SRM AND GFAL TESTI NG FOR LCG2

V.Petoukhov, M.Sapunov,E.Slabospitskaya* , IFVE, Protvino, Russia

Abstract

Storage Resource Manager (SRM) and Grid File Access Library (GFAL) are GRID middleware components used for transparent access to Storage Elements. SRM provides a common interface (WEB service) to backend systems giving dynamic space allocation and file management. GFAL provides a mechanism whereby application software can access a file at a site without having to know which transport mechanism to use or at which site it is running.

Two separate Test Suites have been developed for testing of SRM interface v 1.1 and testing against the GFAL file system. Test Suites are written in C and Perl languages.

SRM test suite: a script in Perl generates files and their replicas. These files are copied to the local SE and registered (published). Replicas of files are made to the specified SRM site. All replicas are used by the C-program. The SRM functions, such as get, put, pin, unPin etc. are tested using a program written in C. As SRMs do not perform file movement operations, the C-program transfers files using "globus-url-copy". It then compares the data files before and after transfer.

GFAL test suite: as GFAL allows users to access a file in a Storage Element directly (read and write) without copying it locally, a C-program tests the implementation of POSIX I/O functions such as open/seek/read/write. A Perl script executes almost all Unix based commands: dd, cat, cp, mkdir and so on. Also the Perl script launches a stress test, creating many small files (~5000), nested directories and huge files. The investigation of interactions between the Replica Manager, the SRM and the file access mechanism will help making the Data Management software better.

For any questions regarding this poster contact the Elena.Slabospitskaya@ihep.ru (SRM) and Matvey.Sapunov@ihep.ru (GFAL).

WHAT IS SRM? WHAT IS GFAL?

Grid Storage interactions require multiple software components:

- Replica Catalog services are explored for public access of data files in Grid
- Storage Resource Manager (SRM) software is used for dynamic resource management
- File access mechanism to access files from a storage system to worker nodes (GFAL).

SRM manages a disk cache or tape archiving system. It has two main functions: dynamic space allocation and dynamic file pinning. Basically SRM is represented as .web service using gsoap/https. SRM does not perform file transfer.

GFAL provides a mechanism for operation with files from UI or WN for users and jobs without transportation this files from Storage. If file exists in a disk cache of Storage, the client can read the file directly from a cache or transfer this file into its local disk. SRM pins this file during the operation time of client. If there is no file in a disk cash, SRM will get a file from its source location (e.g. tape).

Objectives

The development of Test Suites for new LCG middleware – SRM and GFAL

SRM Test Suite

The Test Suite is developed for testing of SRM interface v.1.1 with multiple data files into one SRM request (up to 99) functions. The basic goal is to test all SRM methods defined in SRM v1.1, including multi-file requests

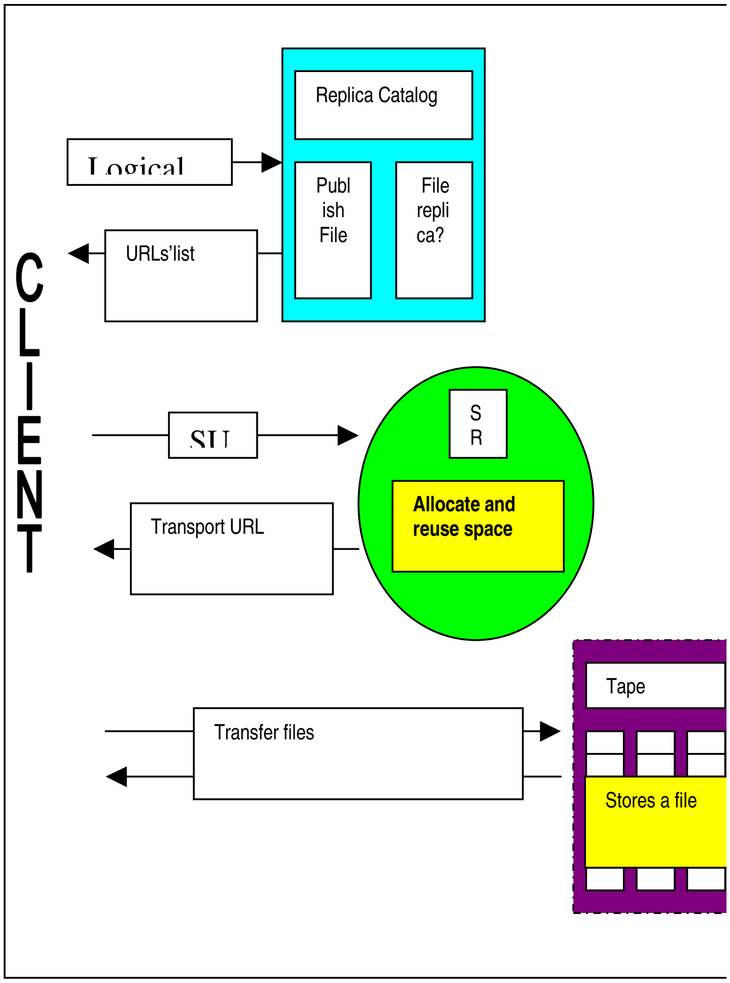


Figure 1. A logical schema of the SRM Test Suite

This suite requires a grid server's set (UI, RM, default classic SE and SRM server).

Sources of this Test Suite are in http://isscvs.cern.ch:8180/cgi-bin/cvsweb.cgi/edg-tests/tests/Stress/SRM/?cvsroot=lcgware

The SRM test suite consists of 2 programs (The **Perl** script is preparatory for the **C-program)**

1) A first program (a script in Perl) generates the data files and their replicas. The names of files and sizes (1-10 Mb) are generated by means of a random number generator. The script creates the specified number of files by "dd". These files are copied to the local SE and registered (published). Replicas of files are made to the specified SRM. All replica names are written to the file and are used by the C-program.

2) The second part of the test suite is a **program written in C-language**. First version of the test programs (srm_testPing, , srm_testPut srm_testgetProtocols, srm_testgetMetaData and srm_testGet) had been developed by **Jean-Philippe Baud .** All basic SRM operations are tested in our Test Suite - get, put, Pin, UnPin, MkPermanent, setFileStatus, getEstGetTime, getEstPutTime, getFileMetaData ,getProtocols, AdvisoryDelete. The C-program leads on couple put/get operations.

The main algorithm of put/get operations is presented below.

srm_put

- Reserve space before the file transfer (srm_put)
- mkPermanent
- set Files Status to 'Running'
- A data files put to SRM server by helps "globus-url-copy"
- set Files Status to 'Done'

srm_get

- *pin file*
- *set Files Status to 'Running'*
- A data files get from SRM server to UI
- set Files Status to 'Done'
- unpin files
- compare files before and after transfer by "diff"
- delete

The program try to delete files by helps. AdvisoryDelete SRM function and "edg-gridftp-rm"

SRM Functions are working :

- soap_call_tns__ping
- soap_call_tns__getProtocols
- soap_call_tns__put,
- soap_call_tns__get,
- soap_call_tns__getMetafileData
- soap_call_tns__setFileStatus (' Done')
- soap_call_tns__setFileStatus (' Running')

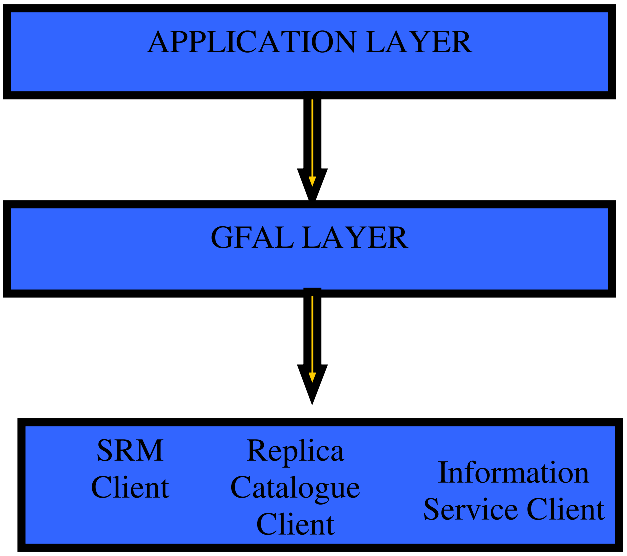
SRM Functions - Stubs:

soap_call_tns__getEstGetTime

- soap_call_tns__getEstPutTime
- soap_call_tns__pin,   Unpin
- soap_call_tns__mkPermanent

GFAL Test Suite

GFAL is additional layer that provide a POSIX I/O interface to heterogeneous Mass Storage Systems in a GRID environment. Supporting services are: Replica Catalogs, Storage Resource Managers, Mass Storage System protocols (File I/O, Root I/O, Rfio I/O, dCAP I/O), MDS



The test suite has been created to check in access to Mass Storage System with standard POSIX I/O functions over mechanism realized in GFAL. We are going to verify a functionality of existing applications and utilities, and tp test standard C functions. Since we are interested in *NIX working environment, the Perl language is used as basic tool for writing the test for external utilities. The important Perl feature is a possibility to emulate *NIX environment during an execution of external applications. Another feature is the ability to realize a complex algorithm without special experience.

The main algorithm assumes the following steps:

- Mounting of the file system over GFAL.
- Execution of most usable *nix commands.
- Checking of transferred data' integrity.
- Creation of much (~300) small files.
- Creation of huge (5*512Mb) files.
- Checking of integrity
- Unmounting file system

In the test suite we check the functionality of widely used system utilities like "cat", "cp", "mv", "diff", "dd", etc. Also, we try to develop stress test for GFAL in our test suite. The stress test flow consists from following steps: creation of many nested directories, 300 directories nested in each other; making of five big files 512 Mb each; a construction of 5000 small files, size of each file may be in range between 512 bytes and 512 Kb; creation of 5000 directories nested in parent directory. All files are created with dd utility.

Unfortunately, only the ability of creation many files was tested. No time or access speed measurements were made. We will include such time or access speed measurements in the new, improved version of test suite.

A small and simple program was written for test of basic C functions, such as open()/seek()/read()/write(). We don't describe this program because it wasn't properly tested. GFAL was checked in April 2004 with our test suite, all I/O operations worked smoothly.

ACKNOWLEDGEMENTS

We wish to thank to Jean-Philippe Baud and Zdenek Sekera for useful discussion and collaborative work.

REFERENCES

[1] I. Bird, B. Hess, A. Kowalski at al. Common Storage Resource Manager Operations
http://sdm.lbl.gov/srm-wg/doc/srm.v1.0.pdf

[2] Arie Shoshani, Alex Sim, Junmin Gu
Storage Resource Managers: **Middleware Components for Grid Storage**.Lawrence Berkeley National Laboratory Berkeley, California
http://www.lbl.gov/~arie/papers/srm.mss02.pdf

[3] Arie Shoshani, Alexander Sim, and Junmin. STORAGE RESOURCE MANAGERS. **Essential Components for the Grid.** Chapter 20. Lawrence Berkeley National Laboratory
sdm.lbl.gov/~arie/papers/SRM.book.chapter.pdf

[4]
Jean-Philippe Baud. GFAL and LCG data management
http://hepwww.rl.ac.uk/hepix/nesc/baud.ppt

[5] The Open Science Grid Consortium, Glossary
http://www.opensciencegrid.org/home/terminology.html

[6] David Foster, Don Petravick, Michael Ernst Grid File Access Proposal
http://www.uscms.org/sandc/reviews/doe-nsf/2003-07/docs/GFA-Proposal-Short-v1.0.pdf

[7]Data Management Expert Panel
http://www.gridpp.ac.uk/gridpp7/gridpp7_casey.ppt