JavaFIRE: A Replica and File System for Grids

Marko Petek¹; Diego da Silva Gomes¹; Claudio Fernando Resin Geyer²; Alberto Santoro¹ and Stephen Gowdy³
¹ UERJ-Rio de Janeiro;² UFRGS-Porto Alegre; ³ CERN-Geneva

E-mail: mpetek@cern.ch, diego@cern.ch, geyer@inf.ufrgs.br, alberto.santoro@cern.ch, gowdy@cern.ch

Abstract. The work is focused on the creation of a replica and file transfers system for Computational Grids inspired on the needs of the High Energy Physics (HEP). Due to the high volume of data created by the HEP experiments, an efficient file and dataset replica system may play an important role on 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 basically immutable. This eases the task of the replica system, because given sufficient local storage resources any dataset just 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-to-peer networks (p2p). P2p networks are an important and evolving mechanism that eases the use of distributed computing and storage resources by end users. One common technique to achieve faster file download 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 when the network conditions are better suited to the transfer. The main tasks achieved by the system are: the creation of replicas, the development of a system for replicas transfer (RFT) and for replicas location (RLS) with a different architecture that the one provided by Globus and the development of a system for file transfer in pieces on computational grids with interfaces for several storage elements. The RLS uses a p2p overlay based on the Kademlia algorithm.

1. Introduction
This work describes JavaFIRE, a File and Replicas System for computational grids based on peer-to-peer (p2p) technology for the storing of data and indexes. By delivering data to several different memory unities it may reduce the points of failure and the stress that exists today on the central repositories in T1s, T2s and T3s. Being p2p it is self-managed, therefore demanding much less manpower than the current solutions in use.

It also implements a dynamic web search interface that greatly reduces the time spent on searching datasets.

2. Model Proposed
The JavaFIRE model proposed in this work is made of three different systems:
- A Vision System: used by the researcher to search for datasets that match parameters previously defined. The Vision System may query a Metadata database for advanced searches;
- A Replica Management System: based on a peer-to-peer layer, looks for different replicas of a dataset in order to improve the throughput and reduce the data transfer times between source and destination;
- A Reliable Transfer System: ensures that the dataset is properly transferred for the researcher machine.

Figure 1 shows the main components of the JavaFIRE architecture.

![Diagram of JavaFIRE Model](Image)

**Figure 1: JavaFIRE Model**

3. **Model Proposed**

When a User wants to search and download a dataset for its own computer, the sequence of activities is the following:

1. A query is sent by the User to be answered by 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 received from the Metadata System, the Vision System returns to the User information about the datasets that fulfill 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 Peer-to-peer 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 Peer-to-peer Network finds the Storage Element(s) where the desired replicas are stored;
8. The Storage Element(s) returns information about the physical location of the dataset file descriptors and others that are need for the transfer to the peer-to-peer Network;
9. The Peer-to-peer Network sends its information to Replica System;
10. The Replica System returns the information to User;
11. The User asks the Reliable Transfer System for the wanted dataset;
12. The Reliable Transfer System sends one or more solicitations to Storage Element(s) for transfer of the dataset;
13. Storage Elements transfer the dataset to Reliable Transfer System;
14. The Reliable Transfer System delivers the dataset to User.
4. Main Features

4.1.1. Web Services Communication
All communication between the different modules of the system is done through Webservices (REST), thus allowing for a high degree of intercommunication due to the use of open standards.

4.1.2. Optimized Transfer Times
Several techniques are applied in order to minimize transfer times, among them:

- Pre-compression of data when worth (algorithm evaluates)
- Transfer several chunks of data from many sources in parallel
- Recovery of lost transfers
- High performance source selection and replacement algorithm
- Different chunks of a file may be stored on different Storage Elements
- No need to have the whole file in the same Storage Element to use the element

4.1.2. Integration with current Grid Middlewares
From the moment of conception, the idea was not to create a stand-alone model, but rather one that would be able to integrate with several Grid middlewares currently being used.

As the model works using Webservices it is quite easy to integrate it, regardless of the Middleware being used.

The system is being tested with three different middlewares. The first of all is Globus, the most widely used of them. Then comes Clarens [2], developed at Caltech specifically for the HEP area and finally with Exehda, a middleware developed in Federal University of Rio Grande do Sul (UFRGS). As Exehda uses Java serialization rather than Web Services, the model also provides APIs using this technology.

JavaFIRE relies on Globus Security Infrastructure (GSI) to handle the security aspects.

4.1.3. Peer-to-peer Layer
The Replica Management Service is based on a peer-to-peer network [4]. Its implementation is ready and is called JavaRMS [5]. It uses Distributed Hash Tables for the lookup service.

The selected p2p algorithm to be used was Kademlia [6].

It makes the location of the different pieces of a Dataset transparent for the end-user and at the same time handles all the distribution of data on the different peers that compose the CMS Computing Grid. For the disk and tape systems those pieces are normal files, so no changes are needed.

This makes the system highly autonomous of human intervention as p2p systems use to be [7].

5. Conclusions
Since its early days, grids and p2p networks are seen as technologies that will converge sooner or later [7].

Grids usually demand much more human intervention to work, as can be seen during the operation of the HEP grids of today.

Meanwhile p2p is self-managed, more fault-tolerant and elastic. The overall stress on the network is highly reduced.

One of the goals of JavaFIRE is to increase the automation level of grids involved in Big Data Science bringing it closer to the level of p2p systems.

JavaFIRE is also a perfect match for researchers wanting to use the CernVM environment to run analysis software on their desktops or laptops because it allows them to download large datasets from different sources at once, reducing the waiting time for the data arrival.
References
[1] I.Foster and C. Kesselman and S. Tuecke. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.
[2] 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.
[3] G. Coulouris, J. Dollimore, T. Kindberg. Distributed Systems Concepts and Design. Addison Wesley 2005.
[4] 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
[5] D.S.Gomes, JavaRMS: Um Sistema de Gerência de Dados para Grades Baseado num Modelo Para-Par. Master Dissertation, UFRGS 2008.
[6] P. Maymounkov, D. Mazières, Kademia: A Peer-to-peer Information System Based on the XOR Metric. Pdf available in: http://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf . Accessed on May, 2012
[7] I.Foster and A.Iamnitchi, On death, taxes, and the convergence of peer-to-peer and grid computing. Peer-to-Peer Systems II, 2003 - Springer