DATA MANAGEMENT IN GRID COMPUTING ARCHITECTURE (A Case Study)

Atif Ata[1], Prof. Naeem ul Hassan Janjua[2]

[1]*MSCS SZABIST, Karachi*

[2]*Bahria University, Karachi*
njjanjua@yahoo.com

Abstract: *Data management is one of the key features of a Data Grid where large amounts of data are distributed and/or replicated to remote sites, potentially all over the world. In general, a Data Grid needs to provide features of a pure computational Grid (resource discovery, sharing etc.) as well as more specialized data management features like replica management which is the main focus of this article.*
The aim of this study is to have a look at data management in grid computing architecture, more specifically data replication. Finally some data management models and systems are discussed
Eventually among many data management systems two most commonly used systems are studied and after the evaluation of their performance they are compared.

Keywords: *Grid Computing, Data Management, Replication, GDMP, edg replica manager.*

1. INTRODUCTION

Grid computing and inspiration behind computational and Data Grids. Moreover it also briefly discusses current Data Grid Technologies.

1.1. Introduction

Grid Computing is a word that has become one of the buzzwords in the IT industry by now. Grid computing is a new and modern approach of modern IT age that leverages the current IT infrastructure to make the most of present resources as well as managing data and computing workloads.

Grid computing can be defined in the words of Gartner
“A grid is a collection of resources owned by multiple organizations that is coordinated to allow them to solve a common problem." [1]

2. GRID COMPUTING AND DATA MANAGEMENT IN GRID ARCHITECTURE

2.1 Introduction

GRID computing in general, how this technology evolves, what are the components that can be termed as building blocks of grid computing. Moreover, it also discusses the Virtual Organizations; which is an important part in enabling GRID Computing. And finally the importance of Data Management in GRID Computing is taken into account.

2.2 Background

The need of Grid was initially felt by large-scale, resource intensive applications, as these applications need more resources than present in a single computing unit, whether it is a computer, a workstation, or a cluster in a single administrative domain.

There also is a need of the output or resulting data to be stored so that it may be used for further analysis and sharing with other researchers. Here comes grid computing as a computing paradigm. It enables the aggregation of large scale computing, networking resources and storage. Grid enables a distributed community to share its resources across different domains.

There are many areas in which amount of data is very large that need to be collected and appropriately stored in data centers that are geographically distributed. Some experiments even produce a petabyte of data in a year.

2.3 OGSA

After the development of GGF (Global GRID Forum) in the late 1990s, researchers provide Open Grid Services Architecture (OGSA) that integrates Globus and Web services approaches. OGSA (Open Grid Service Architecture) discusses the core services for many areas of grid computing.

In the last decade, the Grid community presents a layered model that shows the integration between the services provided on the GRID with the resources. This model

provides a layered abstraction of the Grid. Figure 2.1 illustrates the Community Grid Model being developed.

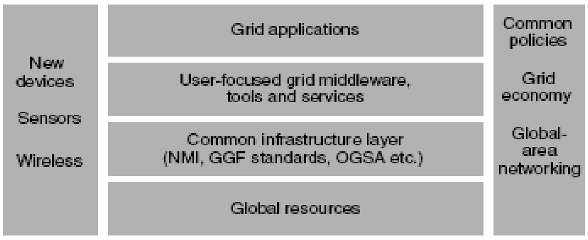

Figure 2.1: OGSA (Open GRID Service Architecture) [2]

2.4. Evolution

“The term Grid computing originated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid” [3].In 1999 CPU scavenging and volunteer computing were popularized. Ian Foster, Steve Tuecke and Carl Kesselman brought the idea of of the grid web services. They were regarded as the “Fathers of the Grid” as they led the effort to create the Globus Toolkit incorporating computation management. There are many other tools that were built to answer some subset of services required to build a global grid.

Recently in 2007 a new terminology Cloud Computing came into popularity. This concept has some similarity to grid computing that is why grid computing is often associated with cloud computing systems but this is not the case always.

2.5. Data Management and Replication

In a typical Data Grid, large amounts of data that are stored in read-only files need to be replicated in a secure and efficient way. The major data management and replication requirements are:

Secure and efficient file transfer
The files are to be copied between many data stores, which are located at distributed sites.

Need for Replica Catalogue
Identical copies of the files exist; so that replicas need to be identified through logical and physical file names.

Replica Management Service
The combination of file transfer with file cataloguing presents it as an atomic transaction to the user.

Interaction between replica management & storage service
Large data stores use secondary and possibly tertiary storage devices.

3. DATA MANAGEMENT AND REPLICATION

3.1 Data Grid Architecture

Globus team was among those effort making units that made a leading effort. The main services offered by this unique framework are as follows:

- Security
- Information Services
- Resource Management
- Data Management

Security

The toolkit mainly contains a Grid Security Infrastructure (GSI). This infrastructure gives a set of security features.

Information Services

The information regarding the status of Grid Resources is provided by the Information Services. This service uses an approach of notification board where all resources write their status.

Resource Management
This is a very component of Grid Architecture that uses the inputs from the information services and enables users to request and use the recourses.

Data Management
This component allows the access and manages the data and data resources in grid computing architecture

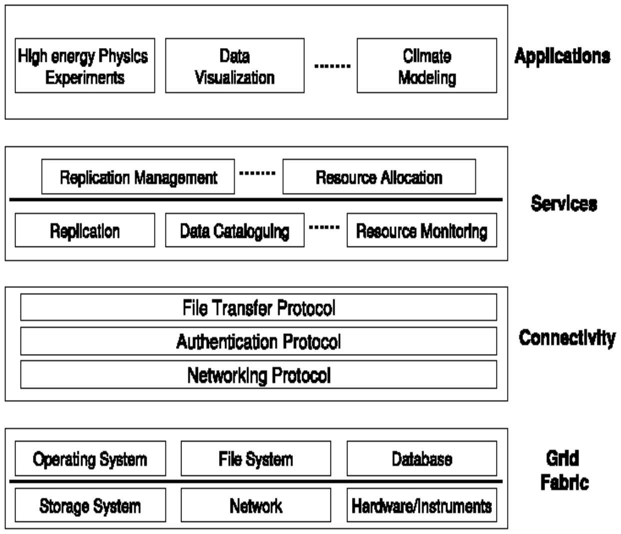

Figure. 3.1. Overview of Grid Architecture [4]

3.2. Importance of Replication

When we talk about the importance of replication in data grid we come to a conclusion that replication is a

very important part of data grid as far as the data management in the data grid is concerned. Data replication is used to reduce access latency and bandwidth consumption in data grid that is why it plays a vital role in data management in grid computing.

4. DATA REPLICATION MODELS

4.1 Replication Models

The most important challenge in data replications systems is the handling of updates at multiple copies simultaneously and the maintenance of the same view of all replicas all the time. Mostly updates affect the application and replica metadata. Replica metadata provides a mapping between a file name and one or more physical locations of the file replicas. Therefore it is necessary to make it sure that the updates are eventually propagated.

We have three models for replication. These models have been practically used. These are:

- Master-Slave
- Client-Server
- Peer-to-Peer

4.1.1. Master-slave

In the master-slave model there is one replica master as obvious from its name and all other are replica slaves. It works on the ideology that slaves must always be synchronized to master. This is a very simple model to understand. Most of the master-slave services ignore all updates or modifications, performed at the slave and makes the slave identical to the master as soon as the synchronization takes place omitting any change in the slave.

4.1.2. Client-server

The client server model has great resemblance with the master slave model as it also has one server that serves multiple clients. In this model data modifications and updates can also be generated at the client. However the clients are not allowed to communicate or sync with each other directly. They must communicate with each other by means of a server. The disadvantage of the model is that failure of a server can isolate all clients served by it.

4.1.3. Peer-to-Peer

All replicas are equal in this model that is why they are called peers. In this model any replica can get synchronized with any other replica without the involvement of a server and any file system modification or update can be applied at any accessible replica.

4.2 Synchronization

In both weak and strong consistency models updates that are made to any replica must be brought into knowledge of other replicas. These updates are propagated at the same time to other replicas or later reconciled with updates at other locations.. In a typical grid environment, the larger sizes of the data items increase the latency. The communication patterns affect the performance of the process of reconciliation along with the physical location of the replicas.

So synchronization and the algorithms to synchronize replicas effectively plays an important role in a grid environment

5. DATA REPLICATION TOOLS

5.1 GDMP - Grid Data Mirroring Package

This tool is one of the initial attempts which were initially started with the collaboration of CMS. It uses a host subscription method to make replicas of data at different storage elements. The working of the tool and its interaction can be understood by the figure 5.1.

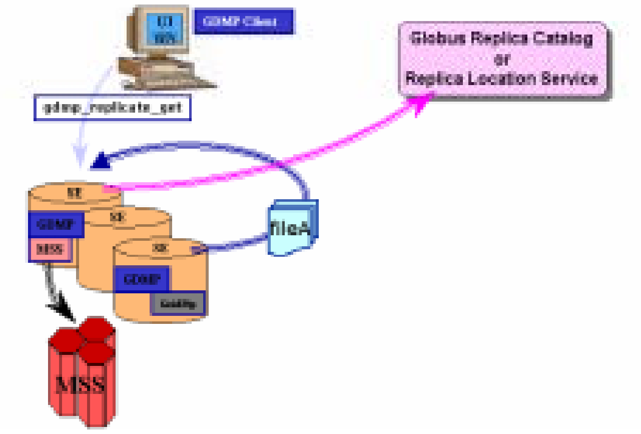

Figure 5.1: File replication/mirroring with GDMP [5]

Enhancements and improvements have been brought into GDMP in several years. Following are some features of GDMP.

- Scalable and stable architecture:
- Reliable replication:
- Retries on error:
- Checks after file transfer:
- Complex server side logging:
- User control over file transfer:
- Availability of Back-ends:
- More Steps for replication:
- Difficult configuration:
- No space management provided:
- Error messages not always clear
- Errors recovery

5.2 Edg-replica-manager

edg replica manager is client side tool. It has replica Catalogue using which it allows registration and replication of files.

Information service helps edg replica manager to find out storage locations on a give storage element. The edg replica manager finds out where to store files of the particular virtual organization a user belongs to the functionality and interaction of edg replica manager is highlighted in figure

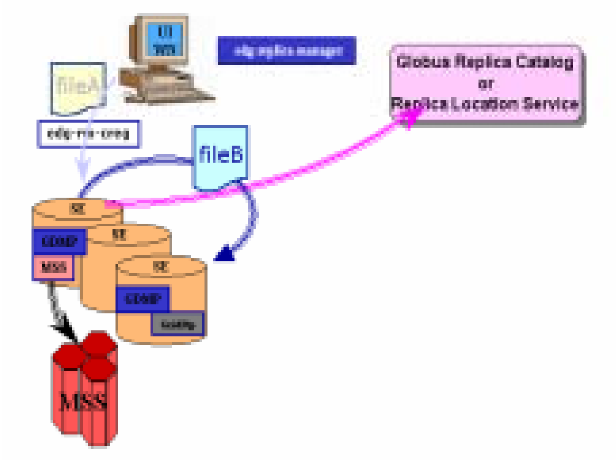

Figure 5.2: edg-replica-manager on the EDG Testbed Similar to GDMP[6]

Following are some features of edg replica manager:

- Functional:
- Third party transfer:
- GSI authorization:
- Easy configuration:
- Steps for replication:
- Error messages
- No roll-back; no transactions:
- No complete interface:

5.3 Comparative Analysis of GDMP and edg-replica-manager

The two tools discussed above are analyzed by taking some replication attributes in Table 5.1. This tool can assist in deciding which tool is better for a particular replication requirement.

Attributes	GDMP	edg-replica-manager
Replication Nature:	In GDMP Replication takes place between Storage Elements Only.	In edg replica manager replication takes place between Storage Elements, User Interface or Computing Elements to Storage Elemnts
Number of Files	GDMP replicates sets of files.	edg replica manager replicates single file.
Interface	GDMP provides MSS interface	edg replica manager uses GDMP's interface
Replication Model	Client-Server	Client Side only
check summing after file transfer:	GDMP does file check summing after file transfer.	edg replica manager does not provide any file check summing.
Complex server side logging:	Server side logging is present in GDMP as it is based upon client server	edg replica manager is client side only therefore there is no server side

	model.	logging.
Automatic Retries	GDMP provides retries on error.	edg-replica-manager doesn’t provide retries.
Users control over file transfer:	User can control file transfer in GDMP.	User cannot control file transfer in edg replica mnager.
Availability of Back-ends:	Back end support present in GDMP.	Back end support not present in edg replica manager.
Steps for replication:	Several steps required for replication in GDMP.	Lesser seteps required in edg replica manager due to its reduced functionality.
Configuration Difficulty:	Configuration is Difficult in GDMP	Configuration is easier in edg replica manager as it provides limited functionality.
Space management:	Space management is not done by GDMP	Space management is not done by edg replica manager
Error messages:	Error messages are not always clear in GDMP	Error Messages are not always clear in edg replica manager

Table 5.1 Comparison: GDMP versus edg-replica-manager (Source: Self)

6. CONCLUSION

It is very much clear from the table shown in the last chapter that GDMP (which is a client server tool) provides more functions than that of edg replica manager. So we can conclude that it depends upon the replication need that we have in different scenarios that what tool we have to use. If we need a simple tool that is easy to configure and need less functionality than edg replica manager is a good choice for us, otherwise is a more functionality is needed than we have to compromise with a more complex tool that is GDMP, that provides more functionality but same time is a bit complex and takes several steps for the replication of files because of its complex nature.

7. FUTURE RECOMMENDATIONS

Tools and carried out a comparative analysis between these two we can say that need of a moderate tool is there which may provide more functionality and on the other hand is also not that complex to configure.

A tool with the ease of use edg replica manager and with the set of functionalities of Grid Data Mirroring Package will be the need for future.

Though, it depends upon the need of the user whether he needs functionality or ease of use. He has to now to compromise one at the cost of other. A tool with both the features will provide a big deal in this regard.

REFERENCES

[1]http://support.sas.com/rnd/scalability/grid/
[2]http://www.grid2002.org/grid2002sample/chapter1.pdf
[3] The Grid: Blueprint for a new computing infrastructure (Ian Foster and Carl Kesselmans Seminal)
[4]Grid Data Mirroring Package (GDMP) by Heinz Stockinger (Publisher: IOS Press)
[5]http://www.gridpp.ac.uk/papers/chep03_TUAT007.PDF
[6]http://www.gridpp.ac.uk/papers/chep03_TUAT007.PDF