Virtual Storage System for the Grid Environment

Darin Nikolow[1], Renata Słota[1], Jacek Kitowski[1,2], and Łukasz Skitał[1]

[1] Institute of Computer Science, AGH-UST, al.Mickiewicza 30, Cracow, Poland
[2] Academic Computer Center CYFRONET AGH, ul.Nawojki 11, Cracow, Poland

Abstract. Experiments in a Grid-based virtual laboratory, as well as, simulation and visualization grid computing usually deal with large data kept in different locations often far away from each other. These data need to be archived. The goal of the Virtual Storage System (VSS) for grid-based accessing is to integrate the mass storage resources distributed geographically into a common storage service. In this paper the architecture of a virtual storage system is discussed and implementation details are presented.

1 Introduction

Grid computing provides computational, visualization and data storage services, by using geographically distributed resources. Some of the grid projects concerns high performance computing and visualization for virtual laboratory applications. Visualization applications running on the grid need to access large amounts of data possibly distributed among the participating sites. Grid data management is an important topic in many grid-related research projects [1,2, 3]. The data obtained during experiments in the Virtual Laboratory (VLAB) or the data being results of simulation or visualization often need to be archived. A Virtual Storage System (VSS) for grid-based accessing, providing the demanded archiving service, is under development for the SGIgrid project [4]. The main goal of VSS is to integrate the mass storage resources residing in the participating computer centers into a common storage service. High Performance Computing (HPC) sites use tertiary storage (like tape libraries and optical jukeboxes) to economically store vast amounts of data. Usually, in such cases the tertiary storage is managed by the Hierarchical Storage Management (HSM) type of software. In the participating sites the DiskXtender HSM software [5] by Legato Systems is used.

Different grid based data management systems for replicated data sets are being currently developed. Storage Resource Broker (SRB) has been developed at San Diego Supercomputing Center [6]. SRB is a client-server middleware providing unified interface for connecting different type mass storage facilities over network. The Reptor system is a prototype of the replica management service developed as a part of the EU DataGrid project [1]. Data Management System (DMS) has been developed as a part of the Progress project [7]. DMS

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 458–461, 2004.

is aimed at providing access to distributed mass storage through integrating the data in a virtual filesystem for the purpose of the computational portal.

The proposed VSS differs from other systems for managing distributed storage resources by its specific functionalities. Each of the mentioned above systems could be used as a base for developing the VSS, by extending the existing systems functionalities.

The rest of the paper is organized as follows: VSS new functionalities are described in Section 2. Some implementation details are given in Section 3. In the last section we conclude the paper and provide some insight into future work.

2 Functionality Details

The VSS provides the following add-on functionalities: data access time estimation, file ordering, replica management, automatic generation and selection of replica, file fragment access, API for the user application, described below.

Access Time Estimation. Access time for data kept in HSM systems can take values from wide range (few miliseconds to tens of minutes). Therefore, it is essential to know in advance the access time for such data, for example for the replica selection algorithm [8]. The HSM access time estimation subsystem attempts to estimate the latency and transfer times for the file, which is eventually going to be requested [9].

File Ordering. The user has ability to order a file, which means to inform the system, when he will need to access the file and how long it will be required. If the file is located on slow media (in the mean of access time), the system forces to transfer it to the fast disks cache and locks it for a given period of time. The next problem is optimization of staging operations for the ordered files, i.e., to select the right moment of issuing a file staging request to the HSM system. Results obtained with *Access time estimation* can be helpful in making the proper selection of the moment to start copying; in order to have some safety margin we compute the *scheduled time* in the following way: $T = order\_time - ETA * X + Y$, where ETA (Estimated Time of Arrival) is the latency time, returned by the HSM estimator, X, Y variables (or functions) describing our safety margin.

Replica management, automatic generation and selection. Replication has two purposes: to increase data safety in the case of destroying or damaging and to increase data availability. In the first case the user has ability to mark a file as "important", which forces the system to replicate this file. In the second case the replication is done automatically.

Selection of optimal replica is based on the local HSM access time estimation and the network transfer rate between the client and the site keeping the required data sets [8].

File Fragment Access. The user has ability to access a specified fragment of the file. This is very useful in a case, when the user needs access to some data in a large, well ordered file. Advantages are the lower latency and the shorter transfer time. Access to file fragments is done by file ordering.

Application Programming Interface. VSS API allows programmers to omit details of specification of communication protocol between client and VSS and focus onto usage of the system. API has been developed for Java programming language.

3 VSS Architecture

DMS [7] has been chosen as a data storage and management system for the task, which is responsible for developing the VLAB virtual laboratory system in the SGIgrid project. In order to keep the project consistent we decided to use DMS as a base for developing VSS.

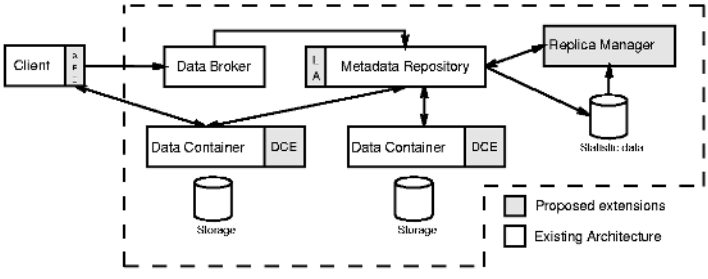

Fig. 1. VSS architecture based on DMS.

In Fig. 1 the DMS-based architecture of VSS is shown. The DMS consists of three main modules: Data Broker, Metadata Repository and Data Container. The Data Broker receives data access requests from the Client, checks permissions, updates the metadata via the Metadata repository module and sends back to the client a handle for accessing the physical data stored on the Data Containers. Metadata Repository keeps the meta data records in a general purpose database. DMS uses the web services technology and SOAP for communicating between components.

In order to develop VSS using DMS as a base, some modules need to be extended with new functionality or new modules need to be added to the architecture. The gray painted boxes (see Fig. 1) indicate the proposed extensions. The DCE (Data Container Extension) module provides access time estimation, file fragment access and file ordering capabilities. The following SOAP methods has been implemeted: estimateFile(), which estimates the acces time for a given physical file; addFileOrder(), removeFileOrder() and updateFileOrder(), which are responsible for managing the file ordering; tranferFile(), which realizes file transfers between data containers. The Replica Manager module is responsible for automatic data replication based on statistical data. The optimization algorithm for replication decides, based on these data, which files have to be replicated. At present, a basic algorithm, taking into account frequency of file

usage and user category, has been implemented. The LA (Log Analyzer) obtains file access statistical information (like the number of references and file transfer performance chracteristics) from DMS logs. It is implemented in the Perl language.

4 Conclusions

In this paper the design and implementation details of the virtual storage system developed as part of the SGIgrid project has been presented. This storage system is aimed at integrating the mass storage equipment installed in the participating sites into a common data archivization service. By using replica automatic generation and selection the system is flexible and suitable for efficient and reliable usage of the distributed storage resources by the grid-enabled applications. The described system is different from the ones being developed in similar projects since it provides additional functionality for the HSM-based storage resources like data access time estimation, file ordering and efficient access to file fragments.

Acknowledgments. The work described in this paper was supported by the Polish Committee for Scientific Research (KBN) project "SGIgrid" 6 T11 0052 2002 C/05836 and in part by KBN project 4 T11C 028 24 and by AGH grant. ACC CYFRONET-AGH is acknowledged.

References

1. "DataGrid – Research and Technological Development for an International Data Grid", EU Project IST-2000-25182.
2. "CROSSGRID – Development of Grid Environment for Interactive Applications", EU Project IST-2001-32243.
3. Dutka, Ł., Słota, R., Nikolow, D., Kitowski, J., "Optimization of Data Access for Grid Environment", presented at 1st European Across Grids Conference, Universidad de Santiago de Compostela, Spain, February 13-14, 2003.
4. SGIgrid: Large-scale computing and visualization for virtual laboratory using SGI cluster (in Polish), KBN Project, http://www.wcss.wroc.pl/pb/sgigrid/
5. Legato Systems, Inc. - DiskXtender Unix/Linux, `http://www.legato.com/products/diskxtender/diskxtenderunix.cfm`.
6. Storage Resource Broker, `http://www.npaci.edu/DICE/SRB/`.
7. PROGRESS, `http://progress.man.poznan.pl/`.
8. Stockinger, K., Stokinger, H., Dutka, Ł., Słota, R., Nikolow, D., Kitowski, J., "Access Cost Estimation for Unified Grid Storage Systems", 4-th International Workshop on Grid Computing (Grid 2003), Phoenix, Arizona, November 17, 2003, IEEE Computer Society Press.
9. Nikolow, D., Słota, R., Kitowski, J. "Gray Box Based Data Access Time Estimation for Tertiary Storage in Grid Environment", 5-th Int. Conf. Parallel Processing and Applied Mathematics, Czestochowa, Poland, September 7-10, 2003, LNCS vol.3019.