Implementation and Exploitation of Fair Shares in LCG

Jeff Templon

May 4, 2006

1 Introduction

Several experiments have indicated that they would like to define different classes of users within their experiment virtual organization, and define different shares of computing capacity for these groups. For example, ATLAS might assign 70% of its CPU allocation to Monte Carlo production, 20% to official analyses of these data, and 10% to 'any other' work.

In order to realize this, tests need to be done on

- mapping of 'grid stuff' like VOMS FQANs to 'site stuff' like unix GIDs
- how to convince schedulers like Maui to do the right thing with the defined shares
- how to handle publishing of the share information to the experiments, how do they take advantage of it, and how to handle dynamics (change of allocation) in a sensible fashion.

This document is essentially a log of what we (the people in the author list) have figured out.

2 Planned Approach

In order to get up and running quickly, and not start designing too much stuff before we have clearly defined the relevant issues, the following sequence of tests has been defined in a discussion of the EGEE Job Priorities Working Group.

1. come up with a prototype scheme for mapping VOMS groups onto a set of pool accounts with the right set of primary (and secondary?) unix gids.
2. come up with a prototype mapping of the relevant GIDs to the right set of Maui groups / accounts / shares
3. implement two queues for the relevant test VOs (ATLAS and CMS), one long and one short, and enable access to these queues by the proper GIDs defined above.

At this point, the experiments can submit jobs to a hard-coded list of such queues and we get feedback about the prototype fair share system and 'training material' on how to improve it.

4. publishing in the information system: use the "vomap" configuration section of the lcg-info-dynamic-scheduler to map VOMS FQANs to unix groups (or vice versa) for the VOView stuff. Requires some coding to the dynamic scheduler plugin, may also require other changes to software that assumes that all GlueCEAccessControlBaseRules look like `vo: voname`.

   This may also be a good point to consider switching over to new-style VO names like /atlas.ch ....

5. WMS matching of jobs by VOView information. Requires implementation of the scheme on the PPS plus backporting of Glue 1.2 support to the WMS on gLite 3.0.

at this point, a new class of tests is enabled, to see how accurate WMS scheduling might be with the new ERT stuff. Never really tested before.

6. Dynamics: experiments try changing fair shares and see whether this is 'easy', first via email and later via the G-PBOX.

3 Group Mapping

Start with ATLAS. Define three new account pools with corresponding primary GIDs; leave old 'atlas' account pool around, will be used for the moment whenever a job is sent without VOMS, as well as when a jobs VOMS proxy does not match any of the three defined groups.

NOTE Pool ***groups*** are being used ... this means that **each user in the relevant VOMS class will be mapped into an individual account, each of which has the same specific GID**. This is **not** like the current YAIM scheme where special VOMS classes are mapped to a single special user, in violation of accepted good security practice and official LCG/EGEE policy.

For the moment assign `atlas` as secondary group. This might be needed in order to get access to ATLAS `VO_SW_DIR` in case any files are not world-readable. If the software is world readable, then we could probably omit the secondary group.

It may be necessary to add the proper magic to the LCMAPS groupmapfile and gridmapfile; LCAS GACLs are also needed in some cases, these can be generated using the command `edg-lcas-voms2gacl`. It's not clear whether we will need them (depends on being in a transition stage between LDAP and VOMS based VOs).

3.1 Concrete Actions

Originally the following was tried:

- mapped `/VO=atlas/GROUP=/atlas/ROLE=production` to unix group `atlb` via the gridmapfile-local mechanism.

- added `atlb` accounts to `atlas` group (so this is now a secondary group for those accounts.
- added group `atlb` to ACL for `atlas` queue in torque server.

Given the way LCMAPS is configured by default, the gridmapfile mechanism did not work. So instead of the steps above, the mapping information needed to be entered directly into the LCMAPS configuration, as follows: for the file `/opt/edg/etc/lcmaps/groupmapfile` add the line

```
"/VO=atlas/GROUP=/atlas/ROLE=production"        atlb
```

and for `/opt/edg/etc/lcmaps/gridmapfile` add

```
"/VO=atlas/GROUP=/atlas/ROLE=production"        .atlb
```

The VOMS proxy was generated using the following command:

```
voms-proxy-init -voms atlas:/atlas/Role=production
```

which generated a proxy with the following attributes (displayed via the command `voms-proxy-info -all`:

```
bosui:~> voms-proxy-info -all
subject   : /O=dutchgrid/O=users/O=nikhef/CN=Jeffrey Templon/CN=proxy
issuer    : /O=dutchgrid/O=users/O=nikhef/CN=Jeffrey Templon
identity  : /O=dutchgrid/O=users/O=nikhef/CN=Jeffrey Templon
type      : proxy
strength  : 512 bits
path      : /tmp/x509up_u500
timeleft  : 11:59:46
VO        : atlas
subject   : /O=dutchgrid/O=users/O=nikhef/CN=Jeffrey Templon
issuer    : /C=CH/O=CERN/OU=GRID/CN=host/lcg-voms.cern.ch
attribute : /atlas/Role=production/Capability=NULL
attribute : /atlas/Role=NULL/Capability=NULL
attribute : 22/Role=22/Capability=22
attribute : 45/Role=45/Capability=45
timeleft  : 11:59:45
```

This combination results in my proxy being mapped to the new secondary ATLAS group (and associated pool accounts), meaning that this group could now be used to arrange for `ROLE=production` users to receive a different fair share within ATLAS.

3.2 Local Notes

3.2.1 VOMSES

The VOMS proxy init did not work intially due to the absence of the VOMS server information. I fixed this by making a file `.edg/vomses` in my home directory and adding the line

```
"atlas" "tbed0152.cern.ch" "15001" \
   "/C=CH/O=CERN/OU=GRID/CN=host/lcg-voms.cern.ch" "atlas"
```

to it. This line has been split at the \ character for formatting convenience, but it is a single unbroken line in the actual file.

3.2.2 Interaction With LDAP

The command to do the secondary group modifications in the LDAP directory is quite tricky. From my laptop using ssh tunneling to the real farmnet server, the command is

```
ldapmodify -H ldaps://localhost:1636/ -W -Z -x -D \
   "cn=Jeff Templon,ou=Managers,dc=farmnet,dc=nikhef,dc=nl" \
   -f tmp.ldif
```

3.2.3 Pushing Profiles

Procedure:

1. edit profiles in private copy of CVS
2. checkin
3. login to `ndpfmgr` account on quattor server
4. `cvs upd` in appropriate directory
5. `pushxprof -f prd -p tbn20`

4 Maui Shares

Map the primary groups to Maui `GROUP`s; bundle these groups together into `ACCOUNT`s that model VOs. Need to play with the fair-share weighting; should be that

$$\text{ACCOUNTWEIGHT} > \text{GROUPWEIGHT} > \text{USERWEIGHT}$$

Reasoning is that it is much more important that VO shares (relative usage of LHCb vs ATLAS) are balanced than it is that the three ATLAS subgroups are in the proper proportion. So the `ACCOUNT` (or VO) weighs heavier. Otherwise we could have the situation that the scheduler allows ATLAS production jobs to run in an attempt to get the FS ratio production/analysis correct within ATLAS, even though this causes ATLAS to take more than their fair share relative to LHCb. A similar argument applies for the relationship `GROUPWEIGHT` vs `USERWEIGHT`. The factors by which these should differ are not yet determined, experience is needed.

There are similar issues with `QUEUETIMEWEIGHT` and `XFACTOR` that have to do with how long a job is sitting in the queue and how long the job is expected to take once it's running. Again we need experience to set these correctly.

5 G-PBOX Setup

My understanding (conversation with Vincenzo, thats why hes in the author list) is that we do the following. We define symbolic names and assign shares to them; we also define unix groups assigned to these symbolic names, but this is strictly site internal.

We tell the VO the following:

```
ATLAS-share1 0.50
ATLAS-share2 0.30
ATLAS-share3 0.20
```

Internal to the site we have another mapping like

```
ATLAS-share1 atlgx
ATLAS-share2 atlgy
ATLAS-share3 atlgz
```

What is transmitted via G-PBOX is a mapping like

```
ATLAS-share1 /atlas.ch/prod
ATLAS-share2 /atlas.ch/anal
ATLAS-share3 /atlas.ch/bobos
```

The site admin has to accept the policy which should prevent mistakes like

```
ATLAS-share1 /evil-vo.za/dos
```

from happening.

This would not work right now since the queue mappings (access control vs queue name) are static, but we can have the info provider query the G-PBOX to find out what the vo mapping should be, the info provider already has a vomap construct.

Our accounting backend also has a vomap construct with which we can map all three shares onto VO atlas when reporting to APEL.