JavaFIRE: a Replica and File System For Grids

Marko Petek, Diego da Silva Gomes, Cláudio F. R. Geyer, Alberto Santoro, Stephen Gowdy

{petek, dsgomes, geyer}@inf.ufrgs.br, santoro@uerj.br, gowdy@cern.ch

Introduction

This works describes JavaFIRE, a File and Replication System for computational grids based on peer-to-peer (p2p) technology for the storing of data and indexes.

By delivering data to individual storage elements 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.

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;

- A Reliable Transfer System: ensures that the dataset is properly transferred for the researcher machine.

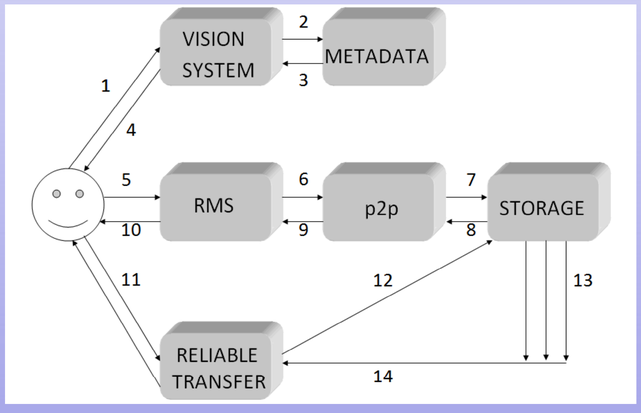

Figure 1- JavaFIRE Model

Operational Algorithm

When a User wants to search and download a dataset for its own computer, 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 information 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 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) return information about the physical location of the dataset and others that are need for the transfer to the peer-to-peer Network;

9- The peer-to-peer Network sends its information to the Replica System;

10- The Replica System returns the information 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 Storage Element(s) for the transfer of the dataset;

13- Storage Elements transfer the dataset to the Reliable Transfer System;

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

Main Features

Web Services Communication

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

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 chucks of data from many sources in parallel
- Recovery of lost transfers
- High performance sources selection and replacement algorithm
- Different pieces of a file may be stored on different storage elements
- No need to have the whole file in a storage element to use it

Integration with current Middleware

From the moment of its conception, the idea was not to create a stand-alone model, but rather one that would be able to integrate with many different Grid Middleware.

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

The system is being tested with 3 different Middleware. First of all there is Globus, the most widely used of all. Then comes Clarens, developed in 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.

Peer-to-peer Layer

The Replica Management Service, is based on a peer-to-peer network. Its implementation is ready and is called JavaRMS. It uses Distributed Hash Tables for the lookup service.

It makes the location of the different pieces of a Dataset transparent for the end-user and at the same time it handles all the distribution of data on the different peers that compose the CMS Computing Grid.

This makes the system highly autonomous of human intervention.

Conclusion

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

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

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

JavaFIRE's goal is to transfer at least part of the p2p approach for grids involved in Big Data Science.

JavaFIRE is also a perfect match for researchers wanting to use the CernVM environment to run analysis software on their laptops.

Bibliography

- I.Foster and C. Kesselman and S. Tuecke. The Grid: Blueprint for a New Computing Infrastructure. *Morgan Kaufmann, 1999.*
- 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.
- G. Couloris, J. Dollimore, T. Kindberg. Distributed Systems Concepts and Design. Addison Wesley 2005.
- 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
- D.S.Gomes, JavaRMS: Um Sistema de Gerência de Dados para Grades Baseado num Modelo Par-a-Par. Master Dissertation, UFRGS 2008.