Guide to LCMAPS version 0.0.16

Martijn Steenbakkers

15 September 2003

1 Introduction

The Gridification subtask of WP4 of the European Datagrid project[1] interfaces the local fabric to other middleware components by a number of services, among which the Local Centre Authorization Service (LCAS) handles authorization requests to the local computing fabric and the Local Credential Mapping Service (LCMAPS) provides all local credentials needed for jobs allowed into the fabric. This document describes a prototype version of LCMAPS, which is the second component released by the Gridification subtask, the first being LCAS.

Initially LCMAPS will only be used by the gatekeeper running on a Computing Element (CE), but eventually other services (e.g. gridftp server) may rely on LCMAPS for their local credential mapping. LCMAPS is implemented as a shared library, which is loaded dynamically by the globus gatekeeper. The gatekeeper has been slightly modified for this purpose and will from now on be referred to as edg-gatekeeper.

LCMAPS is a framework that can load and run one or more 'credential mapping' plugins. The use of a plugin-framework architecture for LCMAPS makes it very easy for sites/organizations to add new functionality to LCMAPS by writing new plugins. The LCMAPS framework consists of the following components:

- the *plugin manager*, which is responsible for managing, loading and running the LCMAPS plugins.
- the *evaluation manager*, which is responsible for the order in which the LCMAPS plugins are called. The evaluation manager is driven

[1] http://www.eu-datagrid.org

by a policy engine, which is documented in more detail here[2] , or as PostScript file[3] or PDF file[4].

Based on the user global credentials (more specifically the user's X509 certificate) and the job specification (JDL), the LCMAPS plugins have to perform either of these two tasks:

1. acquire local credentials (**A**).
2. enforce (apply) the local credentials (**E**).

The local credentials that are gathered (UNIX uids, gids, VO information, AFS/Kerberos (?) tokens), are stored internally, but a new WP4 component, the job repository, is foreseen in which these credentials may be stored as well and which is accessible by other applications and services. The following LCMAPS plugins are currently available:

- plugins providing the functionality that is equivalent to the functionality of the original gatekeeper:
  - `lcmaps_localaccount.mod` (**A**): this plugin collects the local account name from a *gridmap* file. More info ...
  - `lcmaps_poolaccount` (**A**): this plugins collects a pool account name from a *gridmap* file (leases in $GRIDMAPDIR). More info ...
  - `lcmaps_posix_enf.mod` (**E**): this plugin enforces the local credentials in the running process by posix system calls (setuid(), setgid() etc.). More info ...
  - `lcmaps_ldap_enf.mod` (**E**): this plugin enforces the local credentials by setting the primary and secondary gids in the LDAP database that is used by the site as the source of account information for PAM or NSS. More info ...
- plugins that use the VOMS (VO Membership Service) attribute assertions in the user certificate for the credential mapping.
  - `lcmaps_voms.mod` (**A**): this plugin extracts the VOMS information from the user X509 proxy certificate. More info ...

[2] pdl_requirements/index.html
[3] pdl_requirements.ps
[4] pdl_requirements.pdf

- `lcmaps_voms_localgroup.mod` (**A**): this plugin tries to find a local group Id (gid) based on the VO information and a *groupmapfile*. More info ...
- `lcmaps_voms_poolgroup.mod` (**A**): this plugin tries to find a pool group Id (gid) based on the VO information and a *groupmapfile* (leases in $GROUPMAPDIR) More info ...
- `lcmaps_voms_poolaccount.mod` (**A**): this plugin tries to find a pool account based on the VO information and a *gridmapfile* (leases in $GRIDMAPDIR) More info ...

More information on LCMAPS and other components of the Gridification subsystem can be found in:

- the WP4 architecture document D4.2: pdf version[5] or doc version[6].
- LCAS: http://www.dutchgrid.nl/DataGrid/wp4/lcas/edg-lcas-1.1/[7]
- the description of the LCMAPS API: here[8] , PostScript file[9] and PDF file[10].
- the LCMAPS policy description language (pdl): here[11] , or as PostScript file[12] or PDF file[13].
- the README[14], INSTALL[15], and LICENSE[16] files.
- the file containing instructions how to avoid LDAP as a source of user accounting information README.NO_LDAP[17],

A few *example* scripts are added, which can be used to setup poolaccounts, poolgroups in LDAP:

[5] http://hep-proj-grid-fabric.web.cern.ch/hep-proj-grid-fabric/architecture/eu/WP4-architecture-2_1.pdf
[6] http://hep-proj-grid-fabric.web.cern.ch/hep-proj-grid-fabric/architecture/eu/WP4-architecture-2_1.doc
[7] http://www.dutchgrid.nl/DataGrid/wp4/lcas/edg-lcas-1.1/
[8] apidoc/html/index.html
[9] apidoc/latex/refman.ps
[10] apidoc/latex/refman.pdf
[11] pdl_requirements/index.html
[12] pdl_requirements.ps
[13] pdl_requirements.pdf
[14] README
[15] INSTALL
[16] LICENSE
[17] README.NO_LDAP

Table 1: RPMs to be installed.

RPM	min. version	description + URL
`edg-lcmaps`	0.0.16	the LCMAPS library (= pluginframework + utilities) and an example LCMAPS plugin
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp4/gridification/RPMS/
`edg-lcmaps-basic_plugins`	0.0.16	the LCMAPS plugins providing the basic globus-gatekeeper functionality
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp4/gridification/RPMS/
`edg-lcmaps-voms_plugins`	0.0.16	the LCMAPS plugins that base the credential mapping on the VO information inside the user certificate
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp4/gridification/RPMS/
`edg-lcmaps-interface`	0.0.1	LCMAPS interface/API, only needed for software development (new plugins)
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp4/gridification/RPMS/
`edg-lcfg-lcmaps`	1.0	the LCFG object that configures the LCMAPS configuration files
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp4/edg-lcfg/RPMS/
`voms-api`	1.1.16	the VOMS API, used by `edg-lcmaps-voms_plugins`
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp6/RPMS/
`edg_gatekeeper-gcc32dbg_pgm`	2.2.8	the modified globus gatekeeper
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp4/gridification/RPMS/
`globus-config`	0.20-1	globus configuration scripts, including the init.d gatekeeper script
		http://datagrid.in2p3.fr/distribution/globus/config/RPMS/
`edg-lcfg-globuscfg`	1.3.12	the LCFG component to manage the Globus configuration
		http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp4/edg-lcfg/RPMS/
`vdt_globus_essentials`	VDTALT1.1.8	VDT globus rpm that contains a.o. the security libraries
		http://www.lsc-group.phys.uwm.edu/vdt/vdt_rpms/edg/vdt-1.1.8/globus_coarse_rpm/

- lcmaps_gen_poolacc_ldif[18]
- lcmaps_gen_poolgroup_ldif[19]
- lcmaps_make_poolacc_dir[20]
- lcmaps_setup_pool[21]

2 Installation

LCMAPS uses the globus security libraries (gss, gsi, openssl), which are provided by e.g. VDT (Virtual Data Toolkit) and the VOMS API. These libraries in addition to the libraries listed in table1 , have to be installed on the CE.

The LCMAPS library will be installed in `/opt/edg/lib/lcmaps/` and the example configuration files in `/opt/edg/etc/lcmaps/`. The LCMAPS plugins are all installed in the `modules` subdirectory of the directory where LCMAPS is installed.

From CVS:
The LCMAPS library and plugins can also be built directly from the cvs

[18] lcmaps_gen_poolacc_ldif
[19] lcmaps_gen_poolgroup_ldif
[20] lcmaps_make_poolacc_dir
[21] lcmaps_setup_pool

repository[22] by the following steps:

- cvs export -r <version_tag> fabric_mgt/gridification/lcmaps – export the source from CVS using a tagged version (e.g. v0_0_1)
- cd fabric_mgt/gridification/lcmaps; ./autogen.sh – run the bootstrap script to run autotools
- ./configure --prefix=<path> --libdir=<path>/lib/lcmaps --includedir=<path>/include/lcmaps --sysconfdir=<path>/etc/lcmaps – run the configure script
- make rpm – if you want to make the rpm
- make; make install – build and install the LCMAPS Library and the LCMAPS plugins
- make apidoc – if you want to create the API documentation. This is for example useful for developers of new LCMAPS plugins.
- make userdoc – Generate this documentation.

3 Configuration

The configuration involves both LCMAPS itself and the edg-gatekeeper.

3.1 Configuration of the edg-gatekeeper

The edg-gatekeeper is configurable with a few more command line options in addition to the normal globus-gatekeeper options:

-lcmaps_debug_level <debug level>:	set the debug level for LCMAPS (default: 0 (= no debugging))
-lcmaps_db_file <file>:	specifies the filename of the LCMAPS policy file (default: lcmaps.db).
-lcmaps_etc_dir <path>:	specifies the directory where the LCMAPS configuration files are located (default: /opt/edg/etc/lcmaps/).
-lcmapsmod_dir <path>:	specifies the directory where the LCMAPS library is located (default: /opt/edg/lib/lcmaps/).
-lcas_debug_level <debug level>:	set the debug level for LCAS (0–5, default: 0 (= no debugging))
-lcas_db_file <file>:	specifies the filename of the LCAS policy file

(default: lcas.db).

[22] http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/fabric_mgt/gridification/lcmaps/

-lcas_etc_dir <path>:	specifies the directory where the LCAS authorization configuration files are located (default `/opt/edg/etc/lcas/`).
-lcas_dir <path>:	same as -lcas_etc_dir ¡path¿, deprecated.
-lcasmod_dir <path>:	specifies the directory where the LCAS library is located (default `/opt/edg/lib/lcas/`).
-plainoldglobus:	provides the old globus-gatekeeper functionality, LCAS and LCMAPS are not used.
-no_lcas:	do not use LCAS.
-no_lcmaps:	do not use LCMAPS (use standard gridmap functionality of gatekeeper).

The directories where the poolaccount and poolgroup leases are registered, the so called gridmapdir and groupmapdir, can be passed to the gatekeeper by setting the environment variables `$GRIDMAPDIR` and `$GROUPMAPDIR` to the respective directories.

The `globus.conf` file (usually residing in the `/etc` directory) contains the configuration parameters for the globus software. The gatekeeper init.d script uses this file to to configure the edg-gatekeeper. The following lines were added/modified in `/etc/globus.conf`:

```
[gatekeeper]

[...]

globus_gatekeeper=/opt/edg/sbin/edg-gatekeeper

extra_options="-lcas_etc_dir /opt/edg/etc/lcas/ -lcasmod_dir
/opt/edg/lib/lcas/ -lcas_db_file lcas.db -lcmaps_etc_dir /opt/edg/etc/lcmaps
-lcmapsmod_dir /opt/edg/lib/lcmaps -lcmaps_db_file lcmaps.db"
```

The `globus_gatekeeper=` line gives the path of the gatekeeper to be used and the `extra_options=` line the gatekeeper options to be added. The `gridmapdir` and `groupmapdir` entries give the default locations for the poolaccount and poolgroup lease administration directories.

LCFG configuration:

The `globus.conf` file can be created using the globus LCFG object contained in package `edg-lcfg-globuscfg`. The extra lines for the configuration files have to be specified in an LCFGng resource file in the way that is shown in the Computing Element resource file ComputingElement-cfg.h[23].

[23] http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/edg-release/ng_source/ComputingElement-cfg.h

3.2 Configuration of LCMAPS

The LCMAPS reads its configuration, in particular the plugins that it should load and the local site policy from the file `lcmaps.db`. An example file is shown here:

```
# LCMAPS policy file/plugin definition
#
# default path
path = /opt/edg/lib/lcmaps/modules

# Plugin definitions:
example          = "lcmaps_plugin_example.mod"
                        "Some bogus arguments"
localaccount     = "lcmaps_localaccount.mod"
                        "-gridmapfile /etc/grid-security/grid-mapfile"
poolaccount      = "lcmaps_poolaccount.mod"
                        "-gridmapfile /etc/grid-security/grid-mapfile"
                        "-gridmapdir /etc/grid-security/gridmapdir"
                        "-override_inconsistency"
posix_enf        = "lcmaps_posix_enf.mod"
                        "-maxuid 1"
                        "-maxpgid 1"
                        "-maxsgid 32"
vomsextract      = "lcmaps_voms.mod"
                        "-vomsdir /etc/grid-security/vomsdir"
                        "-certdir /etc/grid-security/certificates"
vomslocalgroup   = "lcmaps_voms_localgroup.mod"
                        "-groupmapfile /etc/grid-security/groupmapfile"
                        "-mapmin 0"
vomspoolgroup    = "lcmaps_voms_poolgroup.mod"
                        "-groupmapfile /etc/grid-security/groupmapfile"
                        "-groupmapdir /etc/grid-security/groupmapdir"
                        "-override_inconsistency"
                        "-mapmin 0"
vomspoolaccount  = "lcmaps_voms_poolaccount.mod"
                        "-gridmapfile /etc/grid-security/grid-mapfile"
                        "-gridmapdir /etc/grid-security/gridmapdir"
ldap_enf         = "lcmaps_ldap_enf.mod"
                        "-maxuid  1"
                        "-maxpgid 1"
                        "-maxsgid 32"
```

```
                         "-hostname ldap.example.org"
                         "-port 389"
                         "-require_all_groups yes"
                         "-dn_manager \"cn=Manager,dc=root\""
                         "-ldap_pw /opt/edg/etc/lcmaps/test_pw"
                         "-sb_groups \"ou=LocalGroups,dc=foobar,dc=ough\""
                         "-sb_user \"ou=LocalUsers,dc=foobar,dc=ough\""
                         "-timeout 5"

# Policies:
voms:
localaccount -> posix_enf | vomsextract
vomsextract -> vomslocalgroup
vomslocalgroup -> vomspoolgroup
vomspoolgroup -> vomspoolaccount
vomspoolaccount -> ldap_enf
ldap_enf -> posix_enf

standaard:
localaccount -> posix_enf | poolaccount
poolaccount -> posix_enf
```

The default path to the LCMAPS plugins is specified on the line starting with `path =`. On the following lines aliases are defined for the complete plugin names and their options. For a description of the plugins and the options please refer to the man pages installed with the rpms, which can also be found in apidoc. In the current release the number of aliases attached to a plugin is limited to one. If one wants to use two aliases of for example the "localaccount" plugin, each alias corresponding to different options, this is not possible, unless a physical copy is mode of the plugin. This will be corrected in the next release.

In the lines following the plugin definitions the local site policies are described. The policies follow the word ended by a colon. The policies are evaluated in order of appearance, until a policy evaluation returns a true result. In the example two policies are described:

1. default: This policy does pretty much the same as what the old gatekeeper did: check the gridmapfile with the user's DN for a local account or a poolaccount.

2. voms: This policy uses the VOMS information in the user's proxy X509 certificate. First it checks if the VO info is actually there. If so, it tries

to find local groups for this VO info or if it cannot find local groups it tries to find pool groups. If local groups *were* found, in addition it will try to find pool groups. Then it will try to find a VOMS poolaccount (based on the VO information). The next step is to try to add the gids found to the LDAP directory (`ldap_enf`) and enforce them in (`posix_enf`) the calling process (become the user).

A more elaborate description of the policy description language can be found here[24] , or as PostScript file[25] or PDF file[26].

The configuration files needed by the plugins consist of the (ordinary) `grid-mapfile` (used by the plugins `localaccount`, `poolaccount` and `vomspoolaccount`) and a new file: the `groupmapfile` (used by the plugins vomslocalgroup and vomspoolgroup). This file contains line entries for "VO-GROUP-ROLE" combinations and a corresponding local/pool account. The "VO-GROUP-ROLE" combinations in the user's proxy will be compared to the entries in the `groupmapfile` and if a match is found, a gid is added to the list of local credentials for the user. An example `groupmapfile` is shown here:

```
# Users with the exact VO-group info "/VO=fred/GROUP=fred/ROLE=husband"
# will be added to the local group "fredje1"
"/VO=fred/GROUP=fred/ROLE=husband" fredje1

# Users with the VO-group info starting with "/VO=fred/GROUP=fred"
# will be added to the allocated pool group "pool[1-9]*"
"/VO=fred/GROUP=fred*" .pool

# All users from VO wilma will be added to the allocated pool group "pool[1-9]*"
"/VO=wilma/GROUP=*" .pool
```

Note that one can use `'*'` as a wild character. The `vomspoolaccount` finds a pool account based on the VO information and on the user DN. Therefore, it looks in the `grid-mapfile` for "VO-GROUP-ROLE" combinations as is shown here:

```
# Users with the VO-group info starting with "/VO=fred/GROUP=fred"
# will receive an account from the 'fred' pool
"/VO=fred/GROUP=fred*" .fred

# All users from VO wilma will receive an account from the 'wilma' pool
"/VO=wilma/GROUP=*" .wilma
```

[24] pdl_requirements/index.html
[25] pdl_requirements.ps
[26] pdl_requirements.pdf

The leases that are maintained in the gridmapdir are, however, based on the user DN and the gids found prior to the `vomspoolaccount` call.

The gridmapdir and groupmapdir directories that are needed by the various plugins can be set in the `lcmaps.db` file or by setting the environment variables `$GRIDMAPDIR` and `$GROUPMAPDIR` to the respective directories.

LCFG configuration:
The LCMAPS policy file can also be created using the LCMAPS LCFG object contained in package `edg-lcfg-lcmaps`. The lines for the configuration files have to be specified in an LCFG resource file in the way that is shown in the Computing Element resource file ComputingElement-cfg.h[27]. One should be careful when specifying asterixes and double quotes. The `groupmapfile` will be installed by the filecopy LCFG package `edg-lcfg-filecopy`.

4 Adding LCMAPS plugins

To be done.

5 User guide

Empty.

[27] http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/edg-release/ng_source/ComputingElement-cfg.h