int.eu.grid

http://www.interactive-grid.eu

LFC Explained

LFC: **L**ogical **F**ile **C**atalog

- Logical file management
- Replica management
- File Access

Overview

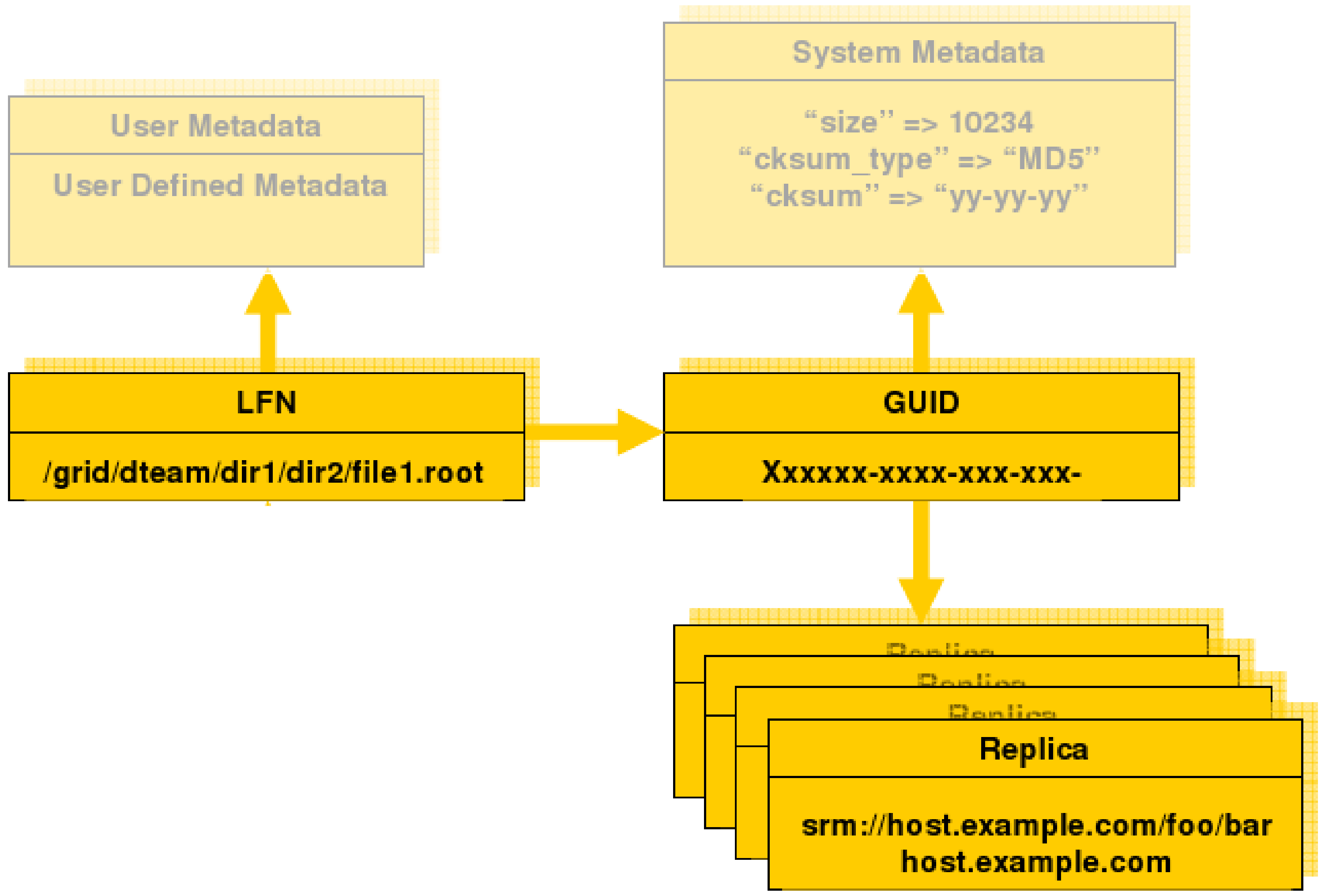
- LFC: **L**ogical **F**ile **C**atalog:
  - Map LFN <=> GUID <=> SURL
    - SURL: Actual storage URL (gsiftp://fzk.de/file.txt)
    - LFN: Logical File Name (lfn:/grid/iusct/file.txt)
    - GUID: Globally Unique ID (guid:9cd7ceb1-2b77-4b73-9262-43b9f3ecc46c)
  - Manage Access
  - Organise LFNs in a directory structure
- One - #(VOs) LFC servers per grid

LFC Commands

- The ususal stuff
  - lfc-chgrp
  - lfc-chmod
  - lfc-chown
  - lfc-ln
  - lfc-ls
  - lfc-mkdir
  - lfc-rename
  - lfc-rm
- And a lot more...
  - lfc-setcomment
  - lfc-delcomment
  - lfc-enterusrmap
  - lfc-entergrpmap
  - lfc-[modify|rm]*map
  - lfc-setacl
  - lfc-getacl
  - lfc-ping

Replica Management (LCG-RM)

- LCG Replica Manager:
  - Intermediate layer between
    - Data storage (SURLs)
    - Logical files (LFNs)
- Features:
  - Copy local file to grid storage (create GUID + SURL)
  - Register new files with LFC (create GUID+LFN)
  - Replicate files to other SEs and keep track (new SURL)
- Commands:
  - lcg-aa
  - lcg-cp
  - lcg-cr
  - lcg-del
  - lcg-fetch
  - lcg-gt
  - lcg-la
  - lcg-lg
  - lcg-lr
  - lcg-ra
  - lcg-rep
  - lcg-uf

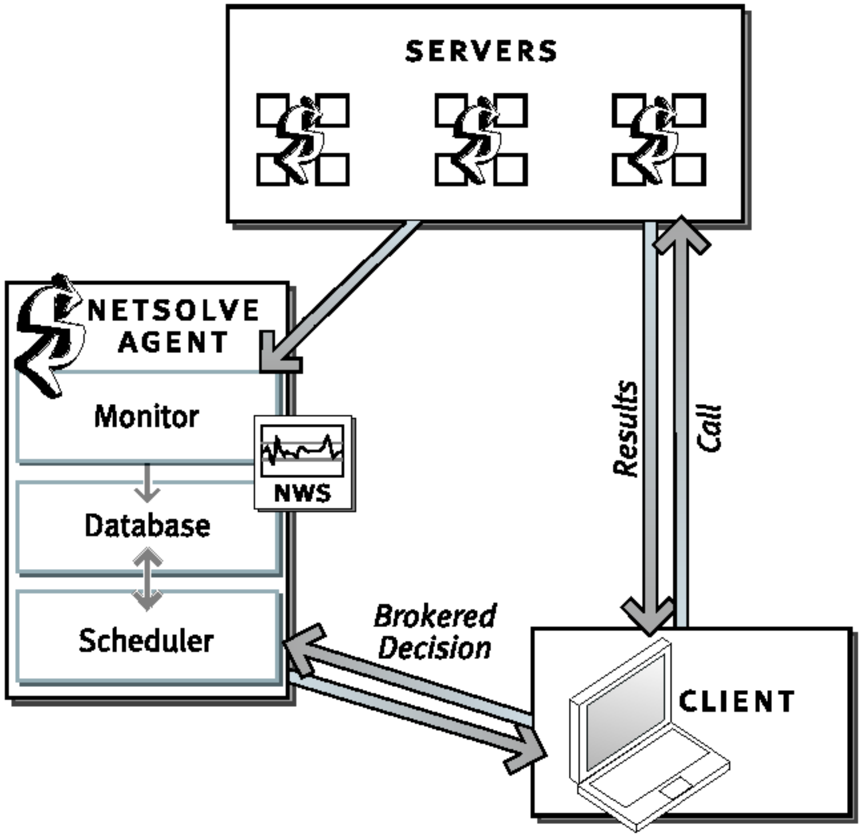

User Metadata
User Defined Metadata
System Metadata
"size" => 10234
"cksum_type" => "MD5"
"cksum" => "yy-yy-yy"
LFN
/grid/dteam/dir1/dir2/file1.root
GUID
Xxxxxx-xxxx-xxx-xxx-
Replica
Replica
Replica
Replica
srm://host.example.com/foo/bar
host.example.com

Grid File Access Library (GFAL)

- GFAL Features:
  - POSIX like file access:
  - gfal_open
  - gfal_read
  - gfal_write
  - gfal_lseek
  - gfal_close
  - and more (gfal_access, gfal_chmod, gfal_closedir, gfal_creat, gfal_mkdir, gfal_opendir, gfal_readdir, gfal_rename, gfal_rmdir, gfal_stat, gfal_unlink)
  - sdf
- lcg-* tools are implemented, using GFAL

GridSolve

- “Client – Agent – Server” architicture:
  - User connects the client to agent
  - Servers (many) report abilities to agent
  - Agent tells client which server to use next



Why did we choose gridsolve

- **User Interface**
- API-style
  - Interface for C, Fortran, Matlab, for remote method invocation (**RMI**):

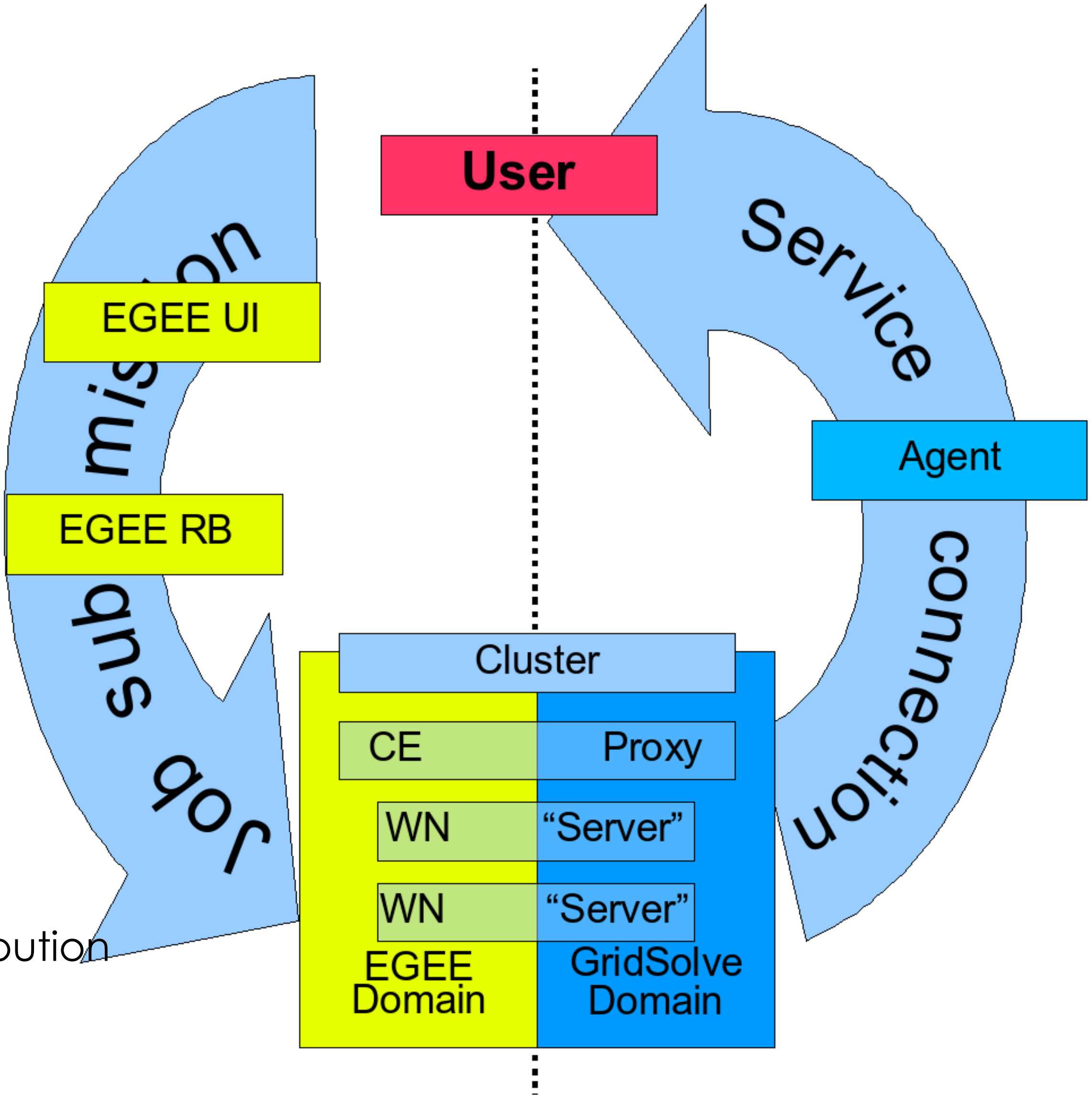
```
result = analysis (x, y);
result = gs_call ('analysis', x, y);
```

- "analysis"
  - is a C, or Fortran function
  - IDL code generator compiles a "problem"
  - Problems are deployed on servers
- Asyncronous calls + "call farming" available

GridSolve on top of int.eu.grid

- Submission of Servers via Migrating Desktop
  - 1-200 times
- Agent on outside host
- Connectivity provided by a proxy host
- Startup of local Client Environment also via MD:
  - Client and Matlab
  - Connect to agent
  - Connect (via proxy) to resources

- Example
  - Fractal calculation
    - Resolution => Data
    - # Iterations => CPU
    - # Resources => Distribution

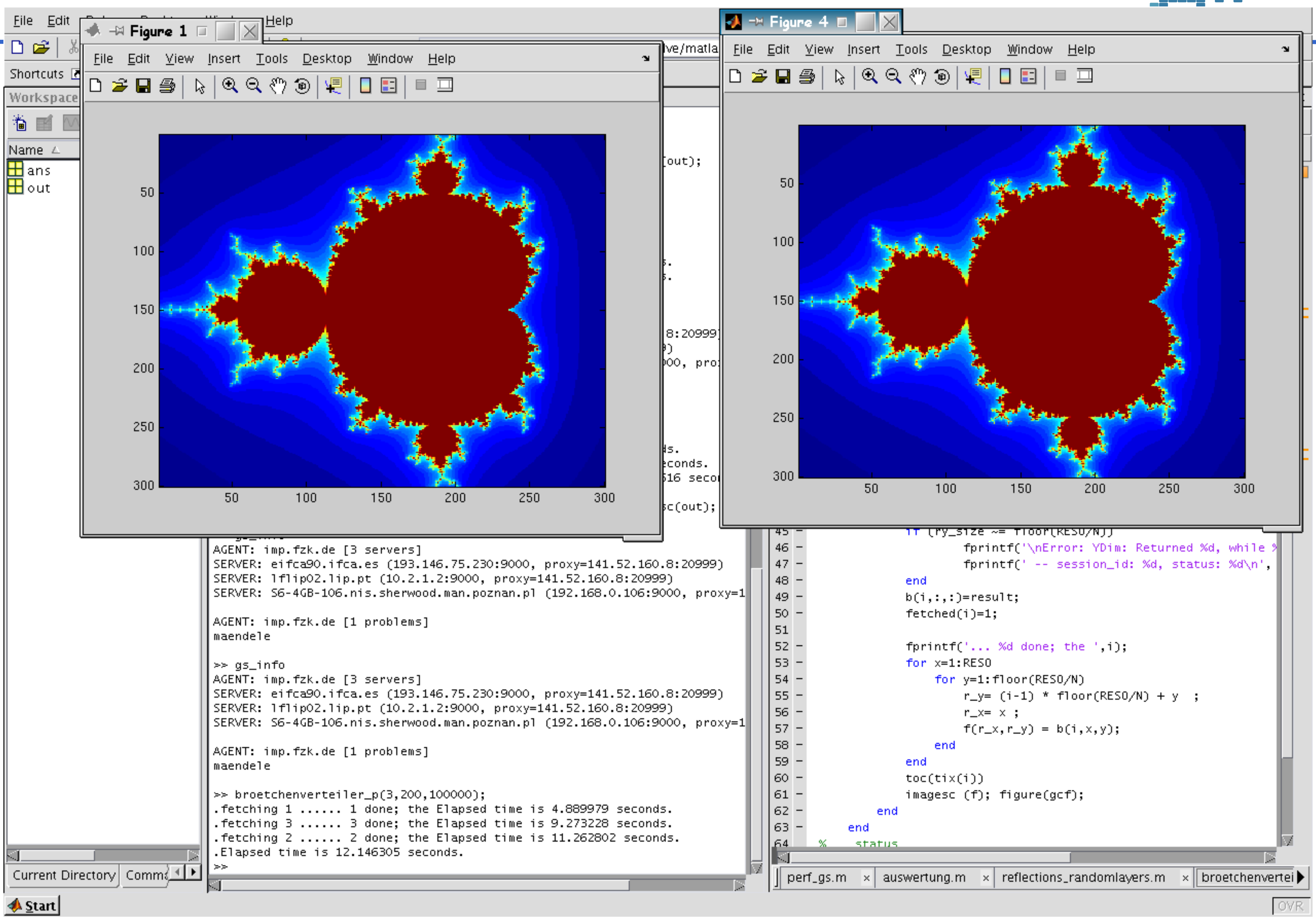


Next Steps

- Deployment of user functions
  - Deployment requires re-linking agains GS sources
  - Deployment requires resubmission of all jobs
  - Java is popular but unsupported
- Implementation of a useful algorithm
  - Current Demonstration not sexy enough for scientists
  - MPI might be beneficial (depending on algorithm)
- Data Handling
  - Get access to data at the servers
  - Currently considering GFAL + GridSolve
- Security
  - Considering use of EGEE's security enhanced DICOM

    (Digital Imaging and Communications in Medicine)

Demonstration



```
AGENT: imp.fzk.de [3 servers]
SERVER: eifca90.ifca.es (193.146.75.230:9000, proxy=141.52.160.8:20999)
SERVER: lflip02.lip.pt (10.2.1.2:9000, proxy=141.52.160.8:20999)
SERVER: S6-4GB-106.nis.sherwood.man.poznan.pl (192.168.0.106:9000, proxy=1

AGENT: imp.fzk.de [1 problems]
maendele

>> gs_info
AGENT: imp.fzk.de [3 servers]
SERVER: eifca90.ifca.es (193.146.75.230:9000, proxy=141.52.160.8:20999)
SERVER: lflip02.lip.pt (10.2.1.2:9000, proxy=141.52.160.8:20999)
SERVER: S6-4GB-106.nis.sherwood.man.poznan.pl (192.168.0.106:9000, proxy=1

AGENT: imp.fzk.de [1 problems]
maendele

>> broetchenverteiler_p(3,200,100000);
.fetching 1 ...... 1 done; the Elapsed time is 4.889979 seconds.
.fetching 3 ...... 3 done; the Elapsed time is 9.273228 seconds.
.fetching 2 ...... 2 done; the Elapsed time is 11.262802 seconds.
.Elapsed time is 12.146305 seconds.
>>
```

```
            if (ry_size ~= floor(RESO/N))
                fprintf('\nError: YDim: Returned %d, while %
                fprintf(' -- session_id: %d, status: %d\n',
            end
            b(i,:,:)=result;
            fetched(i)=1;

            fprintf('... %d done; the ',i);
            for x=1:RESO
                for y=1:floor(RESO/N)
                    r_y= (i-1) * floor(RESO/N) + y  ;
                    r_x= x ;
                    f(r_x,r_y) = b(i,x,y);
                end
            end
            toc(tix(i))
            imagesc (f); figure(gcf);
        end
    end
%    status
```

Marcus.Hardt@iwr.fzk.de

Backup Slides