

Storage Resource Management for Data Grid Applications

PI: Arie Shoshani (LBNL)

CoPIs: Don Petravick (Fermilab), Alex Sim (LBNL)

Other staff: Junmin Gu , Vijaya Natarajan, Alex Romosan (LBNL),

Timur Perlmotov (Fermilab)

(<http://sdm.lbl.gov/srm>)

Summary

Storage Resource Managers (SRMs) are middleware components whose function is to provide dynamic space allocation and file management of shared storage components on the data grid. They are essential grid components that complement Compute Resource Managers and Network Resource Managers by providing storage reservation and dynamic storage availability information for planning and execution of a grid job. In this project, we have developed the concepts and functionality specification of SRMs, and implemented SRMs at LBNL and Fermilab. This technology is already being used by SciDAC supported Collaboratories. This includes the STAR, CLAS, and CDF experiments as part of the Particle Physics Data Grid (PPDG) project, and the Earth Science Grid (ESG) project. Furthermore, the SRM interface development has been coordinated with and adopted by collaborators in the European Data Grid (EDG) project.

Terascale computing often generate petascale data. Such data intensive applications already overwhelm scientists who spend much of their time managing data, rather than concentrate on scientific investigations. Because such data are now vital to large scientific collaborations dispersed over wide-area networks, there is a growing activity of developing a grid infrastructure to support such applications. Storage Resource Managers (SRMs) are middleware components designed to complement Compute Resource Managers and Network Resource Managers for planning and execution of a grid jobs.

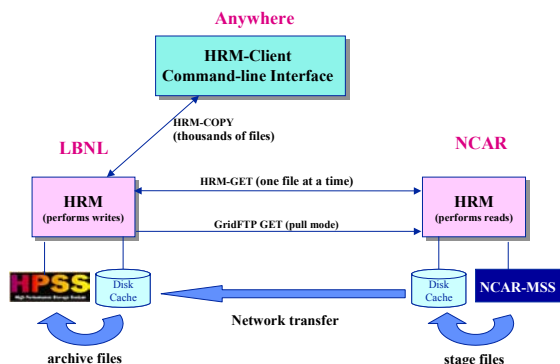
SRMs can be thought of as managing two types of resources: space and files. When managing space, SRMs negotiate space allocation with the requesting client, and/or assign default space quotas. When managing files, SRMs allocate space for files, invoke file transfer services to move files into the space, pin files for a certain lifetime, release files upon the client's request, and use file replacement policies to

optimize the use of the shared space. In addition, SRMs perform automatic garbage collection on unused files by removing files whose lifetime has expired. We highlight below some of the main achievement of this project so far.

1) Robust File Replication using SRMs

File replication of thousands of files is extremely important task in data intensive scientific applications. For example, large Climate Modeling simulations may be computed in one facility, but the results need to be stored in an archive in another facility. Moving the files by writing scripts is too tedious since the scientist needs to monitor for failure, recover from failures. This mundane seemingly simple task is extremely time consuming and prone to mistakes. We realized that SRMs are perfectly suited to perform such tasks automatically. The SRMs monitor the staging, transfer, and archiving of files, and recover from transient failures. Only a single command is necessary to request the multi-file transfer. The figure below shows the setup of having

SRMs at NCAR and LBNL to achieve continuous file replication of hundreds of files in a single request.



2) Implementation of compatible SRMs at three sites

Perhaps the greatest achievements of SRM technology can be obtained by agreeing on a standard interface. If this standard is used as a blueprint of all SRM implementations, then SRM can be accessed uniformly. To prove this point and also to proceed with SRM implementations, Jlab, Fermilab, and LBNL agreed on a uniform interface to be developed to access three different mass storage systems: JasmINE, Enstore, and HPSS. This was successfully achieved, and file transfers between a single client interface and all these systems was demonstrated in SC '02.

3) Development of an SRM for NCAR's mass storage system

NCAR's mass storage system is a unique home-grown system. It could not be accessed from the grid without a special GridFTP software written for it, which was not very likely. The LBNL team decided to adapt the SRM that was developed for HPSS to work with NCAR's MSS. This was done in a very short time, and NCAR's files residing on its MSS can now be stored and retrieved through the grid.

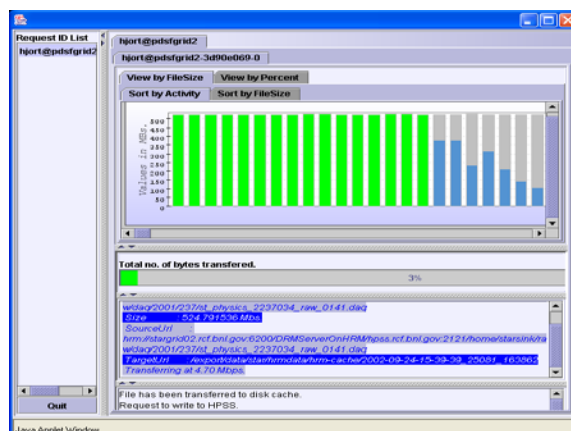
4) Development of a gridFTP-HPSS access through SRM

Some applications do not need SRM services, since they wish to get files directly

from a mass storage system to their application. Normally, GridFTP will be used for that. However, since GridFTP is not available to HPSS, we modified the GridFTP server to communicate with an SRM, which in turns communicated with HPSS. This capability was successfully developed and also demonstrated in SC '02.

5) Development of a web-based File Monitoring Tool

A web-based tool was developed to dynamically monitor the progress of the multi-file request to their destination. The figure below shows the graphical interface that indicates files already transferred, and files in the process of being transferred. This tool is used routinely to monitor progress of massive file replications.



6) Lead a coordinated effort with the EDG on standardizing SRM functionality

Our team approached the EDG community with the idea of developing a joint interface specification. We succeeded in engaging them, and now have a joint team involved in the next SRM design and interface specification. We plan to standardize the SRM interface specification through GGF.

For further information contact:

Dr. Arie Shoshani
Lawrence Berkeley National Laboratory
Tel: (510) 486-5171
Email: shoshani@lbl.gov