ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Pisa

INFN/TC-05/16
December 22, 2005

STORAGE MANAGEMENT IN THE GRID

Flavia Donno[1], Gigliola Vaglini[2]

[1] INFN Pisa, Italy
[2] Dipartimento di Ingegneria dell'Informazione, Universitá di Pisa, Italy

Abstract

An application running on a Grid infrastructure needs to be able to transparently access distributed data available at some storage centers. The Grid middleware and infrastructure are responsible to reserve the necessary space, enforce local authorization policies, privacy and security, allow for transparent access to the underlying local storage system, etc. The goal and main contribution of this paper is to give an overview of the state of the art of storage management technologies in the Grid. Clear overviews are important in order to better drive the research evolution and for general guidance. We present a summary of hardware and software storage solutions adopted in various research centers part of the LHC Computing Grid infrastructure and the open problems to establish standards for “file sharing” and a “master namespace server”. Then we analyze the requirements for a Grid application, we present the Grid solutions proposed and the prototypes currently under test.

Published by ***SIS–Pubblicazioni***
Laboratori Nazionali di Frascati

1 Introduction

Within a single computer, standard elements including the processor, storage, operating system, and I/O exist. The concept of Grid computing is to create a similar environment, over a distributed area, made up of heterogeneous elements including servers, storage devices, and networks – a scaleable, wide-area computing platform. The software that handles the coordination of the participating elements is analogous to the operating system of a computer or server. As it happens on a local computer, application persistent data are stored on disk or tape systems. Whether running on a LAN cluster or over a Grid infrastructure, a computational user task often needs to access or produce data. A certain number of functionalities need to be guaranteed to an application: authorization and security policies (local or global) need to be enforced during data access; mechanisms of file pinning guarantee that a storage garbage collector does not remove files that are used by some application; space reservation is another important feature that allows a Grid scheduler to check for availability of the space and then allocate it. Therefore, the possibility to manage disk-space via the Grid becomes essential for running data-intensive applications on the Grid. Data movement and replication are also important functions to guarantee optimized access to files. In what follows we present the hardware and software solutions adopted in order to realize a storage Grid service, known as the ***Grid Storage Element***.

In *Section 2* we present an overview of the storage solutions currently in use in Grid research computing facilities from the hardware point of view. The big variety of hardware solutions and products imposes the deployment of several management and data access software packages. In *Section 3* we present an overview of the products offered by IT vendors in the area of storage management. The use of *disk pool managers* becomes a popular solution especially among the Grid research communities. However, such systems present often the problem of offering proprietary protocols for data access, forcing the applications to adapt the I/O interface to these systems. Disk pool managers can as well be interfaced to legacy hierarchical storage solutions, such as *Hierarchical Storage Management* (HSM) software or special Mass Storage Management systems. In *Section 4* we illustrate existing solutions and prototypes and we highlight advantages and disadvantages.

In *Section 5* we present the *Storage Resource Manager* (SRM) interface. It has been defined in a working group of the Global Grid Forum. This interface is an attempt to standardize the interfaces to the several storage solutions adopted, and to identify as well a set of functionality needed by Grid applications from Grid Storage Services. As of today several implementations of the SRM interface are available. We give as well an overview of the current SRM implementations and outline what is still missing. Finally a related work report follows. In the conclusions we summarize the next steps in the area of Storage Management for Grid and point the reader to some interesting work in this area.

2 Hardware Storage Solutions

Today’s data intensive applications demand for large storage systems capable of serving hundreds of terabytes or even petabytes of storage space. The access to such a space needs to be fast and is achieved with high performance I/O solutions.

2.1 Disk based solutions

In the LHC Computing Grid (LCG) [1] infrastructure, computing centers are organized in a hierarchical manner. CERN [14] hosts the accelerator and the detectors used by the four High Energy Physics experiments to study the fundamental particles of the matter. CERN is the Tier 0 LCG Computing Center. Each year 10 to 15 Petabytes of data will be acquired from the detectors through complex chains of sophisticated electronic apparatus and stored in robotic tape libraries and disk-based storage systems. First data processing happens at CERN. Important samples of data of the order of a few Petabytes are then replicated to Tier 1 centers. These centers dispose of conspicuous computing and storage resources together with a good local IT support. From Tier 1 sites, physicists located at smaller Tier 2 and Tier 3 centers, disposing of fewer hardware resources and local specialized support, then analyze data.

The most popular storage systems today used in small or medium size computing centers are small storage products that use low-cost parallel or serial ATA disks [3] and can operate at the block or file level and aggregate RAID controllers and capacity. The arrays perform load balancing among self-contained storage modules. Such systems have been adopted to replace the old direct-attached storage (DAS) [3]: normally SCSI disks or RAID systems. Such old systems in fact presented many problems. DAS uses a bus topology in which systems and storage are connected by a bus in a daisy chain. Data availability in such systems is very critical since if any of the components fails, the entire system becomes unavailable. Other issues are limited scalability (only up to 15 devices in a chain), static configuration, captive resources and storage utilization, performance (only 40 MB/s), system administration, etc.

A SAN [3] is a high-speed special-purpose network (or sub-network) that interconnects different kinds of data storage devices with associated data servers. A storage area network is usually clustered in close proximity to other computing resources but may also extend to remote locations for backup and archival storage. A storage area network can use existing communication technology such as IBM's optical fiber ESCON [15] or it may use the newer Fiber Channel technology. SANs support disk mirroring; backup and restore; archival and retrieval of archived data; data migration from one storage device to another; and the sharing of data among different servers in a network. SAN solutions operate at the block level.

While SANs take care of the “connectivity” and “storing” layers of a storage system, NAS systems take care of “filing” operations. NAS [3] is a product concept that packages file system hardware and software with a complete storage I/O subsystem as an integrated file server solution. The network-attached storage device is attached to a local area network (typically, a Gigabit Ethernet) and assigned an IP address. File requests are mapped by the main server to the NAS file server. NAS servers are sold with a few hundred gigabytes and extend up to tens of terabytes of usable storage. They are normally specialized servers that can handle a number of network protocols. Some NAS systems provide for dynamic load balancing capabilities, dynamic volume and file system expansion and offer a single, global namespace. NAS systems can deliver performance of tens of Gigabytes/sec in a standard sequential read/write test.

However, besides being expensive, one of the problems with NAS systems is the incompatibility of proprietary solutions and the inexistence of interoperable NAS heads defining a global name space. A system administrator needs to independently manage different storage partitions defined by the different vendor products.

2.2 Tape based solutions

As of today disk-based solutions can provide storage capacity to up to hundreds of Terabytes. Tape servers can provide Petabytes storage capacity; however often they do not satisfy the performance requirements. Therefore, they are used essentially as tertiary data stores accessible through a transparent user interface. Big robotic libraries are normally deployed in well established computing facilities.

Mass Storage Management Systems: Among the most commonly used Mass Storage System (MSS) software products used to manage tape based solutions are: CASTOR [21] developed at CERN, ENSTORE [8] developed jointly by Fermilab, near Chicago, and DESY in Germany, High Performance Storage System (HPSS) [17] started as a joint effort between industries and research. Such mass storage systems have been developed to provide an answer to the need coming from the research field of providing automatic and transparent access to tape storage. The commercial market normally provides only backup solutions and Hierarchical Storage Manager (HSM) software which often do not satisfy the requirements of fast access to data mainly in read-only mode. One of the most successful commercial products used in the High Energy Physics (HEP) environment is Unitree (DiskXtender) by EMC^2.

3 Grid Storage, Parallel and Distributed Filesystems

3.1 Grid Storage

Lately the term "Grid storage" has crept into the product literature of vendors and refers to two items: a topology for scaling the capacity of NAS in response to application requirements, and a technology for enabling and managing a single file system so that it can span an increasing volume of storage. Scaling horizontally means adding more NAS arrays to a LAN. This works until the number of NAS machines becomes unmanageable. In a "Grid" topology, NAS heads are joined together using clustering technology to create one virtual head. NAS heads are the components containing a thin operating system optimized for NFS (or proprietary) protocol support and storage device attachment. Conversely, the vertical scaling of NAS is accomplished by adding more disk drives to an array. Scalability is affected by NAS file system addressing limits (how many file names you can read and write) and by physical features such as the bandwidth of the interconnect between the NAS head and the back-end disk. In general, the more disks placed behind a NAS head, the greater the likelihood the system will become inefficient because of concentrated load or interconnect saturation. Grid storage, in theory, attacks these limits by joining NAS heads into highly scalable clusters and by alleviating the constraints of file system address space through the use of an extensible file system. The development of storage Grids clearly is geared toward NAS users today, but others might one day benefit from the Grid storage concept. For instance, making disparate SANs communicate and share data with each other in the face of non-interoperable switching equipment is today a complicated problem to solve. By using clustered NAS devices serving as gateways and managers of the back-end SANs, one would gain improved capacity, file sharing and management generally.

At IBM's Almaden Research Center, work is proceeding on a self-described Grid storage project aimed at creating a "wide-area files sharing" approach. In the Distributed Storage Tank (DST) project, the objective is to extend the capabilities in a "Storage Tank" - a set of storage technologies IBM offers that includes virtualization services, file services and centralized management - to meet the needs of large, geographically distributed corporations. IBM is looking at not yet used capabilities in the NFS Version 4 standard to help meet the need. DST extends to NFS clusters that can be used to build a much larger Grid with a single global file namespace across a geographically distributed and heterogeneous environment. Making the approach open and standards-based requires a schema for file sharing that is independent of a server's file and operating systems, and that does not require the deployment of a proprietary client on all machines. IBM is working with the Global Grid Forum's File System Working Group [7] because its intent is to produce a standards-based Lightweight Directory Access Protocol (LDAP) server to act as the master namespace server.

3.2 Distributed Filesystems

Cluster and distributed file systems are an alternative form of shared file system technology. Such file systems do not use a separate meta-data server. They are designed to work only in homogenous server environments and improving storage manageability is not a goal. However they are used at many centers as a solution to share storage among a farm of computing nodes. Using very high-speed connections (Switched Gigabit Ethernet, Infiniband, etc.) such solutions provide for POSIX I/O, centralized management, load balancing, monitoring, and fail-over capabilities, among others. Users do not always have full control over their applications. Adopted proprietary solutions, legacy software, performance factors, etc. often do not allow for changing (re-writing) an application in order to make it Grid-aware. The integration of existing high-performance, parallel file-systems into a Grid infrastructure allow users to take advantage of such technologies. Widely used, high-performance distributed file-systems are IBM/GPFS, LUSTRE and PVFS-2 [2]. Filesystems such as NFS and AFS are widely used, but they present quite a few performance and scalability problems [19,20].

3.3 Parallel Filesystems

The IBM General Parallel File System (GPFS) for Linux is a high-performance shared-disk file-system that can provide data access from all nodes in a Linux cluster environment. Parallel and serial applications can access shared files using standard UNIX file-system interfaces, and the same file can be accessed concurrently from multiple nodes. GPFS provides high availability through logging and replication, and can be configured for fail-over from both disk and server malfunctions. To support its performance objectives, GPFS is implemented using data striping across multiple disks and multiple nodes, and it employs client-side data caching. GPFS provides large block size options for highly efficient I/O and has the ability to perform read-ahead and write-behind file functions. GPFS uses block level locking based on a sophisticated token management system designed to provide data consistency while allowing multiple application nodes concurrent access to a file. When hardware resource demands are high, GPFS can find an available path to the data by using multiple, independent paths to the same file data from anywhere in the cluster, when the underlying storage infrastructure allows it.

Other systems with similar characteristics are PVFSv2 and LUSTRE. LUSTRE is a commercial product by Cluster File System, Inc. initially distributed free of charge. It is advertised as scalable to more than 10,000 clients. It is stable and reliable but quite invasive in terms of changes to the system kernel. It offers metadata redundancy and multiple metadata servers for fault tolerance. Only plain striping of data across the servers is allowed at the moment. Data recovery features and a POSIX interface are also offered. The Parallel Virtual File System (PVFS) project is conducted jointly between The Parallel Architecture Research Laboratory at Clemson University and The Mathematics and Computer Science Division at Argonne National Laboratory. PVFS is “easy” to install and “very light" (it can use the underlying native file system to store data), and provides user-controlled striping of files across nodes. Beside standard UNIX I/O, PVFS provides multiple I/O interfaces, such as MPI-IO via ROMIO. We have tested PVFS-2 version 1.0.1 and shown that it is still not ready for a production environment. We have performed as well some performance and scalability tests to see which file system can satisfy the requirements of a Tier 1 center for LCG, serving between 20 and 50 Terabytes of data to computing farms with about 1000 client nodes. The result seem to go in favor of LUSTRE.

4 Disk Pool Managers (DPM)

In the Grid community, there is a tendency of providing disk pool managers capable of serving large amounts of disk space distributed over several servers. CERN, for instance, adopts this technique to serve several hundreds of Terabytes of disk storage organized in JOBDs (Just a Bunch of Disks) or RAID cabinets as secondary storage for their robotic tape libraries. However, most of the time, such systems do not allow for POSIX I/O, but file access is guaranteed via Grid or specific protocols. A DPM normally presents the catalogue of available files to the users as a browseable file system-like tree; but no real file system is available. This storage management software provides second level storage for data stored in big archiving robotic libraries. Through application software, a user asks for access to a file stored on tape. If the file is available on one of the second level storage disk server, it is served either via custom remote I/O calls or copied on a disk directly accessible by the user. Otherwise, the robotic controller in the tape library is instructed to mount the tape containing a copy of the requested file on one of the available drives and spool the file on the file system of one of the servers managed by the DPM. The DPM is generally in charge of managing the disk space served by the storage servers. It is responsible for deleting unused files saved on tape in order to make space for further requests, pinning a file on disk while it is used, reserving storage for new files, etc.

In what follows we give an overview of disk pool management systems deployed and used in production environment in the major HEP research labs.

d-Cache: dCache [9] is a software-only Grid storage appliance jointly developed by DESY and Fermilab. It is the DPM of the Enstore MSS. The name space is uniquely represented in a single file system tree. dCache optimizes the throughput to and from data clients and smoothes the load of the connected disk storage nodes by dynamically replicating files. The system is tolerant against failures of its data servers. Access to data is provided by various FTP dialects, including GridFTP [11] discussed later in this paper, as well as a proprietary protocol (gsi-dCap), offering POSIX-like file system operations like open/read, write, seek, stat, close. Some of the limitations of the dCache system are the complex configurability, some instability (shown during the CMS Data Challenge [22]) and the authorization control mechanisms. The Grid Security Infrastructure (GSI) [11] protocol implemented for file transfer allows for the check of user credentials in order to allow access to files. However, ACLs and mechanisms to enforce local access or priority policies are lacking.

LDPM: The LCG Lightweight Disk Pool Manager (LDPM) [5] is a complementary solution to the dCache system. It focuses on manageability and therefore it is easy to install and configure. It requires low effort for ongoing maintenance while allowing for easy addition and removal of resources. As for dCache, LDPM supports multiple physical partitions on several disk severs and allows for different types of disk space: volatile and permanent. In order to protect against disk failure LDPM provides support for multiple replicas of a file within the disk pools. As far as data access is concerned, also LDPM supports several proprietary POSIX-like protocols, such as rfio and ROOT I/O. GridFTP is used for file transfers outside the IP domain. The namespace is organized in a hierarchy. Through plug-ins, several types of security mechanisms can be enforced: GSI, Kerberos, etc. The role of a Grid user is then mapped to LDPM Group IDs and enforced. The ownership of files is stored in the LDPM internal catalogues. Unix and POSIX ACLs permissions are implemented. The LCG DPM is a “quite attractive” solution for Tier 1 and Tier 2 sites. However, it has yet not reached a level of stability to be safely deployed in a production environment.

SRB: While dCache and LDPM are LAN solutions, the Storage Resource Broker (SRB) [16] developed at San Diego Super Computing Center is client-server middleware that provides uniform access for connecting to heterogeneous data resources over a wide-area network and accessing replicated data sets. It uses a centralized Meta Data Catalog (MCat) and supports archiving, caching, synchs and backups, third-party Copy and Move, Version Control, locking, pinning, aggregated data movement and a Global Name space (filesystem like browsing). SRB provides as well for collection and data abstraction presenting a Web Service interface. The SRB has been integrated in the LCG Grid infrastructure; however, the centralization of the SRB catalogue has shown its limitations.

SAM: The Storage Access Manager (SAM) [18] developed at Fermilab tries to tackle the storage problem in a slightly different way. The storage resource is considered one of the possible Grid resources and as such is one of the variables in the scheduling decisions for a job. SAM is therefore a job management system with data management integrated patterns. It is interfaced with Enstore [8] and dCache. Before scheduling a job, SAM makes sure that the data sets needed by the application are available at the site where the computation happens. In case they are not, SAM schedules data retrieval and data transfer to the computing site where the job is scheduled to run. SAM is a completely integrated solution and it is “hard” to interface to other solutions or Grid infrastructures, such as LCG. Therefore, at the moment, it has been adopted only at FNAL and at collaborating sites.

5 The Storage Resource Manager

In the Grid environment the need for a homogeneous, transparent interface to storage devices has brought Grid scientists to the definition of the Storage Resource Management (SRM) interface [6]. SRM is a middleware component whose function is to provide dynamic space allocation and file management on shared storage components on the Grid. The SRM effort has resulted in the adaptation of the standard specification, and the development of multiple SRM middleware components that inter-operate. This approach is particularly essential for providing Grid access to complex MSSs, and is now deployed in multiple institutions around the world. The SRM specification standardizes the interface, thus allowing for a uniform access to heterogeneous storage elements. SRMs leave the policy decision to be made independently by each implementation at each site. Resource reservations made through SRMs have limited lifetimes and allow for automatic collection of unused resources thus preventing clogging of storage systems with "forgotten" files. The storage systems can be classified on basis of their longevity and persistence of the data they store. Data can be considered to be temporary and permanent. For example disk caches might allow for spontaneous deletion of the files, while deletion of the file stored in robotic tape storage can be very problematic.

The SRM interface consists of the five categories of functions: Space Management, Data Transfer, Request Status, Directory and Permission Functions. The SRM protocol foresees support for different kinds of communication protocols. One of these protocols is that used for data transfer operations between two SRM servers, when an application requests a third party copy operation (e.g. GridFTP [11]). Another negotiable protocol is the file access protocol used by the application, once the SRM request has been successfully completed. Most of the DPMs in use today implement the SRM protocol. However, the SRM interface proposed is still lacking some needed functionalities such as file locking and quota management.

For what concerns filesystems, as of today there are no SRM interfaces to distributed filesystem. At LBL, a group of researchers have developed the so called Disk Resource Manager (DRM) [12], an SRM interface for local file systems.

The StoRM project [4] promoted by the Italian National Institute of Nuclear Physics (INFN) aims to provide a full SRM v2 interface to a generic parallel distributed filesystem such as GPFS that offers native support for space and ACL management. StoRM allows for transparent and coherent access to storage systems from both a Grid and a local environment while providing access to high-performance, modern parallel file systems.

6 Related work

With our paper we provide an overview of the state of the art of storage management and technologies in the Grid. Many works tackle the problem of storage management. In [3] a good overview of hardware Storage technologies and file systems is given. In the attempt of promoting extensions to the NFSv4 protocol for a file access scalability plus operating system and storage system independence, Dean Hildebrand and Peter Honeyman of the Michigan University give in [2] an overview of existing parallel filesystems and report results on their performance and limitations. DPMs are discussed in

[5,9,16]. The SRM protocol description and the work of the GGF working groups can be found in [6].

7 Conclusions

Although there are many activities on going in the field of storage management in the Grid, still a definitive solution for a Storage Element is missing. While at the hardware level many solutions exist, fundamental features to make them part of a Grid infrastructure are missing. Storage Elements need to be monitored and managed as a whole, need to interoperate in order to ensure that whenever resources availability and quality of service go below a certain threshold at a site, another site can act as a backup taking over the load. Data transfer mechanisms need to be reliable and based on common protocols. Checksums and retries should be included in a data transfer service part of the Storage Element. Furthermore it is very important in a Grid infrastructure to be able to globally manage space, assign quota, and ensure that resources are not wasted by malicious hackers. The efforts started in the Global Grid Forum Grid Storage Management and Grid File System Working Groups are definitively good steps forward in this direction.

The Grid is supposed to be capable of including various kinds of devices or resources such as computers, networks, data, software, storage, etc. A Grid scheduler needs to interact with lower level scheduling systems. Today, most of these local resource management systems just react to requests in a best effort fashion according to some priority list scheduling without additional information or guarantees when the requested resource will actually be available. This needs to be augmented by features like reservations for storage, network and CPU resources, deadlines or estimates about availability of data or the execution schedule. Grid scheduling should include capabilities that allow planning of the job execution. Grid scheduling should be capable of coordinating the allocation of these resources for Grid jobs. A working group under the umbrella of the GGF, called Grid Scheduling Architecture Research Group (GSA-RG) [10], is actively pursuing the goals mentioned above. Finally production projects such as EGEE and LCG [1,13] aim to provide an international multi-purposes Grid infrastructure to serve e-science and industry consortia world-wide. Such projects are a good test infrastructure to consolidate proposed ideas and standards in this field.

8 References

1. LHC Computing Grid: http://www.cern.ch/lcg/, June 2005.
2. D. Hildebrand, P. Honeyman. Exporting Storage Systems in a Scalable Manner with pNFS, 22nd IEEE and 13th NASA Goddard Conference on Mass Storage Systems and Technologies, Monterey, CA, April 2005.
3. M. Farley. Storage Networking Fundamentals, Cisco Press; ISBN: 1587051621, Dec. 2004.
4. F. Donno, A. Ghiselli, L. Magnoni, R. Zappi. StoRM: Grid Middleware for Disk Resource Management, Comp. in High Energy Physics, CHEP, La Jolla, California, March 2004.
5. J.-P. Baud, J. Casey. Evolution of LCG-2 Data Management, CHEP, La Jolla, California,

March 2004.

6. GGF – SRM Working Group: https://forge.gridforum.org/projects/gsm-wg/, June 2005.
7. GGF – GFS WG: https://forge.gridforum.org/projects/gfs-wg/, June 2005.
8. F. Donno. Enstore Systen Analysis for Data Handling in CDF Run II, CDF Publications,
FERMILAB, CDF Note no. 4775 (pp 10), October 12, 1998.
9. M. Ernst, P. Fuhrmann, T. Mkrtchyan, J. Bakken, I. Fisk, T. Perelmutov, D. Petravick: Man-
aged data storage and data access services for Data Grids, CHEP, La Jolla, California, March
2004.
10. GGF – Grid Scheduling Architectures Working Group,
http://www-ds.e-technik.uni-dortmund.de, June 2005.
11. The Globus Toolkit: http://www.globus.org, June 2005.
12. Disk Resource Manager: https://plone3.opensciencegrid.org/activities/DRM/docs/drmmain
13. EGEE Project: http://www.eu-egee.org/, June 2005.
14. CERN: http://www.cern.ch, June 2005.
15. ESCON: http://www.c2p.com/ESCONTraining.pdf, June 2005.
16. SRB: http://www.sdsc.edu/srb/, June 2005.
17. HPSS: http://www.hpss-collaboration.org/hpss/index.jsp, June 2005.
18. SAM: http://d0db.fnal.gov/sam, June 2005.
19. T. Olivares, L.Orozco-BarbosaB, F.Quiles, A. Garrido, P.J.Garcia. Performance study of
NFS over Myrinet-based clusters for parallel multimedia applications, 2001 IEEE Canadian
Conference on Electrical and Computer Engineering- CCECE, Toronto (Ontario), Canada,
May 13-16, 2001.
20. S. Blumson: AFS write performance – A campaign paper, CITI Technical Report, July
1992, http://www.citi.umich.edu/techreports/reports/citi-tr-92-7.pdf
21. CASTOR: http://www.cern.ch/castor/, June 2005.
22. D. Bonacorsi, CMS at CNAF Tier1: Pre-challenge production and Data Challenge 04, III
Workshop Calcolo-Reti INFN, Castiadas (CA) - May 24-28, 2004.
http://www.ca.infn.it/ws2004/doc/III_giornata/bonacorsi.ppt