Data Management Tests

David Collados

Testing Team

Contents

- gLite IO tests done.
- gLite IO pending tests.
- File Placement Service test plan.
- Catalogs test plan.

gLite IO Tests Done

- glite_creat: 4 tests (create files of different sizes in LFN and GUID format) - Bug #5079
- glite_open: 7 tests (open inexistent and existing files in LFN and GUID format using RDONLY and CREAT modes)
- glite_close: 2 tests (close file using valid and invalid handle)
- glite_write: 4 tests (write to a null handle, write file of 512B, write to RDONLY opened file, write 1GB file) – Bug #5367
- glite_read: 2 tests (read from null handle, read 512B from an existing file)
- glite_lseek: 5 tests (seek in null handle, negative offset from begin, positive offset from EOF, offset bigger than file size, offset to mid file) – Bug #5363
- glite_fstat: 2 tests (fstat for null and valid handle) - Bug #5101

gLite IO Pending Tests

- **Regression Tests (working on at present time):**

  Bugs: #4414, #4415,#4873, #5101, #5329
- **Stress Tests**
  - X concurrent clients reading the same file (X in [1, 5, 10, 30, 50]) from one or several WN or UI.
- **Check that created files are registered in the catalog.**

FPS Tests Plan 1/3

Functionality tests of the service using the Java API

- **Submit a TransferJob of X files from SE1 to SE2 (bulk file transfer files in a single job)**
- **Submit Y TransferJobs of X files each from SE1 to SE2 (job storm)**
- **Submit a TransferJob of X files from a SE to several SEs (bulk file transfer with multiple destinations)**
- **Submit Y TransferJobs of X files each from some SEs to several other SEs. (job storm with multiple sources and destinations)**
  - Being X and Y [10,100,1000]
  - Check that a valid request ID is returned and performance of the transfer: time spent to submit all jobs, how many were done or failed.
  - These tests will make use of LFNs and GUIDs (also using long file format).
  - SEs should be at different sites.

Note: All these tests test R0025 NA4 requirement (HEPCAL UC#dsreplica)

FPS Tests Plan 2/3

- **Cancel a given job**
  - Check that the status of the job is set to CANCELLING or CANCELLED.
  - Check that files with status TRANSFERDONE are kept and not deleted.
  - Check that files currently transferring will complete and will not be cancelled, while job remains in CANCELLING state.
  - Check that files pending are CANCELLED and not transferred.
  - Check that job moves to CANCELLED stated when no more files being transferred.
- **List the status of jobs and files (included above)**

FPS Tests 3/3

- **Dataset upload and download: HEPCAL UC#dsupload, UC#dsdownload**
- **Test file upload/download using the FPS servers.**
  - FTS supports protocols: http, rfio, gsiftp
  - Can use FPS to up/download files from local storages supporting any of these protocols to SEs and viceversa.
- **Repeat TransferJob tests using those protocols for a big number of files. Test performance.**

Catalog Tests 1/3

- **Basic Tests:**
  - Transfer Job with 5 files to a SE. Check in catalog and SE that files really exist.
  - Transfer again the same Job and check that there is no replication but a fast result.
  - Register 10 entries using LFN and SURL. Check that a GUID is created.
  - Add replicas (1 or more SURLs to an existing LFN).
  - Long file name format.

Catalog Tests 2/3

- **Bulk Submission Tests:**
  - Tests <METHOD> using 100, 1000, 10000 files from one local client.
  - Tests <METHOD> 100, 1000, 10000 files from one remote site client.

  Where <METHOD> can be:
  - Create entries (based on LFN and SURL),
  - Remove (LFN) and removeReplica (SURL) ,
  - Queries: listLfn (SURLs), listReplicas (LFNs), listGuidsWithoutLfn (filter), or other methods like getPermission (LFNs), getStat (LFN creationTime), etc.

The purpose of these tests is to analyze the performance of the system when accessing from the same or different sites and using a single or a mixture of the methods.

Catalog Tests 3/3

- **Concurrent Submissions**
  - Repeat bulk tests with multiple clients, all local.
  - Repeat bulk tests with multiple clients, both from the local and from remote sites.
  - Measure effect on catalogs performance increasing the number of clients.
  - Stability of the system (hangs?, fails?).

Test with a number of clients in range 10->100 and check performance.