A model to implement a files and replicas system in Clarens

Marko Petek[1], Diego da Silva Gomes[1], Conrad Steenberg[2], Cláudio F.R. Geyer[1], Tiarajú Asmuz Diverio[1], and Alberto F. S. Santoro[3]

[1] UFRGS, Porto Alegre, RS, Brazil
petek@inf.ufrgs.br, dsgomes@inf.ufrgs.br, geyer@inf.ufrgs.br, diverio@inf.ufrgs.br,
[2] Caltech, Pasadena, CA, USA
conrad@hep.caltech.edu,
[3] UERJ, Rio de Janeiro, RJ, Brazil
alberto.santoro@cern.ch

Abstract. The area of High-Energy Physics deal with a huge amount of data. New experiments being built will produce very large files. Grid computing offers distributed storage and computational power enough to deal with those files. The work presents a model to implement files and replicas system in Clarens, a grid middleware developed in the California Institute of Technology to be used on the analysis of data to be gathered by the Compact Muon Solenoid experiment being builted in the Cern.

1 Introduction

Grid computing is an important trend in modern distributed, high performance computing. Since its proposal by Foster and Kesselman [1], Grid computing has been leveraged by researchers in several scientific disciplines.

Among the several areas with research going on to the use of grid computing, one of the main ones is the High Energy Physics (HEP).

The biggest HEP laboratory in the world is the European Organization for Nuclear Research (CERN) in Geneva. A new experimental facility, the Large Hadron Collider (LHC) [2] is being constructed which is scheduled to begin data taking in 2007. The LHC will contain four different experiments or detectors. One of them, the Compact Muon Solenoid (CMS) [3], will be used by a collaboration of scientists from 36 countries, including Brazil.

CMS is designed to produce an extremely high volume of data. Current estimates put this number at around 5 Peta bytes/year. This data will be made available to researchers for further analysis. Due to the sheer volume of data, innovative solutions to the problem of storage and movement for use in Physics analyzes needs to be explored.

The following of this papers develops as follows. Chapter 2 gives a background on the current technologies and resources researched. Chapter 3 states the objectives of the work. Chapter 4 introduces the proposed model. In Chapter 5 the current status of the implementation is presented. Finally Chapter 6 is the conclusion of the paper.

2 Background

To manage the data torrent produced, a multi-tiered computational architecture, devised by Caltech, was adopted by CMS. This architecture relies heavily on the use of Grid computing technologies for data production and analysis. One of the components of this Grid in the U.S. is the Clarens [4] Grid-enabled web services [5] toolkit, which allows secure, high-performance access to computing resources using widely used Internet protocols.

One of the main features of the Clarens middleware implementation is that it performs well even on modest hardware, and scales very well with added CPU, memory and storage resources. However, because it is relatively new, Clarens still lacks some services. Among these are data management services, including replica, metadata and file movement services.

Besides Clarens, another component developed by Caltech is MonaLisa [6]. It is a distributed monitoring system built using Java/Jini technology. MonaLisa is widely used in the CMS to monitor resources, and to distribute and manage the resultant information.

Due to the previously cited high volume of data, an efficient file and dataset replica system is an important part of the computing model. Data replica systems allow the creation of copies, distributed between the different storage elements on the Grid. In the HEP context, the data files are immutable except in extremely rare cases. This eases the task of the replica system, because given sufficient local storage resources any given dataset only needs to be replicated to a particular site once. Concurrent with the advent of computational Grids, another important theme in the distributed systems area that has also seen some significant interest is that of peer-topeer networks (p2p) [5]. P2p networks are an important and evolving mechanism that facilitates the use of distributed computing and storage resources by end users. The main differences between grid computing and p2p networks are in the necessity of user intervention, fault tolerance and user numbers. In fact, Foster sees the convergence of Grid computing and p2p networking as inevitable [7].

Now in its third generation [5], p2p networks are being widely researched as an aid in data location and distribution. The main characteristic of the third generation p2p networks is the use of Distributed Hash Tables (DHTs) for content indexing, using completely non-centralized overlay networks. Building a Replica

Location Services (RLS) based on p2p and DHTs have been proposed suggested by [8].

Some of the problems caused by the data volume of CMS was already cited before. Even if a replica with a low transmission cost is located, it might be advantageous to make use of multiple replicas in order to minimize transfer times.

One common technique to achieve faster file downloads from possibly overloaded storage elements over congested networks is to split the files into smaller pieces. This way, each piece can be transferred from a different replica, in parallel or not, optimizing the moments in that the network conditions are better suited to the transfer. A popular protocol to handle those transfers is the Bit Torrent [9].

There is still the storage problem. New mechanisms to support distributed storage are being developed. On this moment, the Clarens team is evaluating the dCache/SRM [10], and the Logistical Backbone/LSTORE [11]. All them have positive and negative points.

The Logistical Backbone (upon which L-STORE is built) stores the data in a transparent way in several distributed servers. The same file may be partitioned in several servers without any knowledge of the end-user. The use of Monalisa to monitor the Logistical Backbone to optimize data placement taking into account network conditions is a possible area of future study.

Finally, the motivation for this work is to develop a system that allows users to easily find CMS data based on several criteria, bringing the concept of "views" of the database area to file management. Users should be able to use the system without having to issue complex database queries in a way that fits the CMS data organization and naming conventions. To do this they will use a web interface with a familiar look to them. Besides the web interface, a set of Application Programing Interfaces (APIs) will be developed to allow programmatic interaction with the "file system".

3 Objectives

The main objective of this proposal is the development of a Dataset Location System in the context of the CMS based on Clarens middleware.

The work will consist of the following:

1. Creation of replicas.
2. Development of a system for replicas transfer (RFT) to Clarens.
3. Development of a system for replicas location (RLS) to Clarens.
4. Use of Monalisa as a monitor and support tool.
5. Development of a system for file transfer in pieces (Torrents) on computational grids with interfaces for several persistency elements.

6. Development of a metadata query system (visions) of Datasets based on web interfaces.

4 System architecture

The figure below shows the proposed model:

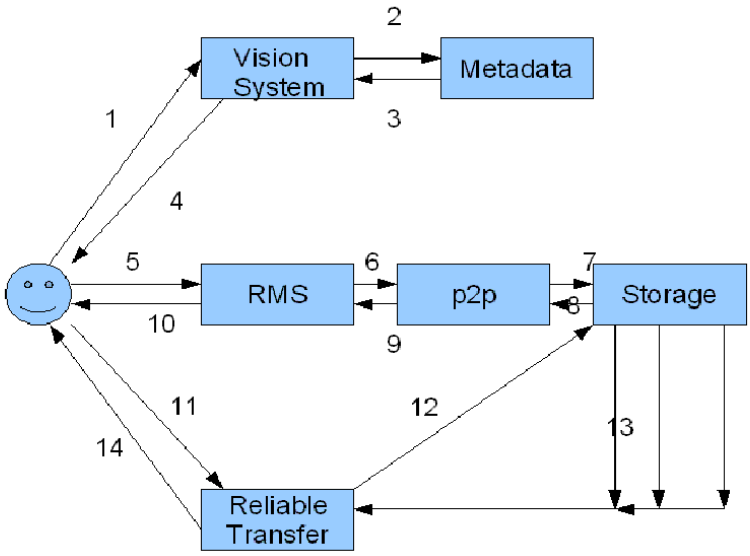

Fig. 1. Proposed Model

When a User wants to search and receive a dataset, the sequence of activities is the following:

1. A search is sent by the User to the Vision System. In this search the User specifies the parameters that the searched datasets must follow. Those parameters may be specified on a strict way (match exactly the parameter value) or by a search interval;

2. The Vision System queries the Metadata System;

3. The Metadata System returns the results to the Vision System;

4. Based on the result gotten from the Metadata System, the Vision System returns to the User informations about the datasets that fill the required

parameters;

5. The User asks for the Replicas System the localization of the desired dataset;

6. The Replicas Systems communicates with an underlying p2p Network sending the unique identification of the dataset (ID). The Replica System doesn't know the physical location of the dataset, only its ID;

7. The p2p Network finds the Storage Element(s) where the desired replicas are stored;

8. The Storage Element(s) return informations about the physical localization of the dataset and others that are need for the transfer to the p2p Network;

9. The p2p Network sends its information to the Replica System;

10. The Replica System returns the informations to the User;

11. The User asks the Reliable Transfer System for the wanted dataset;

12. The Reliable Transfer System sends one or more solicitations to the Storage Element(s) for the transfer of the dataset;

13. The Storage Elements transfer the dataset to the Reliable Transfer System;

14. The Reliable Transfer System delivers the dataset to the User.

Even if an optimization is possible, sending information straight among the Storage Element and the User and possibly in other points, it was decided to use an architecture with bi-directional communication through the different modules for reasons of implementation simplicity and layers isolation.

4.1 Discussion

1. Creation of replicas.

2. Development of a system for replicas transfer (RFT) to Clarens.

3. Development of a system for replicas location (RLS) to Clarens.

These components are already under development as part of the the M.Sc. dissertation of Diego da Silva Gomes [12]. The work exploits characteristics of the Clarens based HEP Grids, mainly the existence of "super-peers", that are the nodes where Clarens server instances are running. It uses an hybrid concept, where the searches inside a Virtual Organization (VO) are done by "diffusion".

The RLS must return not only the "best" replica, but a list with several of the best ones so that the file transfer tool may download pieces of any of these for both performance and redundancy purposes.

The use of "inverted files" is being studied as a way to solve the limitation that the DHTs shouldn't sort the table keys logically, due to the effect of the hash algorithm. Inverted Files are tuples (content, key) that point to one main file based on the content of one of its fields. For instance, an inverted file could have the tuples "newman","dataset1" ... "newman", "dataset5" pointing to all datasets belonging to the "newman" researcher [13].

The queries to the replica system may be submitted to any instance of Clarens that is part of a given VO.

The files (Datasets) that must be controlled by the replica system must be registered with it. Once this is done, all copy, delete and update operations will be made through the replica system, until the file is unregistered from it. The design of an optimal registration mechanism is still the subject of study, with some proposals suggesting automatic registration while in others the registration must be made explicitly.

4. Use of Monalisa as a monitor and support tool;

Given its characteristics of being a highly distributed environment and an already running in a major part of the HEP network computational resources, it is natural to choose Monalisa as the resource monitoring system. The fact that its development was made on the same group that developed Clarens also strengthens this choice.

5. Development of a system for file transfer in pieces (Torrents) on computational grids with interfaces for several persistence elements.

The project is to develop a generic Reliable File Transfer system (RFT) based on the Bit Torrent protocol with extensions that allow the authentication control on the Clarens context. This generic tool could then be used outside the Clarens/CMS context by other applications that need this transfer model.

The different persistency solutions mentioned earlier will be evaluated for storage of the actual files and collections.

6. Development of a metadata query system (view) of Datasets based on web interfaces.

Today Clarens has a partial metadata management system (MMS). Its data are stored on the RefDB [14], that is one of the persistence tools used by the CMS Grid. Some prototype methods to access metadata where implemented by Frank Van Lingen and have been tested in the CMS context [15].

The work proposes an extension of the above prototype, aggregating more information in the metadata system and to develop a better access mechanism. The metadata system must be coupled to the replica system [16], to make sure that a consistent view of the state of the system is maintained as far as possible when the copy, deletion and update operations happen.

5 Implementation

At this moment (February 2006) the implementation of the RFT and RLS are under way. First evaluation tests are scheduled to the month of April.

The Vision and Metadata systems are in the final stages of definition. Their implementation is expected to start on the second half of 2006.

6 Conclusion

The work proposes the definition and development of a completely distributed web service based system to maintain, locate and transfer Datasets in the context of Clarens and the CMS experiment. Besides the dissertation already in progress, at least one more is envisioned for the development of the file transfer tool. This work may also lead to further dissertations.

The system is totally distributed and based on web services and web interfaces.

References

1. I.Foster and C. Kesselman and S. Tuecke. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.
2. Large Hadron Collider. `http://http://lhc.web.cern.ch/lhc/`
3. Compact Muon Solenoid. `http://cmsinfo.cern.ch/Welcome.html`
4. C. Steenberg, E. Aslakson, J. Bunn, H. Newman, M. Thomas, F. Van Lingen. The Clarens Web service architecture. Computing in High Energy and Nuclear Physics (CHEP), La Jolla California, Mar. 2003
5. G. Couloris, J. Dollimore, T. Kindberg. Distributed Systems Concepts and Design. Addison Wesley, 2005.
6. I. Legrand, MonALISA - Monitoring Agents using a Large Integrated Service Architechture. International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Tsukuba, Japan, December 2003.
7. I. Foster, A. Iamnitchi, On Death, Taxes and the Convergence of Peer-to-Peer and Grid Computing. Peer-to-Peer Systems II: Second International Workshop, IPTPS 2003 Berkeley, CA, USA, February 21-22, 2003.
8. M. Cai, A. Chervenak, M. Frank, A Peer-to-Peer Replica Location Service Based on A Distributed Hash Table. Proceedings of the SC2004 Conference, 2004.

9. B. Cohen. Bittorrent protocol specification. http://www.bitconjurer.org/BitTorrent/protocol.html
10. P. Fuhrmann, dCache: the commodity cache, proceedings of the Twelfth NASA Goddard and Twenty First IEEE Conference on Mass Storage Systems and Technologies, Washington DC 2004.
11. A. Bassi, M. Beck, T. Moore, J.S. Plank, The Logistical Backbone: Scalable Infrastructure for Global Data Grids.
12. ERAD 2006, Escola Regional de Alto Desempenho. http://www.sbc.org.br/erad/2006/index.php
13. D. Knuth, The Art of Computer Programming, Volume 3: Sorting and Searching, Third Edition. Addison-Wesley, 1997.
14. V. Leféubre, J. Andreeva. RefDB: The Reference Database for CMS Monte Carlo Production. Computing in High Energy and Nuclear Physics (CHEP), La Jolla California, Mar. 2003.
15. http://newman.dnsalias.net/refdb/
16. Allcock, B., et. al., Data Management and Transfer in High Performance Computational Grid Environments. Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771.