



*The Abdus Salam
International Centre for Theoretical Physics*



1967-35

Advanced School in High Performance and GRID Computing

3 - 14 November 2008

Grid tutorial hands-on session on data management and advanced job submission

TAFFONI Giuliano
*Osservatorio Astronomico di Trieste/INAF
Via G.B. Tiepolo 11
34131 Trieste
ITALY*

Architecture of gLite Data Management System

Dr. Giuliano Taffoni
INAF – Information Technology Division

What are we talking about

- Grid Data Management Challenge
- Storage Elements and SRM
- File Catalogs and DM tools
- File Transfer Service

DM challenge

Heterogeneity

Data stored on different storage systems using different access technologies

Distribution

Data are stored in different locations
Need to keep track where data is stored

Data Description

Data are stored as files: need a way to describe files and locate them according to their contents

Problems and solutions

- Common interface to storage resources:
 - SRM
- Keep track where data is stored
 - File catalogue
- Scheduled, reliable file transfer
 - FTS
- Describe file content
 - Metadata service

Assumptions

- The lowest granularity of the data is on the file level (we deal with files rather than data objects or tables);
- Files:
 - Mostly, write once, read many
 - 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)
 - Files may be copied from/to local filesystems (WNs, UIs) to the Grid (SEs)

Storage Element

- The Storage Element is the service which allow a user or an application to store data for future retrieval
- Manage local storage (disks) and interface to Mass Storage Systems(tapes) like: Castor, HPSS
- Be able to manage different storage systems uniformly and transparently for the user (providing an SRM interface)
- Support basic file transfer protocols
 - GridFTP mandatory
 - Others if available (https, ftp, etc)
- Support a native I/O (remote file) access protocol
 - POSIX (like) I/O client library for direct access of data (GFAL)

Storage Resource Manager

- Data are stored on disk pool servers or Mass Storage Systems
- Storage resource management needs to take into account
 - Transparent access to files (migration to/from disk pool)
 - File pinning
 - Space reservation
 - File status notification
 - Life time management
- The SRM (Storage Resource Manager) takes care of all these details
 - The SRM is a single interface that takes care of local storage interaction and provides a Grid interface to the outside world
- In gLite, interactions with the SRM is hidden by higher level services (DM tools and APIs)

gLite SE types

- Classic SE:
 - GridFTP server
 - Insecure RFIO daemon (rfiod) – only LAN limited file access
 - Single disk or disk array
 - No quota management
 - Does not support the SRM interface

gLite SE types

- Disk pool managers (dCache and gLite DPM)
 - manage distributed storage servers in a centralized way
 - Physical disks or arrays are combined into a common (virtual) file system
 - Disks can be dynamically added to the pool
 - GridFTP server
 - Secure remote access protocols (gsidcap for dCache, gsirfio for DPM)
 - SRM interface

gLite SE types

- STORM
 - SRM interface
 - Interface over POSIX filesystem
 - POSIX ACLs based on LFC ACLs
 - Temporary ACLs
 - GridFTP access
 - Secure remote access protocols

The data management service

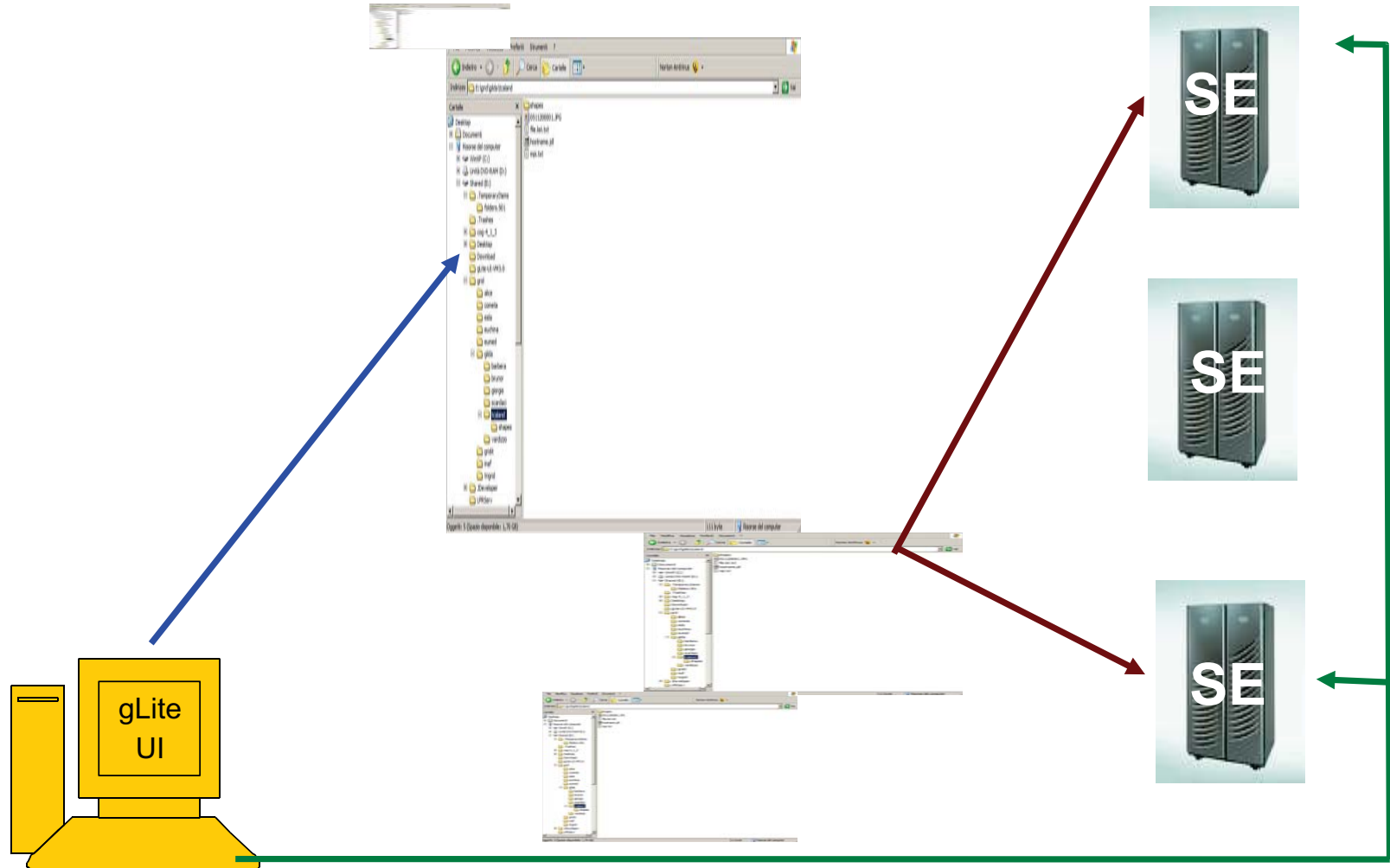
- Data transfer is done by a number of protocols (gsiftp, rfio, file, etc`)
- Usage of a central **File catalogue**



By using high level data management tools which enable transparency of the transport layer details (protocols) , storage location and the internal structure of the SE's

The SE is a “black box”

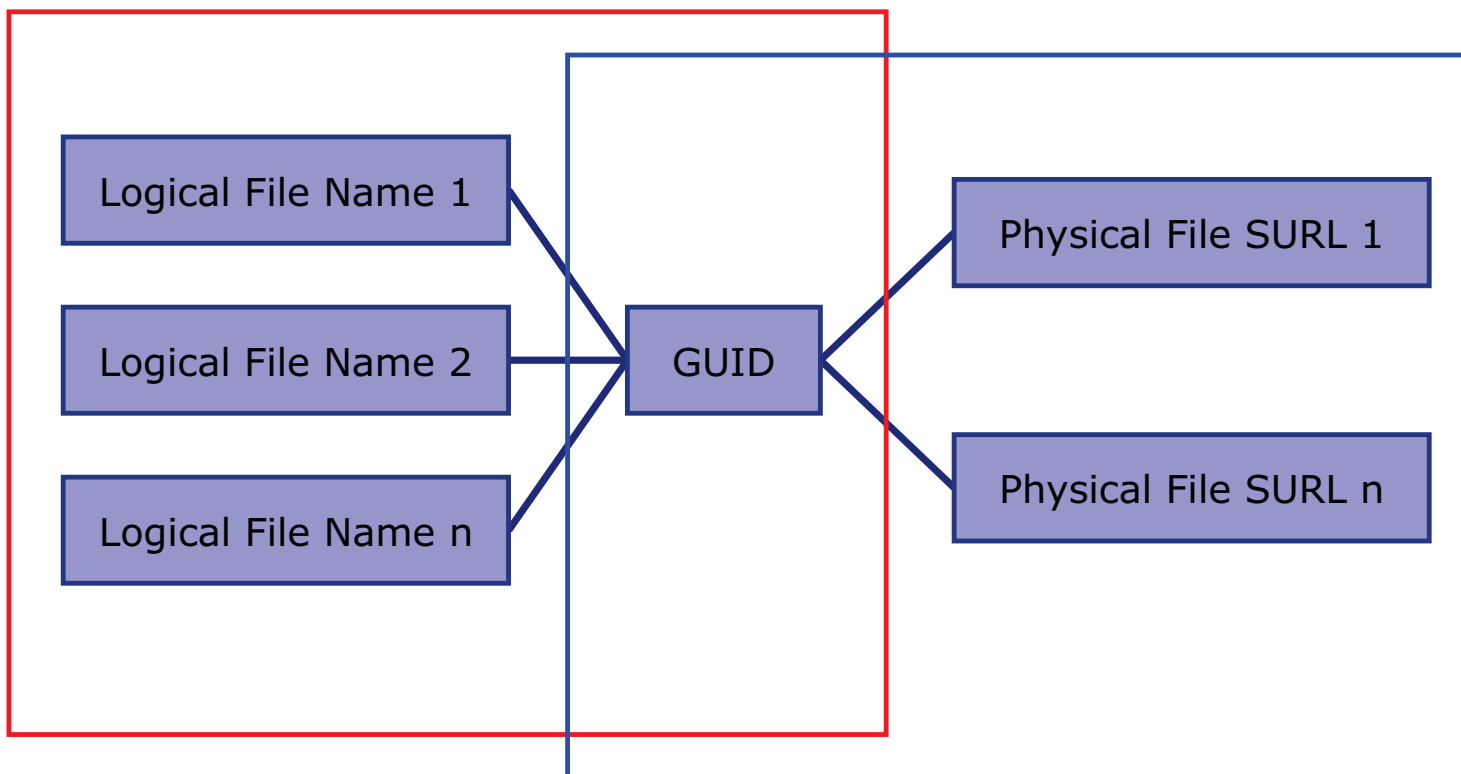
File Catalog



File Naming Conventions

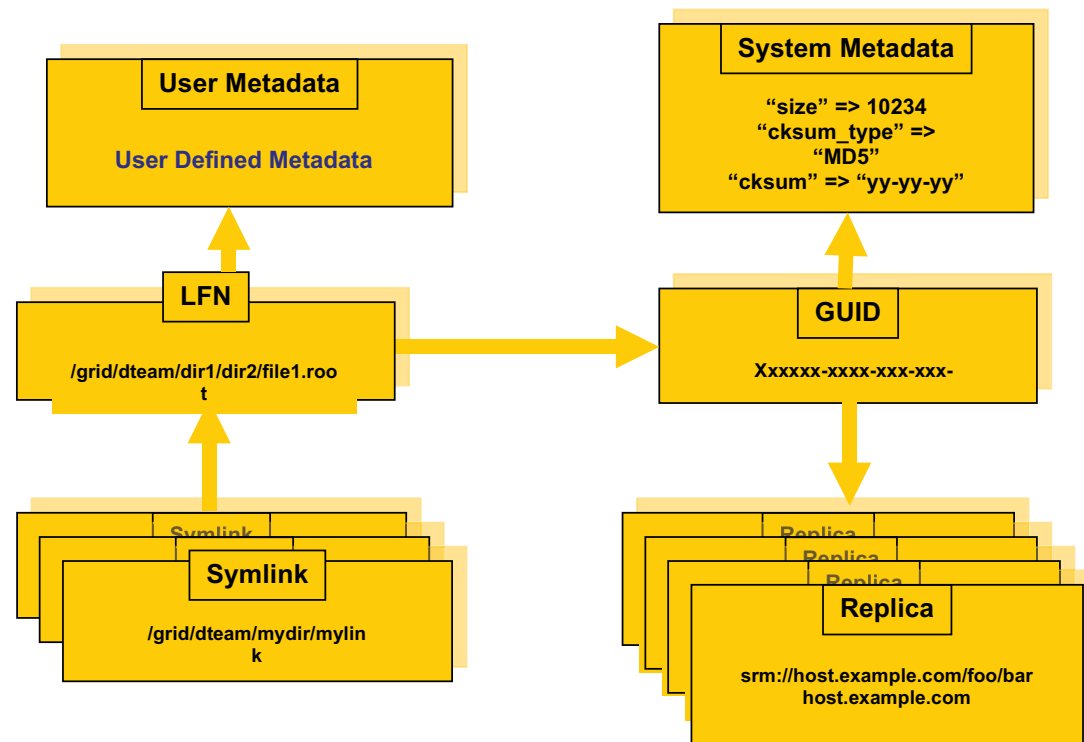
- **Logical File Name (LFN)**
 - An alias created by a user to refer to some item of data, e.g.
 - “lfn:/grid/ gilda/20030203/run2/track1”
- **Globally Unique Identifier (GUID)**
 - A non-human-readable unique identifier for an item of data, e.g.
 - “guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”
- **Site 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://grid009.ct.infn.it/dpm/ct.infn.it/gilda/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”

Naming schema



The LFC

- It keeps track of the location of copies (replicas) of Grid files
- It has:
 - Symbolic links to it (additional LFNs)
 - Unique Identifier (GUID)
 - System metadata
 - Information on replicas
 - One field of user metadata



DM commands

- **lcg-cp** Copies a Grid file to a local destination
- **lcg-cr** Copies a file to a SE and registers the file in the LRC
- **lcg-del** Deletes one file (either one replica or all replicas)
- **lcg-rep** Copies a file from SE to SE and registers it in the LRC

DM commands

- **lcg-lg** Gets the guid for a given lfn or surl
- **lcg-aa** Adds an alias in RMC for a given guid
- **lcg-la** Lists the aliases for a given LFN, GUID or SURL
- **lcg-gt** Gets the turl for a given surl and transfer protocol

DM commands

- **lfc-ls** List file/directory entries in a directory.
- **lfc-mkdir** Create directory.
- **lfc-rename** Rename a file/directory.
- **lfc-rm** Remove an empty directory.
- **lfc-chmod** Change access mode of a file/directory
- **lfc-chown** Change owner and group of a file/directory

Grid File Access Library

Interactions with SE require some components:

- File catalog services to locate replicas
- SRM
- File access mechanism to access files from the SE

GFAL does all this tasks for you:

- Hides all these operations
- Presents a POSIX interface for the I/O operations
 - Single shared library in threaded and unthreaded versions
`libgfal.so`, `libgfal_pthr.so`
 - Single header file: `gfal_api.h`
- User can create all commands needed for storage management
- It offers as well an interface to SRM

GFAL File I/O API

```
int gfal_access (const char *path, int amode);  
int gfal_chmod (const char *path, mode_t mode);  
int gfal_close (int fd);  
int gfal_creat (const char *filename, mode_t mode);  
off_t gfal_lseek (int fd, off_t offset, int whence);  
int gfal_open (const char * filename, int flags, mode_t mode);  
ssize_t gfal_read (int fd, void *buf, size_t size);  
int gfal_rename (const char *old_name, const char *new_name);  
ssize_t gfal_setfilchg (int, const void *, size_t);  
int gfal_stat (const char *filename, struct stat *statbuf);  
int gfal_unlink (const char *filename);  
ssize_t gfal_write (int fd, const void *buf, size_t size);
```

GFAL Catalog API

int **create_alias** (const char *guid, const char *lfn, long long size)

int **guid_exists** (const char *guid)

char ***guidforpfn** (const char *surl)

char ***guidfromlfn** (const char *lfn)

char ****lfnsforguid** (const char *guid)

int **register_alias** (const char *guid, const char *lfn)

int **register_pfn** (const char *guid, const char *surl)

int **setfilesize** (const char *surl, long long size)

char ***surlfromguid** (const char *guid)

char ****surlsfromguid** (const char *guid)

int **unregister_alias** (const char *guid, const char *lfn)

int **unregister_pfn** (const char *guid, const char *surl)

Now it's time for exercises