ARCO: A LONG-TERM DIGITAL LIBRARY STORAGE SYSTEM BASED ON GRID COMPUTATIONAL INFRASTRUCTURE

Han Fei, Paulo Trezentos, Nuno Almeida, Miguel Lourenço
ADETTI, ISCTE, Portugal
Edificio ISCTE, Avenida das Forcas Armadas,1600-082 Lisboa, Portugall

José Borbinha, João Neves
National Library of Portugal
Campo Grande, 83-1749-081 Lisboa, Portugal

Keywords: Grid Computing, Digital Library, Storage, Large file transfer

Abstract: Over the past several years the large scale digital library service has undergone enormous popularity. Arco project is a digital library storage project in Portuguese National library. To a digital library storage system like ARCO system, there are several challenges, such as the availability of peta-scale storage, seamless spanning of storage cluster, administration and utilization of distributed storage and computing resources, safety and stability of data transfer, scalability of the whole system, automatic discovery and monitoring of metadata, etc. Grid computing appears as an effective technology coupling geographically distributed resources for solving large scale problems in the wide area or local area network. The ARCO system has been developed on the Grid computational infrastructure, and on the basis of various other toolkits, such as PostgreSQL, LDAP, and the Apache HTTP server. Main developing languages are C, PHP, and Perl. In this paper, we discuss the logical structure sketch of the digital library ARCO system, resources organization, metadata discovering and usage, the system's operation details and some operations examples, as also the solution of large file transfer problem in Globus grid toolkit

1 INTRODUCTION

1.1 Background of digital library storage system

With the swift popularity of providing digital content and multimedia service through internet, traditional libraries are in a changing role. The data services of a library provided through internet have undergone fundamental changes not only in style, but also in the quality and quantity of the data. Previously, library network provision was serviced by a single stand alone server. Daily network connections were limited to data volumes of merely several megabytes. But since then, things have changed dramatically.

Nowadays, people are no longer content just to browse and look up some book catalogues and introductions; they want to read digitalized books online, or download the book to convenient movable devices, such as notebooks and handhold book reading gadgets. Meanwhile this digital content are in multi styles, such as text file, static images, and media streams. However, inside the library, all digitalized data need to be stored in devices with huge amount of storage capacity. As a result, administrative software and toolkits for utilizing data storage resources inside the digital virtual library become complex and difficult to develop and uphold. In the storage system of a digital library, data are transferred from one part of the library storage space to another part constantly; new books are digitalized and copied into the library system;

Fei H., Trezentos P., Almeida N., Lourenço M., Borbinha J. and Neves J. (2005).
ARCO: A LONG-TERM DIGITAL LIBRARY STORAGE SYSTEM BASED ON GRID COMPUTATIONAL INFRASTRUCTURE.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 44-51
DOI: 10.5220/0002534000440051

old version are deleted or updated; metadata catalogue need to be set up and updated continually.

BN (Biblioteca Nacional) is the Portuguese national library. BN needs to combine leading information processing techniques and undertake the mission of digital publishing, provide digital books, preserve cultural artefacts, and process metadata, etc. As a result, it became an urgent demand to have a system with a large storage capacity.

1.2 Storage system demand in BN

Each year, millions of users and clients, all over the world, connect to BN through the internet and use the online service provided. The service includes not only simply looking up book catalogue and browsing brief html files, but also online reading of digital books and downloading of digital content for late reading or rendering. All of these data need to be digitally stored somewhere.

With the time passing by, the storage system needs to be not only big enough for old data, but also fully scalable in capacity for the future, because almost every day, big amount of digital data is produced (around 400 GB/day). The scale of the needed storage capacity is petabytes. For the time being, from the view point of mature and viable technique solution, only some kind of storage cluster or server farm can provide storage capacity in this scale.

Nowadays, an entry-range server with a powerful CPU and a large capacity is much cheaper than several years ago (1-2 EURO / Gigabyte).

A dual CPU server with about 800-1200 Gigabytes hard drives (4 hard drives) turn out then to be a proper choice for data storage system. With all of these kinds of computers connecting with each other, there is another challenge remaining unsolved. How to build a system capable of use these resources? The storage demand in BN is not only just simple a matter of scale well, since the storage devices and network connections are just the primary basis of a digital library. Digital library still need administrative software, toolkits and middleware software. All of these items need to be combined together into the functional virtual library. The ARCO project in BN aims at leading edge techniques and solutions, to develop a digital library storage system with computational storage resources and a set of software toolkits.

2 ARCO PROJECT OBJECTIVES AND RELATED TECHNIQUES

2.1 Project objectives

Through the ARCO project, BN want a usable storage system able to handle a huge amount (about several hundreds of Tera Bytes) of digital content.

The ARCO system will be the main reservoir of the digital library in BN. Internally, the storage system need to be constructed based on a layered concept, in order to get better use of already available toolkits and techniques. Any developed part of the system can be re-used in the future. Under various considerations, the following points have been proposed.

The physical basis of the system is a cluster of PCs, without keyboard and monitor. In the experiment and developing stage ARCO has 30 PCs providing a storage capacity of 24 TB.

All these PCs are connected by 1Gigabit fast Ethernet, but the software and the middle-ware will not make any assumption to the speed of network connection. Data sharing should be available between different part of departments inside BN, and between BN and other major libraries in other areas.

To solve these issues, ARCO is based on GRID architecture, because it introduced an abstraction layer to handle with the distributed data storage environment.

The topological structure of the logical digital library will be scalable in several levels to satisfy future demand.

The system administrative operation will be provided in PHP scripted webpages. This is easy to use and convenient to operate over a different locale through network.

There are three interfaces having been defined for operations taken on different logical entities level. The first interface is provided through webpage which is scripted in PHP. The second is provided through command line in the grid computing environment. The third is provided in API (Application Programming Interface). The command line and API interfaces can be re-used by other projects in the future.

The ARCO system provides the grid computational resource automatically discovering feature. All the local resource and hardware attributes are collected and stored in LDAP server directory service, and late are queried through by LDAP clients. The resource attributes can be used in several ways, such as data and workload automatically redistribution or resource usage monitorization.

Metadata creation, querying, and administration will be provided in ARCO system. Metadata are stored in a PostgreSQL database.

ARCO system provides grid resource hardware and network connection monitorization. For example the CPU working environment is discovered through ACPI interface, and some working conditions are continuingly monitored.

To guarantee the system safety and information integration, ARCO system adopts a basic security mechanism, to authorize different administrators with different level of rights to operate on various logical entities of the library storage system.

All of these physical demands and logical concepts have composed of the ARCO system. In this paper we will discuss in details about the logical structure of the system and some examples of system operations.

2.2 Globus grid toolkits

Grid computing provides effective techniques for coupling distributed resources, solving large-scale computational and storage heavily problems over the network. Globus grid toolkits(The Globus Project: Globus Quick Start Guide)(IBM Redbook: Globus Toolkit 3.0 Quick Start Guide), as a middle-ware, manages and uses resources in local or wide area network.

Globus grid toolkits includes mainly three parts(Ian Foster and Carl kesselman, 1999)(Ian Foster et al., 2002). The first is GRAM, the Globus Resource Allocation Manager, which is a basic service that provides capabilities to submit job from one grid node to another node. GRAM unites computational resources in grid environment, and provides a common user interface so that you can submit and administrate jobs on multiple machines on the Grid fabric environment. Without the grid interface, first you will meet with different security login procedure in a hetero-environment; second you need to copy the executable file to the destination machine for later execution; third you will try to start running the job on the remote machine; finally you will need to collect and transfer the computing results or any error message back to your local node. GRAM not only unites computational resources, but also unifies utilization interfaces. When you submit a job, you can configure the job executing environment (i.e. source data, result data, standard output and standard error file) by RSL (Resource Specification Language) language sentence.

The second main part of Globus grid toolkit is for information services, which provides information about grid resources. In Globus toolkit, such utilities include the MDS (Monitoring and Discovery Services)(The Globus Project: MDS 2.1 User's Guide). Without MDS service, you must to write a script or program to collect, analyse, and transfer resource attributes from all grid nodes back to your local node. In the case of Linux, perhaps you need to make a statistic report about proc directory in every computer, and then transfer this statistic information back to your local computer, and finally generalize a report about all resources of the grid cluster. In a grid environment, MDS periodically collect local information and store it into LDAP directory server. You only need to send a LDAP query, and then you will get all these useful information.

The third key part of Globus is for data management and transferring, which includes such utilities as GridFTP and globus-url-copy. The GridFTP and globus-url-copy provide multiple protocol transferring; meanwhile use the same unified security mechanism as other parts of Globus grid toolkit.

3 ARCO SYSTEM STRUCTURE

Looking from different angle and standpoint, we will get different impression to ARCO system.

3.1 Digital Library abstract structure

To get a better understand we can compare the ARCO storage system with the traditional library. In a traditional library, there are catalogue for looking up books in an organized way; in ARCO there are metadata catalogue (PostgreSQL database) to provide a books list, which can be browsed and looked up in several methods. In according to different branch of subjects, the traditional library, which can have a very big area for storing books, have book shelves with specific subject label. Books belonging to one subject will put onto shelves belonging to this subject. The subject and correspondent shelves can be organized in a complex way of several classes.

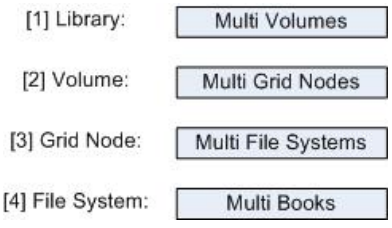

Figure 1: ARCO system abstract structure

Figure 1 is the ARCO system abstract structure. Like the traditional library, ARCO library is composed by multiple volumes. A volume is

composed by multiple grid nodes, which are servers in computing grid environment. But one volume can have a mirror volume to guarantee the information integration. Beside of be a mirror of another volume, a mirror volume is as functional the same as a normal one.

A grid node is composed by multiple file systems. Physically, file systems is a partition of a hard drive, or occupy a whole hard drive, even span over several hard drive in the case of soft RAID or LVM. Finally, books are stored in file system.

But a node can be detached from the grid cluster through a special operation, in which case, all the data reside inside this node will be automatically redistributed to other nodes of the same volume. This is a way for guarantee the information integration of the volume.

In traditional library, a book will have several physical copies. In a digital library as ARCO, mirror copy is needed, but only for the security and information integration reason. The mirror in ARCO is arranged on the volume level. A mirror volume is organized in the same way as normal volumes, only the metadata catalogue of the mirror volume has a record item to say that this volume is a mirror of which volume and the metadata catalogue of the main volume will has a record to show the mirror volume name.

3.2 System interface

Figure 2 demonstrate the layered structure of the ARCO system user interface.

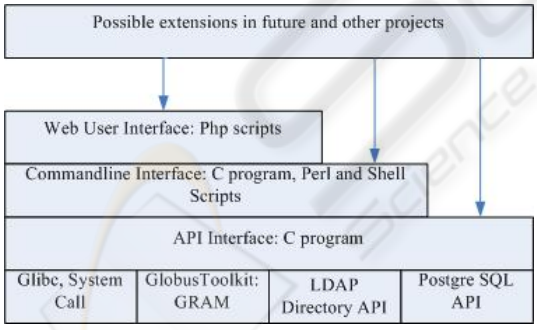

Figure 2: ARCO system multiple interface structure

Looking from the outside, the ARCO system provide three interfaces, which are http protocol web interface, command line interface and API interface. Through web interface users can query the system information or take some operations. Under this case, users are divided into different groups and granted different right to take specific operations. User need to login into the system, then all the available commands and menus to this specific security class will be appear in several web pages. The web page is scripted in PHP and rendered by Apache server. The login and security control is also done by PHP scripts. Whenever any operation is asked by the user, in fact, PHP scripts will call command line interface to do the actual thing.

In ARCO system, command line interface takes full control of the system. It utilizes the security mechanism of operating system, database and Globus grid to check the user's identification. The web page interface provides only preconfigured operations and functions, compared with this, the command line interface are fully free to not only preconfigured operations, but also to any operation scripted on the fly. In the case of some special operations that haven't been configured, the administrator can use command line interface, and write a script (Shell, Perl, Python etc.) to do it. The command line interface can be used in the future for other project or system function expanding.

The API interface is the basis of all the system operation. The entire high level interface will finally call the API interface to fulfil the operation. The API interface can work as a library tool that in the future can be used by other programs.

3.3 User, Group and security definition

The ARCO system will provide a large storage capacity. Digitalized books and media data will constantly flow into the system. From the viewpoint of security, operators and users need to have different level of authorities.

Figure 3 shows the ARCO system security mechanism. There are two levels of security. In the high level, ARCO system works as a storage reservoir and the operations have been defined to operate at this level. There are also many users and administrators. What kind of operations can be taken by which user must be strictly limited in a preconfigured way? The high level security mechanism guarantees the system safety and the information integrity.

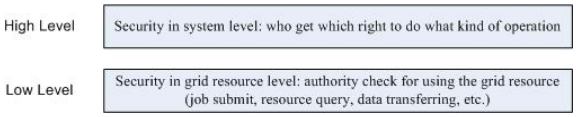

Figure 3: ARCO system security structure

In a summary, the ARCO web interface has adopted independent security mechanism which is scripted in PHP and rendering by Apache server. But the command line and API interface don't provide

this kind of high level security mechanism, which just assume that people get touch to this level will have fully control over not only the hardware but also the system services. The figure 4 is the ARCO system security schema in the web interface. All the ARCO system operations will finally step down into APIs for some kinds of services and resources, and then in here, it is the grid security mechanism takes effects.

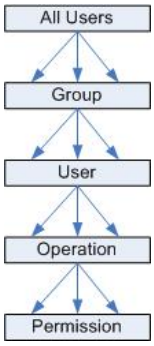

Figure 4: ARCO system security schema

4 SYSTEM OPERATIONS

The ARCO system not only has some static features, such as the system interfaces in various level and zones of volumes and grid nodes and file systems, but also has dynamic snapshots. Dynamically, new works are constantly stored into the system, ephemeral versions are replaced or deleted, new subjects are created, subjects are merged or reorganized, new servers are added into the grid, servers are shutdown and moved out for maintaining reason, accident are taken place in hardware level and software level then fault recoveries are taken place. Dynamic operations and static feature, in together, compose the ARCO system.

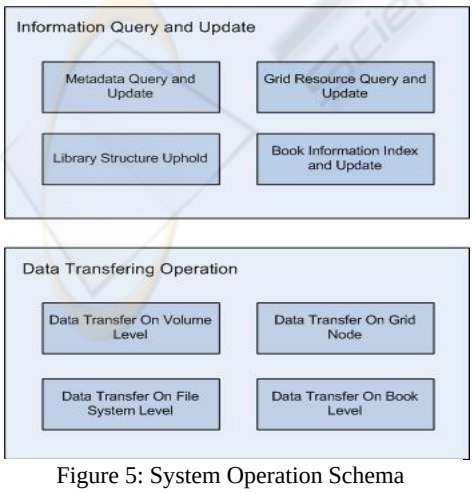

Figure 5: System Operation Schema

Figure 5 shows the operation classes in ARCO system. All the operations can be generally divided into 2 groups, the first is information querying and updating, and the second is data transferring related. The objectives of dividing operations into classes and groups are for a better and clearer logical structure, and to get maximum code reuse.

In ARCO the metadata define in a detailed mode the virtual digital library and parameters of the grid infrastructure resources. All the data transferring operations need first to know not only the usable grid storage and computing resources, but also the logical structure of the library. Metadata querying provides a way to know the library logical structure, which is the same as in traditional library, where the librarian need to know where each bookshelf is and what each bookshelf is labelled. Grid resource query and update is the same concept as the librarian needs to know the bookshelves usage and distribution details in each catalogue section. Library structures uphold is to reorganize the library. Book information index and update provide the book catalogue.

From high level such as a volume to low level such as a single book, the data transferring operations are divided into different groups. Some of the data transferring operations will take long time, operations as these are better to be arranged to execute in batch mode. All transferring operations, complex or simple, finally will be analysed and divided into unit single file transfer. The single file transfer unit are described in JDL (job description language) and write down in a JDR (job description record). In the case of short operation, the JDR will be submitted to execute at once, and the result can be got interactively. To a complex and time consuming operation, all its JDRs of file transfer units are stored in a JDR link list structure, which can be write to a file in hard drive and late read back. The JDR link list will be submitted to execute in batch mode.

The JDR includes not only the digital library metadata and library structure information, but also grid resources details and job executing configuration. The grid resources demand and job executing configuration of the JDR are written in grid RSL (Resource Specification Language).

4.1 Metadata about objects, job description record and batch mode

In ARCO system, metadata is mainly stored in PostgreSQL database. The metadata information is very important, which not only defines the logical structure of the digital library, but also is the

bookkeeping of the content storage. To take a simple example, the metadata is just like the super block and inodes and the journal database of a file system; and the grid storage resource is like the storage block of a file system. Almost all of the digital library operation are tightly related with querying and update the ARCO metadata.

volume: name, id, mirror, mirror-info, date, username, status, log;
grid node: domain-name, ip, volume-id, status, log;
file system: dynamically query from LDAP directory;
book: name, id, size, grid-node-ip, full-path-name, status, date, log;

4.2 Operation example: Insert new book into ARCO system

Qualified operators with administrator rights can insert new digital books or files into ARCO system. When the operator submits an "insert new book" job to ARCO system, some metadata item must be clearly defined: (Inside quotation mark, we compare it with traditional library operation).

- **Book name** (Book name in traditional library);
- **Source server** domain name, full path (Where is the book);
- **Volume id** (Which catalogue section the book belong to? i.e. Society or Law or Science.);
- **Insert** (This should be a new book).

Then the ARCO system will look up the PostgreSQL database to make sure what grid nodes are belong to this Volume and if there is another mirror Volume:

- **Grid nodes** domain name list of **this volume** (Bookshelves belong to this catalogue section);
- **Grid nodes** domain name list of **mirror volume**.

Then we query the LDAP server in each Grid node and get the basic information:
- All available **file systems** and **free space** (The free space in available bookshelves);

Then make choice of one proper server and file system according to specific policy, i.e. largest one first, or the first available first. The next step is to copy the file from the source entry point to the destination. If this volume has mirror, the book will also copy into the mirror volume. If the copy operation is successful then insert the metadata into the PostgreSQL database:

Book name, id, size, grid-node-ip, full-path-name, status, date, log, and operator.

Books, which reside inside a volume, can be indexed, looked up and browsed through web services. Sometimes the operation is to insert a new book, but there is already a book with the same name in this volume. Under this case, the system will send out warning message and do nothing. There is a book update operation in ARCO for update an old version. But the system allows the storage of different book with the same name but in different volumes.

4.3 Operation example: Volume copy

In ARCO system, many basic operations can combine with each other to compose of complex semantics structures and sentences. It is like in our natural language, when we want to express ourselves or generalize complex imagination, we always construct our thinking by using basic idea unites. In ARCO system, volumes don't share grid nodes, so that a volume is logically and physically independent with its copy. After copying, the duplicated volume can be labelled as the mirror of the main volume; otherwise, the volume copy operation can just act as an intermedium step of more complex operations.

A volume can be very large, it can include many grid nodes, each node can have many file systems, and each file system can have many books stored. The volume copy operation is time consuming; it can take very long time to finish, so we have programmed its executing in batch mode.

The first step of volume copy is to query the node name list of the destination volume, and then to each grid node, send LDAP query and get file system details and free space information about file systems. The second step is to make a decision if the total free space of the destination volume can hold all the new files, after that, create a job description record for each book. This record has enough information, so that basic functions of insert a new book operation can be reused in here. All the job description records are stored in a link list structure, the link list can be written to a file and late read back from the file. The job link list is the basis of the batch mode execution. After the file transferring finished successfully, the final step is to update the metadata information about the new volume and all its books.

4.4 Operation example: grid node shutdown

The grid nodes are the physical basis of ARCO storage system. One volume can have multi nodes. Some times a node need to be shutdown and taken out off the grid cluster, under this case, we still need to keep all the data intact, so all books in the shutdown node need to be transferred to other nodes in the same volume.

Firstly, the gird node shutdown procedure need to query and get the node name list of the same volume, after that, send LDAP query and get all the file system details and free space. Because the shutdown procedure can cost very long time, the entire book transferring jobs will be stored as a job description record in a link list structure. Finally, the operation will be submitted to execute in batch mode.

5 BIG FILE TRANSFER PROBLEM AND SOLUTION

Globus-url-copy (version 2.9 at the time of this project) supports several transport protocols such as GridFTP and GASS. It was designed to transfer small files between grid nodes, but some of the files in ARCO system are very big; usually they are media stream files. In our test, GridFTP can not transfer files bigger than 2 Gbytes.

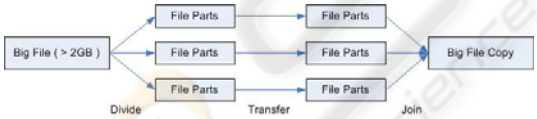

Figure 6: Big File Transfer

Figure 6 shows how we have worked around the big file transferring problem in Globus toolkit. For a big file bigger than 2 Gbytes, firstly, we divide it into smaller file parts with every single part bellow the 2 Gbytes limitation; then, transfer each parts to destination server; finally, join all the parts into a full file.

In ARCO system, some digital books include multi files and stay in a directory tree structure. In theory, we can step down and traverse the directory tree, meanwhile transfer each file and create correspondent subdirectory in destination server, but work in this way is dangerous, and can not keep the information integration. Besides the big file transfer problem, there are other small bugs in Globus version 3.0 (see the release notes in www.globus.org). So we have decided to treat book of multi file or single file in the same way, to tar it into file parts, then transfer file part, finally join file parts. There is an excellent backup tool, Dar(Disk ARchive software), which can backup files or directories into several files of given size.

6 CONCLUSION

Now we have fulfilled the first phase of the ARCO project. The original objective has been realized. For the time being, the ARCO system running on about 30 PCs (dual CPU Pentium 2.4Ghz, 200G x 4 Hard drives, 1G fast Ethernet, without monitor and keyboard). The system has undergone some experiment, such as new books copying into the system, digital library structure re-constructing in different level, security mechanism testing, transferring large files (as large as 40 Gbytes), etc.

Grid technique has become the de facto solution for combining and utilising geographically distributed computing resources. In ARCO system, we have utilized Globus grid toolkit, to provide a integrated and unified storage system for the digital library. For the time being, many other solutions still need to be taken to overcome some limitations and immature parts of the Globus grid toolkit.

From now on it will be much easier to BN store and search for digitalized content, the system has a good abstraction layer (provided by the Grid services) and it's very easy to increase the storage capacity. The cost of this system is low, because it uses desktop computers with IDE hard disks (less than 500 € each) and in case of failure it's always possible to use the volume mirror.

Finally we can conclude that ARCO is a simple and cheap solution to handle large TB of digital content.

REFERENCES

The Globus Project: Globus Quick Start Guide
http://www.globus.org/toolkit/documentation/QuickStart.pdf

IBM Redbook: Globus Toolkit 3.0 Quick Start Guide
http://www.redbooks.ibm.com/redpapers/pdfs/redp3697.pdf

The Globus Project: MDS 2.1 User's Guide
http://www.globus.org/mds/mdsuserguide.pdf

Disk ARchive software
http://dar.linux.free.fr/

Ian Foster and Carl kesselman, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999

Ian Foster, Carl kesselman, Jeffrey M. Nick, and Steven Tuecke, Grid Services for Distributed System Integration, Computer, 35(6), 2002

Global Grid Forum Documents and Recommendations: Process and Requirements, GFD-C.1, C.Catlett

Viktor Berstis, Fundamentals of Grid Computing, IBM Redbook.

Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke, "The Physiology of the Grid – An Open Services Architecture for Distributed Systems Integration", Draft document, version: 6/22/2002

S. Tuecke, K. K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, P. Vanderbilt, "Open Grid Services Infrastructure (OGSI) ", Global Grid Forum, Draft document, version 1.0, 5/4/2003

António Serra, Paulo Trezentos, Carlos Serrão, Miguel Dias, "Parallel Jpeg2000 Enconding On A Beowulf Cluster"

Han Fei, Paulo Trezentos, Nuno Almeida et Al, "Enabling Queries Using Grid-brick Approach - A Distributed Data Storage Architecture", (2002)