentially performance) reasons. Additionally, running the DHCP server inside the UML guest requires a somewhat more delicate networking setup in the

host. Namely, what is needed is a bridged network in the host machine to allow the dhcp server inside UML receiving the broadcasted packets.

In the present work we give detailed instructions for setting up such a UML-LCFG server, using a prebuilt root-filesystem image we provide for download. This UML instance (which we will call *UML_LCFG*) will be running inside a machine that we call *HOST*. This host computer can be running any Linux distribution, although in the “HOW-TO” notes we assume using a Red Hat 7.3 host; names and versions of a couple of needed packages could (and most probably will!) differ otherwise. However, running a UML guest is possible in any distribution, and the steps described in the Notes should apply without any major issues.

The provided *UML_LCFG* image is preconfigured, requiring essentially the replacement of the site’s network address and domain in a handful of files. The day to day running of the LCFG server, i.e., updating the middleware releases and the configuration of the cluster nodes, is streamlined by the usage of CrossGrid’s tool “cg-lcfg-go”. It should be noted that the whole setup easily supports both a CrossGrid [5] as well as a LCG deployment.

REQUIREMENTS AND SETUP DESCRIPTION

In this section we briefly discuss some requirements related to the LCFG server, which need to be fulfilled for setting up an EDG/LCG cluster.

The package repository

While the *UML_LCFG* server will produce the configuration profiles and hold the lists of RPMs to be installed in the clients, the RPM packages themselves should be held outside the UML image due to space and performance reasons. The complete LCG package repository contains nowadays 23GB of packages, although packages needed for installing the clients add up to around 1GB. Therefore we recommend to host the RPM repository in some external NFS server, which could be the *HOST* machine itself.

DNS entries

For installing the client nodes you *need* to have DNS entries for them. You can use the DNS server provided inside the *UML_LCFG* server itself but you will need to keep this UML instance always running. We aware that if external access to your cluster is required you will probably be forced to use your site/institution’s DNS server anyway.

* garcia@iwr.fzk.de

† hardt@iwr.fzk.de

‡ georget@ucy.ac.cy

Network time service

For the Globus Security Infrastructure (and therefore the whole EDG/LCG middleware) to work you need to have your system clocks correctly synchronized. The installation software will configure ntp clients in each node, but you have to provide an NTP time server. If you plan to keep your UML instance always running, you can use the one provided inside the *UML_LCFG* server itself. Otherwise an alternative would be to use a NTP time server provided by your site/institution, or some other public NTP server like `europa.pool.ntp.org`.

The DHCP server

For doing the installation a DHCP server is required. Running it inside the *UML_LCFG* server itself requires setting up bridging in the *HOST* machine. This is so because the dhcp daemon inside the UML box has to be able to receive the broadcasted requests. This additional configuration needed in the *HOST* can be done automatically by running the script “`lcfgng-uml-with-bridge.start`” provided together with the other files. This script will also start the UML instance under a non-privileged user identity (“`umluser`” by default), not as “`root`”.

DEPLOYING THE LCFG SERVER

All the files mentioned in this document are available for download from the webpage

`http://cvs.fzk.de/s1/uml-lcfg`

The required files are:

- `UML-LCFG.txt` : the main “HOW-TO” file containing the procedure to follow,
- `linux-uml-2.4.20` : a precompiled UML kernel binary supporting bridging,
- `lcfgng-rootfs.gz` : the (compressed) root-filesystem image of the *UML_LCFG* server,
- `lcfgng-uml-with-bridge.start` : a script which configures bridging in the *HOST* machine and starts the *UML_LCFG* instance as a non-privileged user.

Some RPM packages you might need are also provided there.

We refer the reader to the main document “`UML-LCFG.txt`” for the detailed setup instructions. Only a general overview of the setup is provided below.

After downloading the required files, you will configure the `lcfgng-rootfs` root-filesystem image with your site’s network parameters. This is done *before* starting the UML instance itself, simply by mounting the filesystem image and editing the configuration files inside. The configuration files that must be edited are all linked from `/root/LCFG/etc`. They are clearly commented inside, and they also contain a sample entry referring to the *fzk.de*

site. Therefore a plain search and replace for the domain name *fzk.de* and the network IP 141.52. will do most of the job...

You are also instructed to set your own root password at this stage.

The `/etc/dhcp.conf` file inside the image can be used for configuring the dhcp server. This applies both if this daemon is run internally as well as if you chose to keep it external to avoid bridging.

A simple set of firewall rules are also set up inside the preconfigured image, blocking most of the access from outside your site except for Secure Shell connections.

After finishing the above changes, you must unmount the image and start the User Mode Linux binary (namely `linux-uml-2.4.20`) with this file as root filesystem. This can be done by typing the corresponding command line at the prompt. However, as already mentioned above, a bridged network setup is needed for running an internal dhcp server.

This bridged setup can be conveniently done with the script `lcfgng-uml-with-bridge.start`, which will at the end run your UML instance. This script will first read your current network setup and detect the default interface. This last one is then (turned down and) connected to a bridge, together with the virtual interface which UML will use later. After the network setup is complete, the screen command is used to run the UML process. This allows the administrator to easily reconnect to it later without the need of a network login.

Installing your cluster

Once your *UML_LCFG* is running, you must prepare the installation of your cluster nodes. We assume that you will install either a CrossGrid release (currently CG2.0.X in production), or a standard LCG distribution (currently LCG2.2.0). In any case, we recommend that you keep at hand the very detailed “CrossGrid’s LCFGng Cluster Installation Guide” [7] or the “LCG Installation Manual” [8], respectively, skipping in both cases the LCFG server setup instructions. We will not repeat here the steps which are already described there.

Even if planning to install an LCG cluster, you might still find it convenient and easier to use CrossGrid’s LCFG server tool “`cg-lcfg-go`” [9]. This script

- downloads the latest LCG/CrossGrid profiles from the CVS repository,
- downloads any newly needed rpm packages (required either by the *UML_LCFG* server or by any of the clients),
- updates the packages in the *UML_LCFG* server (required for correctly processing the client profiles),
- compiles the xml profiles which the clients will read through http.

This tool is available in the path `/opt/cg/sbin/` (already in root's PATH), and is fully configurable. By default, the settings are read from `/etc/site/cg-lcfg-go.conf`. A couple of functional configuration files are already provided:

- CrossGrid: `/etc/site/cg-lcfg-go.cg-conf`
- LCG: `/etc/site/cg-lcfg-go.lcg-conf`

Just copy or link the right one to `cg-lcfg-go.conf`. When the tool is invoked, you will be asked interactively which of the above steps to execute.

Important paths

There are a few important paths that you must know about, because most of the cluster configuration happens there:

- Your RPM package repository:
`/opt/local/linux/7.3/RPMS`
Remember this is an NFS mounted folder as explained above, exported from the `RPM_HOST`.
- The profiles of your client hosts:
`/opt/local/linux/7.3/cvs/my-site/ng_source`
- The rpmlists of your client hosts:
`/opt/local/linux/7.3/cvs/my-site/ng_rpmcfg`

Client profiles

Once your `UMLLCFG` server is functional, you still have to set up the configuration files for the cluster you want to install. These are the source profiles containing the node configuration settings (in the `ng_source` subfolder), and the lists of RPM packages to install in each node (in the `ng_rpmlist` subfolder). All these files are located in the subtree `/opt/local/linux/7.3/cvs/`, as already emphasized above. The relevant subfolders are conveniently linked from within the path `/root/LCFG`.

Note that in this setup, where the RPM package repository resides outside your LCFG server, you need to modify the client profiles accordingly. This simply means to add the RPM repository mountpoint in addition to the default `SITE_LCFG_SERVER:/opt/local/linux/7.3`. For this, edit the `nfsmount` entries in the file `ng_source/redhat73-cfg.h` as follows:

```
-----
nfsmount.nfsmount  lcfg rpmsrepo
nfsmount.nfsdetails_lcfg
    /export/local/linux/7.3
    SITE_LCFG_SERVER:/opt/local/linux/7.3
    ro,noexec
nfsmount.nfsdetails_rpmsrepo
    /export/local/linux/7.3/RPMS
    <RPMS_REPO_SERVER>:/opt/local/linux/7.3/RPMS
    ro,nfsvers=3,noexec
-----
```

(note that the indented lines above are continued from the previous ones). Moreover, if you are installing a LCG cluster, make sure to add the following line in your `site-cfg.h` file:

```
#define RPMDIR /export/local/linux/7.3/RPMS/lcg
```

This is needed for the `updaterpms` object to find all the RPMs you're providing.

For more details refer to the CrossGrid/LCG installation guides, [7] and [8].

Cluster installation

Once your profiles are prepared, you have to run

```
cg-lcfg-go -p
```

If the `cg-lcfg-go` output shows "0 errors" and "XML published" for each of the cluster nodes, then this means that the profiles are ready to be used by the clients. It is then time to activate the PXE boot and power up the machines to be installed.

Once the client machines are installed and running, a few postinstallation steps remain to be done. These are described in detail in the corresponding CrossGrid/LCG Installation notes, [7] and [8].

ACKNOWLEDGEMENTS

The present work was developed in the framework of the EU CrossGrid project, IST 2001-32243.

REFERENCES

- [1] <http://www.eu-datagrid.org>
- [2] <http://lcg.web.cern.ch/LCG>
- [3] <http://www.lcfg.org>
- [4] <http://user-mode-linux.sourceforge.net>
- [5] <http://www.eu-crossgrid.org>
- [6] <http://cvs.fzk.de/sl/uml-lcfg>
- [7] CrossGrid's LCFGng Cluster Installation Guide, <http://cgi.di.uoa.gr/~xgrid/cgfiles/LCFGng-v2.0.pdf>
- [8] Releases on the LCG Deployment homepage, follow the 'lcg2-install-notes' link in the latest release section: <http://grid-deployment.web.cern.ch/grid-deployment/cgi-bin/index.cgi?var=releases>
- [9] Package named `cg-wp4-lcfg-server-files-*`, the latest version is available from <http://gridportal.fzk.de/distribution/crossgrid/releases/allfiles/7.3/cg/wp4/>