

Extending DPM with DMLite

Alejandro Álvarez Ayllón
on behalf of the LCGM development team



Overview

- Installing DMLite on an existing DPM instance
- Installing DMLite-based HTTP server
- Improving performance
 - Enabling MySQL plugin
 - Enabling Memcache
- Adding new pool types
 - Enabling Hadoop
 - Enabling S3
- Installing DMLite-based SRM
 - Accessing a file hosted in Hadoop through SRM+HTTP



Installing DMLite

- We are going to use DMLite 0.6.0 from our development repository
 - Will be released soon, though
- First, we need to install the libraries
 - `yum install dmlite-libs`
- And then, at least one plugin
 - `yum install dmlite-plugins-adapter`



Configuring DMLite

- The files we are interested in, are under `/etc/dmlite.conf.d/`
- There, we choose which plugins we want to enable, and in which order
 - Plus their parameters
- Installing enables by default
 - But we need to set a couple of parameters



Installing WebDAV

- We are going to use LCGDM-DAV 0.12.0
 - `yum install lcgdm-dav-server`
- Defaults should be good enough
 - But we probably will need to change some settings in `ssl.conf`
- Now, we must restart Apache
 - `service httpd restart`
- And we should be able to access our server



Improving performance

- Installing the MySQL plugin, we will go directly to the database, instead of passing by the daemons
 - `yum install dmlite-plugins-mysql`
- And Memcache will keep a cache in memory of the namespace
 - `yum install dmlite-plugins-memcache`
 - I assume there is a running memcached server already running
- As mentioned before, they are enabled by default
 - A restart of the affected daemons is needed, though
- But we need to configure them
 - `/etc/dmlite.conf.d/mysql.conf`
 - `/etc/dmlite.conf.d/memcache.conf`
- And disable the Adapter namespace plugin, leaving Pool handling
 - `/etc/dmlite.conf.d/adapter.conf`



Adding new Pool types

- We are going to add a Hadoop pool to our existing DPM!
 - `yum install dmlite-plugins-hdfs jdk`
- Now, we edit `/etc/dmlite.conf.d/hdfs.conf` to disable the namespace handling, and leave only Pool handling
- And we obviously need a Hadoop pool to be created
- Now, we restart `httpd`



Adding new Pool types

- We need to enable HTTP access in our Hadoop data servers
 - lcgdm-dav-server + dmlite-plugins-hadoop
- And configure a common secret for the token generation and validation



Adding new Pool types

- We will disable the filesystem pool, to force any new file to go to Hadoop
- Using CURL, we do a PUT
- And now we have Hadoop files under the same roof as our previous native pool!



Installing DMLite-based SRM

- Let's make our Hadoop files part of the Grid!
- If you want to provide access to them through SRM, dmlite-srm is needed
 - yum install dmlite-srm
- No need for extra configuration, since the same configuration for DAV works for the SRM daemon
- Now, we do a SRM request asking for an HTTP TURL...
- And it just works!

