A Data Grid Prototype for Distributed Data Production in CMS

Mehnaz Hafeez[a,b], Asad Samar[a], Heinz Stockinger[b,c]

(a) California Institute of Technology, Mail Code 256-48, 1200 E. California Blvd., Pasadena, CA 91125, U.S.A.
(b) CERN, European Organization for Nuclear Research, CH-1211 Geneva 23, Switzerland
(c) Inst. for Computer Science and Business Informatics, Univ. of Vienna, Rathausstr. 19/9, A-1010 Vienna, Austria

Abstract. The CMS experiment at CERN is setting up a Grid infrastructure required to fulfil the needs imposed by Terabyte scale productions for the next few years. The goal is to automate the production and at the same time allow the users to interact with the system, if required, to make decisions which would optimise performance.

We present the architecture, design and functionality of our first working Objectivity file replication prototype. The middle-ware of choice is the Globus toolkit that provides promising functionality. Our results prove the ability of the Globus toolkit to be used as an underlying technology for a world-wide Data Grid. The required data management functionality includes high speed file transfers, secure access to remote files, selection and synchronisation of replicas and managing the meta information. The whole system is expected to be flexible enough to incorporate site specific policies. The data management granularity is the file rather than the object level.

The first prototype is currently in use for the High Level Trigger (HLT) production (autumn 2000). Owing to these efforts, CMS is one of the pioneers to use the Data Grid functionality in a running production system. The project can be viewed as an evaluator of different strategies, a test for the capabilities of middle-ware tools and a provider of basic Grid functionalities.

INTRODUCTION

In autumn 2000 distributed High Level Trigger (HLT) studies take place at both sides of the Atlantic. Regional Centres in France, Great Britain, Italy, Russia and the USA will produce data that have to be replicated to and from CERN as well to/from other Regional Centres. Based on this requirement we have developed a software tool called Grid Data Management Pilot (GDMP) that supports data replication in a Data Grid environment. The first production prototype is used by the CMS experiment to transfer Objectivity [9] database files between Regional Centres.

GDMP can be seen as a mature production software as well as an evaluator of existing Grid technologies.

Let us elaborate on the aspect of being an evaluator. The concept of a Grid [2] system is rather new in the HEP community and has its roots in the Particle Physics Data Grid [7] as well as currently started Grid projects like GriPhyN [5] and DataGrid [1, 6]. Up to now, no HEP experiment has been using Grid tools based on the Globus [3] middle-ware in order to do data production in a distributed way. We consider this as a pioneer step of CMS in the direction of Data Grids in production systems. Our aim has been to take as much as possible of the tools provided by Globus in order to have a replication system that is based on a single middle-ware system.

DISTRIBUTED DATA PRODUCTION IN CMS

CMS has planned four trigger levels with increasing complexity and sophistication in the trigger algorithm. These trigger levels are called level-1, 2, 3 and 4. In some cases triggers at level-2, 3 and 4 are grouped together and called the *higher level triggers (HLT)*.

At LHC, data will be collected by the CMS detector where level-1 trigger has only a small latency protected by a pipeline. The trigger rate at level-1 is around 1 billion events/sec. It is not possible to store all these data, so the HLT will reduce the rates to manageable numbers. Firstly, at level-1 the trigger rate is reduced from 1 billion events/sec to 100,000 events/sec. After level-2 and level-3 the trigger rate is reduced to 100 events/sec. The studies of HLT using simulated data requires large datasets. One million simulated events after level-1 is equivalent to 10 seconds of LHC running.

For the HLT studies, the CMS collaboration has organised five groups according to the physics channels namely: egamma, muon, jetmet, btau and level 1 trigger. For the current production more than 6 million events are requested for the simulation. This corresponds to roughly 6 TB of data. A number of collaborating CMS institutes is involved in the current production such as: INFN, Caltech, IN2P3, Fermilab, CERN, Helsinki Institute of Physics, Moscow, Bristol and Pakistan.

The CMS data production consists of following steps:

1. simulation of the detector response for a given physics channel,
2. digitisation of the above data with event pile-up,
3. reconstruction of physics objects such as tracks, clusters, jets, etc. using the data from step 2.

The preparations of distributed data production have been going on since last year. A data transfer production system based on Perl scripts, HTTP and secure shell copy transfers has been put in place [10]. The purpose of these scripts is to allow a user to see the contents of an Objectivity's federated database catalogue from anywhere with WWW access, without the need to run any part of the Objectivity software locally. Furthermore, the script software allows for data transfer of not only Objectivity but any kind of file, for instance Zebra-fz or Root files.

These Perl scripts have been the initial input for the architecture and design of GDMP. The idea was to extend the functionality by introducing a security environment and providing a fully automated replication mechanism based on a Grid infrastructure using Globus. The current architecture of GDMP presented in this paper is based on Objectivity, i.e. GDMP is an asynchronous data replication tool for replicating Objectivity files over the wide area network.

GDMP can be seen as a Grid-enabled successor of the initial Perl scripts. However, GDMP does not replace the initial software system completely. Since GDMP is currently based on Objectivity's native file catalogue to initiate file replication, it is only used for replicating Objectivity files. The Perl scripts are deployed for fz-file transfers and are still a fall-back solution to GDMP. In a future release, GDMP will incorporate a replica catalogue [4] provided by the Globus toolkit which allows for flexibility in file transfer. Once the replica catalogue is used, GDMP will support file replication for any kind of file and can replace the initial software based on Perl completely.

THE GDMP ARCHITECTURE

GDMP is a multi-threaded client-server system that is based on the Globus toolkit. The software consists of several modules that closely work together but are easily replaceable. In this section we describe the tools used from Globus as well as the modules and the software architecture of GDMP. The core modules are Control Communication, Request Manager, Security, Database Manager and the Data Mover. An application which is visible as a command-line tool uses one or several of these modules.

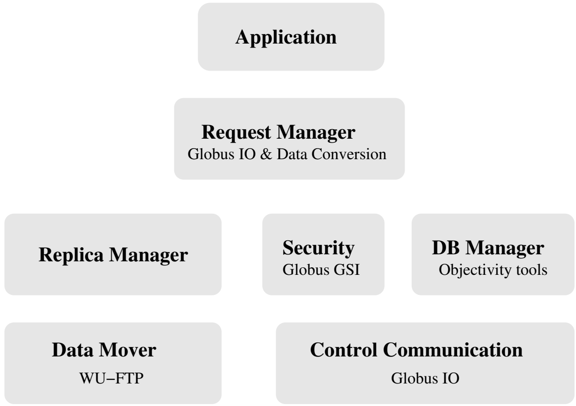

FIGURE 1. Logical datamodel of the tool

Since GDMP is focusing on the Data Grid functionality, only the following limited set of the entire Globus toolkit is used by GDMP.

- Globus GSI for security
- Globus IO and data conversion for communication
- Globus thread library for the multi-threaded GDMP server

Figure 1 shows the current architecture and all the modules of the software. A detailed description of the modules can be found in [8]. All the software modules are written in C++ and run on Solaris 2.6, 7 and Linux RedHat 6.1.

The GDMP server is a daemon constantly running on sites which produce data or want to export their data to other sites.

The server itself uses the communication module for receiving requests from application clients and a thread pool to handle multiple clients concurrently. For each client one thread is used.

The server uses a dedicated server certificate and thus a proxy on its own which is included in the GDMP software distribution. Thus, when a server is started, the required Grid proxy is gained automatically.

Since all the access to data has to be done in a secure environment, a client has to be authorised and authenticated before it can request a service from the server. We use the *single login* procedure which is available through Globus, i.e. once a client has successfully got the proxy on one machine, it can send requests to any server without any further password entered (provided the local client is authorised to access the server).

REPLICATION POLICIES, DATA MODEL AND APPLICATIONS

Currently, GDMP is restricted to replicate only Objectivity files due to the use of the native Objectivity federation catalogue to handle files in GDMP. We will replace the Objectivity file catalogue by the Globus Replica Catalogue [4] in order to support a flexible replication model.

The GDMP architecture is based on the subscription model where each site that wants to get notified about changes at other sites subscribes to a remote site. In principle, a site, where Objectivity files are written, has to trigger the GDMP software which notifies all the "subscriber" sites in the Grid about the new files. In detail, a data production site announces newly written data by publishing its catalogue.

The "subscriber" (destination) sites receive a list of all the new files available at the source site and can determine themselves when to start the actual data transfer. The data transfer is done with a WU-FTP server and an NC-FTP client.

In principle, a site only needs three commands to participate in the automatic replication process. A site can subscribe to another site by issuing the command `gdmp_host_subscribe`. A production site announces that new files are available by using the tool `gdmp_publish_catalogue`. The consumer can then decide when to start the file transfer from the producer to the consumer site with the tool `gdmp_replicate_file_get`.

GDMP allows a partial-replication model where not all the available files in a federation are replicated. This is achieved by applying a filter on the file catalogue. This allows for a partial replication model where the producer as well as the consumer can limit the amount of files to be replicated.

FAULT TOLERANCE AND FAILURE RECOVERY

In the current version of GDMP, each site is itself responsible for getting the latest information from any other site in case of a site failure. A site can recover from the site failure by issuing the command `gdmp_get_catalogue` and receiving the entire catalogue information from another site.

In case of a broken connection, re-sending of several files is not needed since as soon as a file arrives safely at the destination site, the file is attached and the file entry is deleted from the import catalogue immediately. Only the file which is currently been sent when the network connection breaks, has to be resent. Since the implementation of WU-FTP has a “resume transfer” feature, not even the entire file has to be transfered but only the part of the file that is still missing since the last check point in the file. This allows for an optimal utilisation of the bandwidth in case of network errors.

CONCLUSIONS

We have been developing a file replication tool that allows for secure and fast data transfers over the wide-area network in a Data Grid environment. With our production software we have proved that Globus can be used as a middle-ware toolkit in a Data Grid. This has been a pioneer step in the direction of a Data Grid and to the best of our knowledge first software approach where a wide-area replication tool based on Globus is used in a production system. Furthermore, this work can also be regarded as an evaluator of Grid tools and thus has valuable input for other Data Grid activities like DataGrid, PPDG and GriPhyN.

REFERENCES

1. The CERN DataGrid Project: http://www.cern.ch/grid/
2. Ian Foster and Carl Kesselman (editors), The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, USA, 1999.
3. The Globus Project: http://www.globus.org
4. The Globus Data Grid effort: http://www.globus.org/datagrid/
5. The GriPhyN Project, http://griphyn.org
6. Wolfgang Hoschek, Javier Jaen-Martinez, Asad Samar, Heinz Stockinger, Kurt Stockinger, Data Management in an International Data Grid Project, to appear in *1st IEEE, ACM International Workshop on Grid Computing (Grid’2000)*, Bangalore, India, Dec. 2000.
7. The Particle Physics Data Grid (PPDG), http://www.cacr.caltech.edu/ppdg/
8. Asad Samar, Heinz Stockinger. Grid Data Management Pilot (GDMP): A Tool for Wide Area Replication, to appear in *IASTED International Conference on Applied Informatics (AI2001)*, Innsbruck, Austria, February 2001.
9. Objectivity Inc., http://www.objectivity.com
10. Tony Wildish. Accessing Objectivity catalogues via the web. http://wildish.home.cern.ch/wildish/Objectivity/scripts.html