LFC: The LCG File Catalog

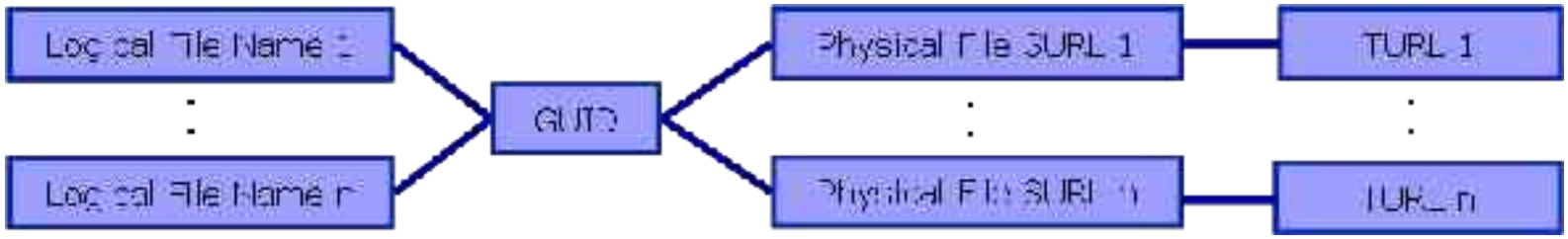
Tony Calanducci
NA3: User Training and Induction. INFN CT
Retreat between GILDA and ESR VO on gLite
Bratislava, 27-30.06.2005

Introduction

- **User and programs produce and require data**
- **Data may be stored in Grid datasets (files)**
  - Located in Storage Elements (SEs)
  - Several replicas of one file in different sites
  - Accessible by Grid users and applications from “anywhere”
  - Locatable by the WMS (data requirements in JDL)
- **Also…**
  - Resource Broker can send (small amounts of) data to/from jobs: Input and Output Sandbox
  - Data may be copied from/to local filesystems (WNs, UIs) to the Grid

Name conventions

- **Logical File Name (LFN)**
  - An alias created by a user to refer to some item of data, e.g. "lfn:cms/20030203/run2/track1"
- **Globally Unique Identifier (GUID)**
  - A non-human-readable unique identifier for an item of data, e.g. "guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
- **Site URL (SURL) (or Physical File Name (PFN) or Site FN)**
  - The location of an actual piece of data on a storage system, e.g. "srm://pcrd24.cern.ch/flatfiles/cms/output10_1" (SRM) "sfn://lxshare0209.cern.ch/data/alice/ntuples.dat" (Classic SE)
- **Transport URL (TURL)**
  - Temporary locator of a replica + access protocol: understood by a SE, e.g. "rfio://lxshare0209.cern.ch//data/alice/ntuples.dat"



File Catalogs in LCG

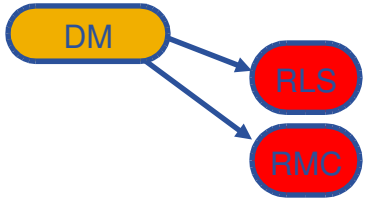
- **File catalogs in LCG:**
  - They keep track of the location of copies (replicas) of Grid files
  - The DM tools and APIs and the WMS interact with them
- **EDG's Replica Location Service (RLS)**
  - Catalogs in use in LCG-2
  - Replica Metadata Catalog (**RMC**) + Local Replica Catalog (**LRC**)
  - Some performance problems detected during Data Challenges
- **New LCG File Catalog (LFC)**
  - In production in next LCG release; deployment in January 2005
  - Coexistence with RLS; migration tools provided:

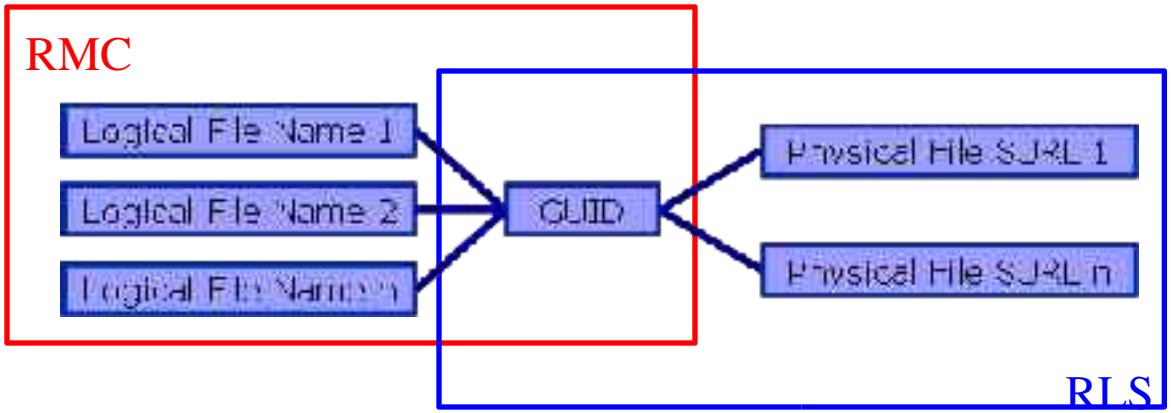
http://goc.grid.sinica.edu.tw/gocwiki/How_to_migrate_the_RLS_entries_into_the_LCG_File_Catalog_%28LFC%29

  - Accessible by defining: $LCG_CATALOG_TYPE=lfc and $LFC_HOST
  - Better performance and scalability
  - Provides new features: security, hierarchical namespace, transactions...

The RLS

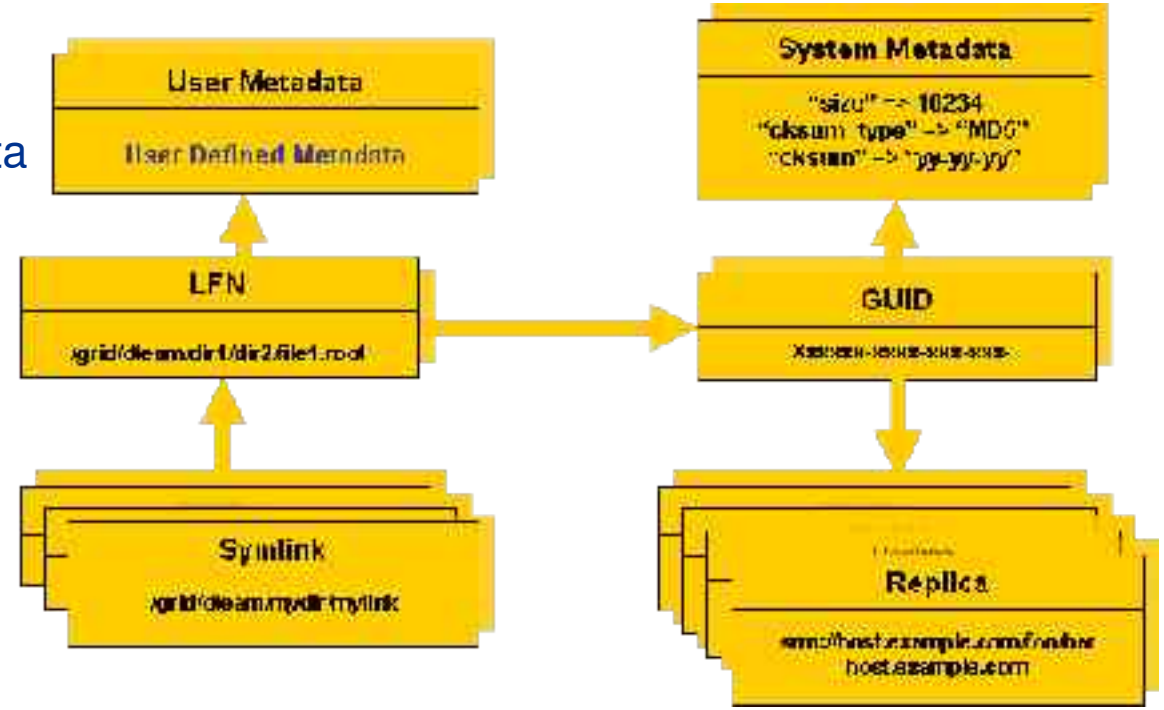
- **RMC:**
  - Stores LFN-GUID mappings
  - Accessible by edg-rmc CLI + API
- **RLS:**
  - Stores GUID-SURL mappings
  - Accessible by edg-lrc CLI + API
- **Main weaknesses:**
  - Insecure (anyone can delete catalog entries)
  - Bad performance (java clients…)





The LFC

- **One single catalog**
- **LFN acts as main key in the database. It has:**
  - Symbolic links to it (additional LFNs)
  - Unique Identifier (GUID)
  - System metadata
  - Information on replicas
  - One field of user metadata



The LFC (II)

- **Fixes EDG catalogs performance and scalability problems**
  - Cursors for large queries
  - Timeouts and retries from the client
- **Provides more features than the EDG Catalogs**
  - User exposed transaction API (+ auto rollback on failure)
  - Hierarchical namespace and namespace operations (for LFNs)
  - Integrated GSI Authentication + Authorization
    ➔ Mapping with local UID/GID problem being solved (pool of accounts)
  - Access Control Lists (Unix Permissions and POSIX ACLs)
  - Checksums
- **New features will be added (requests welcome!)**
  - Integration with VOMS, Fireman
  - POOL Integration is in progress
  - Sessions
  - Bulk operations

Setting up the LFC client

- **Very simple installation (also included in YAIM):**
  - Install single RPM: in WN, UI, RB
  - Specify the host of the server (required for the moment!)
    > export LFC_HOST=<LFC_server_hostname>
  - Test the client

- **Using lcg_utils and GFAL:**
  - Define the catalog to use:
    $LCG_CATALOG_TYPE=lfc
  - Define the server hostname
    - The LFC server must be published in the BDII ($LCG_GFAL_INFOSYS)
    - Or use environmental variable: $*LFC_HOST=<LFC_server_hostname>*

- **Env variable: LFC_HOME**
  - Can be set to use relative LFNs
  - *LFC_HOME=/grid/gilda/myDir* ➔ */grid/gilda/myDir/myFile* becomes *myFile*

LFC Troubleshooting

- **Environment variables:**
  - *$LFC_HOST* not set and catalog not published in BDII
    - ➔ lfc-ls… *send2nsd: NS009 - fatal configuration error: Host unknown: …*
    - ➔ lcg-lr… return nothing (or *"No such file or directory"*)
  - *$LCG_CATALOG_TYPE* wrongly or not set (default "edg")
    - Files that appear and disappear
      - ➔ lcg-lr… return nothing (or *"No such file or directory"*)
    - Unsupported VOs
      - ➔ lcg-lr… return "*Invalid argument*" (and "*LRC, RMC endpoint not found*")
- **Other configuration errors**
  - VO directory not defined by root in the LFC hierarchy
    - Unsupported VOs
      - ➔ lcg-lr… return "*Invalid argument*" (and "*LRC, RMC endpoint not found*")
- **Attention!**
  - lcg_utils do not create directories automatically (feature)
    - ➔ explicit use of *lfc-mkdir* required (as user)
    - ➔ *$LFC_HOST* must be set

LFC Interfaces (II)

- **LFC client commands**
  - Provide administrative functionality
  - Unix-like
  - LFNs seen as a Unix filesystem (/grid/<VO>/ …)
- **LFC C API**
  - Alternative way to administer the catalog
  - Python wrapper provided
- **Integration with GFAL and lcg_util APIs complete**
  - ➔ lcg-utils access the catalog in a transparent way
- **Integration with the WMS completed**
  - The RB can locate Grid files: allows for data based match-making
  - Not yet tested in production

Data Management CLIs & APIs

- **lcg_utils: lcg-* commands + lcg_* API calls**
  - Provide (all) the functionality needed by the LCG user
  - Transparent interaction with file catalogs and storage interfaces when needed
  - Abstraction from technology of specific implementations
- **Grid File Access Library (GFAL): API**
  - Adds file I/O and explicit catalog interaction functionality
  - Still provides the abstraction and transparency of lcg_utils
- **edg-gridftp tools: CLI**
  - Complete the lcg_utils with low level GridFTP operations
  - Functionality available as API in GFAL
  - May be generalized as lcg-* commands

Data Management CLIs & APIs (II)

- **All-purpose CLIs and APIs for EDG and LCG**
- **File & replica management**
  - edg-rm
- **Catalog interaction (only for RLS catalogs)**
  - edg-lrc
  - edg-rmc
- **Use discouraged**
  - Worst performance (slower) than lcg_utils
  - New features added only to lcg_utils
  - Currently they are just a wrapper on GFAL anyway
  - The catalog commands do not interact with LFC

lcg-utils commands

Replica Management

lcg-cp	Copies a grid file to a local destination
lcg-cr	Copies a file to a SE and registers the file in the catalog
lcg-del	Delete one file
lcg-rep	Replication between SEs and registration of the replica
lcg-gt	Gets the TURL for a given SURL and transfer protocol
lcg-sd	Sets file status to “Done” for a given SURL in a SRM request

File Catalog Interaction

lcg-aa	Add an alias in LFC for a given GUID
lcg-ra	Remove an alias in LFC for a given GUID
lcg-rf	Registers in LFC a file placed in a SE
lcg-uf	Unregisters in LFC a file placed in a SE
lcg-la	Lists the alias for a given SURL, GUID or LFN
lcg-lg	Get the GUID for a given LFN or SURL
lcg-lr	Lists the replicas for a given GUID, SURL or LFN

LFC C API

Low level methods (many POSIX-like):

lfc_access
lfc_aborttrans
lfc_addreplica
lfc_apiinit
lfc_chclass
lfc_chdir
lfc_chmod
lfc_chown
lfc_closedir
lfc_creat
lfc_delcomment
lfc_delete
lfc_deleteclass
lfc_delreplica
lfc_endtrans
lfc_enterclass
lfc_errmsg
lfc_getacl
lfc_getcomment
lfc_getcwd
lfc_getpath
lfc_lchown
lfc_listclass
lfc_listlinks
lfc_listreplica
lfc_lstat
lfc_mkdir
lfc_modifyclass
lfc_opendir
lfc_queryclass
lfc_readdir
lfc_readlink
lfc_rename
lfc_rewind
lfc_rmdir
lfc_selectsrvr
lfc_setacl
lfc_setatime
lfc_setcomment
lfc_seterrbuf
lfc_setfsize
lfc_starttrans
lfc_stat
lfc_symlink
lfc_umask
lfc_undelete
lfc_unlink
lfc_utime
send2lfc

LFC commands

Summary of the LFC Catalog commands

lfc-chmod	Change access mode of the LFC file/directory
lfc-chown	Change owner and group of the LFC file-directory
lfc-delcomment	Delete the comment associated with the file/directory
lfc-getacl	Get file/directory access control lists
lfc-ln	Make a symbolic link to a file/directory
lfc-ls	List file/directory entries in a directory
lfc-mkdir	Create a directory
lfc-rename	Rename a file/directory
lfc-rm	Remove a file/directory
lfc-setacl	Set file/directory access control lists
lfc-setcomment	Add/replace a comment

lfc-ls

Listing the entries of a LFC directory

lfc-ls *[-cdiLlRTu] [--class] [--comment] [--deleted] [--display_side] [--ds] path…*

where *path* specifies the LFN pathname (mandatory)

- Remember that **LFC has a directory tree structure**
- /grid/<VO_name>/<you create it>

- All members of a VO have read-write permissions under their directory
- You can set LFC_HOME to use relative paths

> lfc-ls /grid/gilda/antonio
> export LFC_HOME=/grid/gilda
> lfc-ls -l antonio
> lfc-ls -l -R /grid

-l : long listing
-R : list the contents of directories recursively: Don't use it!

Creating directories in the LFC

lfc-mkdir [-m mode] [-p] path...

- Where *path* specifies the LFC pathname
- Remember that while registering a new file (using lcg-cr, for example) the corresponding destination directory must be created in the catalog beforehand.
- Examples:

  > lfc-mkdir /grid/gilda/antonio/demo

  You can just check the directory with:

  > lfc-ls -l /grid/gilda/antonio

  drwxr-xrwx 0 19122 1077 0 Jun 14 11:36 demo

lcg-utils integration

Let us copy and register a file using lcg-utils

> *export LCG_CATALOG_TYPE=lfc*

> *lcg-infosites --vo gilda se*

```
**************************************************************
These are the related data for gilda: (in terms of SE)
**************************************************************
Avail Space(Kb) Used Space(Kb)  Type    SEs
----------------------------------------------------------
                . . .
28429708        2730072         disk    testbed005.cnaf.infn.it
70014168        4132600         disk    gilda-se-01.pd.infn.it
388209224       254568          disk    grid-se.bio.dist.unige.it
```

> *lcg-cr --vo gilda -l demo/test -d gilda-se-01.pd.infn.it file:`pwd`/test*

guid:0c3994b0-634f-4401-9434-e83a8e4bf14e

> **lcg-lr --vo gilda lfn:demo/test**

sfn://gilda-se-01.pd.infn.it/shared/gilda/generated/2005-06-14/file567eb5f3-17d5-4e0f-a1ca-a8caef3d4d08

> **lfc-ls -l demo**

-rwxrwxrwx 1 19122 1077 28 Jun 14 11:39 test

lfc-ln

Creating a symbolic link

lfc-ln -s file linkname

lfc-ln -s directory linkname

Create a link to the specified *file* or *directory* with *linkname*

– *Examples:*

> lfc-ln -s /grid/gilda/antonio/demo/test /grid/gilda/antonio/aLink

Let's check the link using lfc-ls with long listing (-l):

> lfc-ls -l

lrwxrwxrwx 1 19122 1077 0 Jun 14 11:58 aLink -> /grid/gilda/antonio/demo/test

drwxr-xrwx 1 19122 1077 0 Jun 14 11:39 demo

lfc-setcomment

Adding/deleting metadata information

lfc-setcomment *path comment*

lfc-delcomment *path*

lfc-setcomment adds/replaces a *comment* associated with a file/directory in the LFC Catalog

lfc-delcomment deletes a comment previously added

- This is the only metadata (one field) supported by the catalog
- Examples:

  > lfc-setcomment demo/test "nice file"

- Let's see what happened:

  > lfc-ls --comment /grid/gilda/antonio/demo/test

  /grid/gilda/antonio/demo/test Nice file

Bibliography

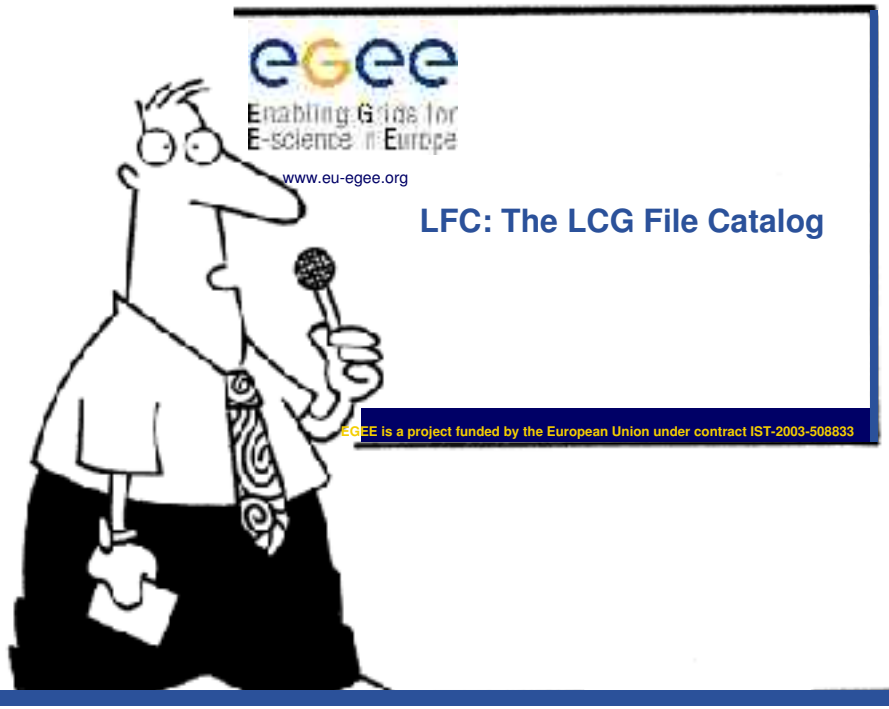
- Information on the file catalogs
  - LFC, gfal, lcg-utils:

    "Evolution of LCG-2 Data Management (J-P Baud, J. Casey)"

    http://indico.cern.ch/contributionDisplay.py?contribId=278&sessionId=7&confId=0

  - LFC installation, administration, migration from RLS:
    - Wiki entries indicated through the presentation:
      - *http://goc.grid.sinica.edu.tw/gocwiki/How_to_set_up_an_LFC_service*
      - *http://goc.grid.sinica.edu.tw/gocwiki/How_to_migrate_the_RLS_entries_into_the_LC*
  - LFC contacts:
    - Jean-Philippe.Baud@cern.ch
    - Sophie.Lemaitre@cern.ch

The End



Hope you enjoy this lecture.
Thank you for attending !