IL NUOVO CIMENTO Vol. 32 C, N. 2 Marzo-Aprile 2009
DOI 10.1393/ncc/i2009-10408-6

Colloquia: CSFI 2008

StoRM: A Manager for Storage Resource in Grid

A. Ghiselli, L. Magnoni and R. Zappi
INFN CNAF - viale Berti-Pichat 6/2, I-40127 Bologna, Italy

(ricevuto il 4 Agosto 2009; pubblicato online il 28 Settembre 2009)

Summary. — Nowadays, data intensive applications demand high-performance and large-storage systems capable of serving up to various Petabytes of storage space. Therefore, common solutions adopted in data centres include Storage Area Networks (SAN) and cluster parallel file systems, such as GPFS from IBM and Lustre from Sun Microsystems. In order to make these storage system solutions available in modern Data Grid architectures, standard interfaces are needed. The Grid Storage Resource Manager (SRM) interface is one of these standard interfaces. Grid storage services implementing the SRM standard provide common capabilities and advanced functionality such as dynamic space allocation and file management on shared storage systems. In this paper, we describe StoRM (STOrage Resource Manager). StoRM is a flexible and high-performing implementation of the standard SRM interface version 2.2. The software architecture of StoRM allows for an easy integration to different underlying storage systems via a plug-in mechanism. In particular, StoRM takes advantage from storage systems based on cluster file systems. Currently, StoRM is installed and used in production in various data centres, including the WLCG Italian Tier-1. In addition, Economics and Financial communities, as represented by the EGRID Project, adopt StoRM in production as well.

PACS 89.20.Ff – Computer science and technology.

1. – Introduction

High-energy physics data-intensive applications demand large storage systems capable of serving hundreds of terabytes of storage space in a fast and reliable way, with high throughput and easy management. Nowadays, Storage Area Network (SAN) solutions are commonly deployed at LHC computing centres, and parallel file systems such as GPFS [1] and Lustre [2] allow for reliable, high-speed native POSIX I/O operations.

To build a data management system that can adapt to the heterogeneous data storage resources (disk, tapes, mass storage systems), the grid community has adopted standard interfaces to virtualize the underlying resources. In particular, HEP community has adopted the Storage Resource Manager (SRM) interface standardized in the context of the Open Grid Forum (OGF). By implementing this interface, grid storage services provide a consistent homogeneous interface to the Grid to manage storage resource.

© Società Italiana di Fisica

StoRM (STOrage Resource Manager) is a flexible and high-performing implementation of Storage Resource Manager (SRM) standard interface version 2.2. It is designed to be easily adapted to different underlying storage system via a plug-in mechanism, and, in particular, it leverages the advantages of high-performance storage systems based on cluster file system such as GPFS file system from IBM and Lustre from Sun Microsystems. StoRM is designed to support guaranteed space reservation and direct access (native POSIX I/O call), as well as other standard Grid access protocols.

Moreover, StoRM is able to leverage the advantages resulting from cluster approach in a Grid environment, enabling data intensive applications to directly access data into the storage resource without interact with any other transfer services.

2. – Storage Resource Manager

The Grid, by definition, is composed by many and different (and heterogeneous) resources. In particular, for storage resources, a user may want to write and read files accessing different types of storage (*e.g.*, different file systems, different storage systems like disks or tapes, etc.) in a uniform way.

Storage Resource Managers (SRM) [3] are middleware services whose function is to manage the dynamic use of space and files in a storage resource on the Grid, through a standard interface that provides uniform access to storage resource. From one side a client can request to the SRM to reserve space and to manage files and directories, from the other side a SRM can dynamically decide which files to keep in the storage space and which files to remove (*e.g.*, when free space is needed).

2˙1. *File types*. – A file can be volatile, permanent or durable. A *volatile* file is a temporary file with a lifetime associated. When the lifetime expires the file can be deleted by the garbage collector of the SRM. Therefore, a client can *pin* a file for a specific amount of time (lifetime of the pinning) and be sure to have the file available for the whole operation time. A *permanent* file can only be removed by the owner. A *durable* file has a lifetime associated with it, but has a different behaviour when the lifetime expires. It can be removed by the garbage collector only after confirmation by the owner.

2˙2. *Space types*. – Clients can negotiate and acquire space that the SRM assigns to them. Otherwise, the SRM assigns a default space size that depends on its policy. Similar to file types, the spaces assigned by the SRM have types. Thus, Volatile and Durable space types have a lifetime associated with them. Permanent space has unlimited lifetime. Files of a certain type can only be assigned to a space of the same type. Furthermore, the lifetime of a file cannot exceed the lifetime of the space it is assigned to. Spaces are also associated to storage quality characteristics. The concept of *storage classes* is used to distinguish different properties of the storage system managed by a SRM. For example a user can choose to create a file in a slow but reliable tape-based system or in a faster disk-based system.

3. – StoRM

StoRM [4] is a storage resource manager for disk-based storage systems implementing the SRM interface version 2.2. It is designed to support guaranteed space reservation and direct access (native POSIX I/O calls) to the storage, supporting as well other standard access libraries like RFIO [5]. This allows applications to perform a standard POSIX

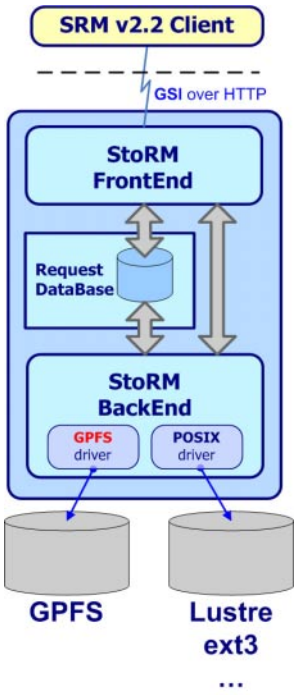

Fig. 1. – StoRM architecture.

operation without interacting with any external service that emulates data access, with the result of improving the performance when the underlying file system is efficient.

StoRM takes advantage from high-performance parallel and cluster file systems, but it works on top of any standard POSIX file system with ACL (Access Control List) support, like XFS and ext3. Indeed, StoRM uses the ACLs provided by the underlying file systems to implement the security model.

3˙1. *Architecture*. – Figure 1 shows the StoRM architecture. StoRM has a multilayer architecture composed by two main stateless components: the *Front-End* (FE) and the *Back-End* (BE). The Front-End exposes the SRM web service interface, manages user authentication and stores the requests data into the database. The Back-End is the core of StoRM service since it executes all synchronous and asynchronous SRM functionalities. It takes care of file and space metadata management, enforces authorization permissions on files and interacts with file transfer services. Moreover the Back End is able to use advanced functionalities provided by the file system (for example by GPFS and XFS) to accomplish space reservation requests. The Back End logic is decoupled from the wrapper component that provides file system support. This plug-in mechanism allows to easily add new support for different file systems.

In StoRM the database is used to store request information and metadata of files and space. One or more instances of the StoRM components can be deployed on different machines using a centralize database service. StoRM has a light and flexible namespace mechanism that leverages the underlying file system structure (*i.e.* it is the file system that knows where files are stored and this information is not replicated into a database),

hence StoRM does not need to query the database to know the physical location of a file that can be deduced from the requested SURL.

3˙2. *StoRM in cluster*. – The idea behind StoRM is to provide a SRM service capable to satisfy the high availability and scalability requirements coming both the small and growing centre and Tier1-scaled centre of HEP community. The result is a simple, configurable and highly scalable architecture. Most of the StoRM components support replication, that means the capability to deploy multiple instances in different hosts, to avoid bottleneck and enhance load balancing.

Multiple Front-End instances can be deployed on separate machines and configured to work on the same database and the same Back-End service. For the load balancing a simple dynamic DNS configuration can be used. A CNAME must be created into DNS to aggregate FE services in a single SRM endpoint, in this way additional new front-end services can be easily added on-the-fly to improve the rate of requests managed by the system. The communication between the different components take place in two ways. The database is used to store and dispatch asynchronous requests, whilst a direct RPC connection is used as channel for the synchronous calls.

In StoRM each Storage Area can be configured to use to a dedicate GridFTP server, and, with the same CNAME techniques, a pool of GridFTP servers can be used to satisfy high throughput requirements. In addition to this approach, StoRM v1.4 introduce a smart GridFTP load balancer to use GridFTP pool without act at DNS level. The load balancer takes care of distributing FTP requests among the specified set of GrifFTP server, following the proper balancing policy. These balancing policies can be simple ones, as the case of RoundRobin or Weight-based distribution, or advanced policies, that rely on a GridFTP sensor installed on the GridFTP machines to know at real time the load on server hosts, in order to proper distribute requests among them.

3˙3. *Security*. – Grid storage management raises several storage security issues: data integrity, confidentiality, authentication, non-repudiation and authorization are only some examples. Every aspect must be taken into account at different levels (site and virtual organization). StoRM manages some important security aspects at site level, in particular regarding authorization to a specific resource (StoRM service, file and space). Authorization is based on subject authentication.

- Grid user authentication: StoRM uses the Grid Security Infrastructure (GSI) [6] for authentications. Some FQANs (short form for Fully Qualified Attribute Name) could be present within X.509 proxy certificates, as defined by the Virtual Organization Membership Service (VOMS) [7]. Upon a request from a grid user belonging to a managed Virtual Organization (VO), the VOMS service produces a VOMS proxy certificate, that is a X.509 Attribute Certificate Extension. The user must have a valid proxy certificate to submit the SRM request to the StoRM server.

- Mapping grid user to local user: The access restriction to storage resources is based on the mapping from a grid identity to a local user identifier (ID). The grid identities mapping is accomplished by a call to the LCMAPS service, so at runtime every Grid user is associated to a local user ID.

- Authorization Sources: StoRM interacts with different external authorization services (authorization sources) to verify if the user can perform the specified operation

on the requested resources. Typically, authorization policies within the authorization sources are expressed in terms of Grid identities and SURLs. Each authorization source is queried with a specific order and the results are evaluated with a deny-override algorithm.

– AuthZ Enforcement: Once the user request has been authorized, StoRM enforces the permission on the data by setting a file system ACL for the corresponding user on the specified resource. StoRM sets up ACLs on the physical file to allow direct access from a worker node. A new access control entry (ACE) of the ACL is built from the local user to whom Grid credentials are mapped and with the permission returned by the authorization source. The ACE added will be removed once the data access completes or when the access operation lifetime expires.

4. – Conclusion

In this paper we presented StoRM as a flexible and high-performance solution for Storage Resource Management in Grid environment for parallel and cluster file systems and more advanced storage solutions. The StoRM layered architecture gives scalability and reliability, the driver approach promotes easy integration with different storage resources and the direct access support provides a seamless way to integrate non–Grid-aware applications. With this approach data centres are able to choose the preferred underlying storage system maintaining the same SRM service, leveraging on high-performance parallel file systems like GPFS and Lustre in Grid environment.

REFERENCES

[1] Schmuck F. and Haskin R., *GPFS: A shared-disk file system for large computing clusters*, in *Proceedings of the 1st USENIX Conference on File and Storage Technologies, FAST'02* (SENIX Association, Berkeley, CA, USA) 2002.
[2] Sun-Microsystem, *Lustre File System - White Paper*, `http://www.sun.com/software/products/lustre/docs/lustrefilesystem_wp.pdf` (Oct 2008).
[3] Shoshani A., Sim A. and Gu J., *Storage resource managers: essential components for the grid*, in *Grid Resource Management: state of the art and future trends* (Kluwer Academic Publishers, Norwell, MA, USA) 2004, pp. 321-340.
[4] Ghiselli A., Magnoni L. and Zappi R., *StoRM, a flexible solution for storage resource manager in Grid*, in *Nuclear Science Symposium Conference Record 2008, NSS'08, Proc. IEEE.*
[5] RFIO/IB Project, `http://hikwww2.fzk.de/hik/orga/ges/infiniband/rfioib.html`.
[6] The Globus security team, *Globus toolkit 4 grid security infrastructure: A standard perspective* (2005).
[7] Alfieri R., Cecchini R., Ciaschini V., dell'Agnello L., Frohner A., Gianoli A., Lorentey K. and Spataro F., *Voms: An authorization system for virtual organizations*, in *Proceedings of the 1st European Across Grids Conference, Santiago de Compostela* (Springer-Verlag) 2003.