Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 412–421

Procedia Computer Science

International Conference on Computational Science, ICCS 2011

A method for reliably managing files with RNS in multi Data Grids

Yutaka Kawai[a], Adil Hasan[b], Go Iwai[a], Takashi Sasaki[a], Yoshiyuki Watase[a]

[a] *Computing Research Center, High Energy Accelerator Research Organization (KEK), Tsukuba, Ibaraki Japan*
[b] *School of English, University of Liverpool, Liverpool UK*

Abstract

This paper describes a method for reliably managing files distributed in different kinds of Data Grids with RNS (Resource Namespace Service). RNS provides hierarchical namespace management for name-to-resource mapping as a key technology when using Grid resources for different kinds of middleware. We define attribute expressions in XML for the RNS entries and give algorithms to access distributed files stored within different kinds of Data Grids.

The volume of digital data and the size of an individual file are increasing due to the introduction of high-resolution images, high-definition audiovisual files, etc. The reliable storage of such large files is becoming problematic with whole file replication as a failure in the integrity of the file is difficult to localise. Our method involves managing large files in Data Grids by splitting them into smaller units in a traceable manner and then managing the smaller units. The RNS catalog service contains EPR (Endpoint Reference) and metadata that describe the original locations as well as the checksum values. The example in this paper shows how our Grid application can retrieve the actual file locations and the checksum values from the RNS service via SAGA (A Simple API for Grid Applications). An application can access the distributed files as though they were files in the local file-system without worrying about the underlying Data Grids.

This approach can be used with various Data Grid systems to enhance file reliability.

Keywords: RNS, Data Grid, SAGA, checksum

1. Introduction

The volume of digital data and the sizes of the individual files are increasing due to the introduction of high-resolution images, high-definition audiovisual files, etc. The reliable storage of such large files is becoming problematic and replication failures anywhere in such a file are difficult to localize.

In this note we describe a method of managing large files in different kinds of Data Grids by splitting them into smaller units in a traceable manner and managing the smaller units with the RNS (Resource Namespace Service)[1]. RNS catalog server contains the metadata that describes the original locations of the divided pieces and their MD5 checksum values. We also describe the tools developed to demonstrate the method that allows a file to be split before ingestion into Data Grids and assembled after extraction from the Data Grids. We can store metadata information in RNS and RNS allows the large file to be discovered in the Grid systems.

Email addresses: yutaka.kawai@kek.jp (Yutaka Kawai), adilhasan2@gmail.com (Adil Hasan), go.iwai@kek.jp (Go Iwai), takashi.sasaki@kek.jp (Takashi Sasaki), yoshiyuki.watase@kek.jp (Yoshiyuki Watase)

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.043

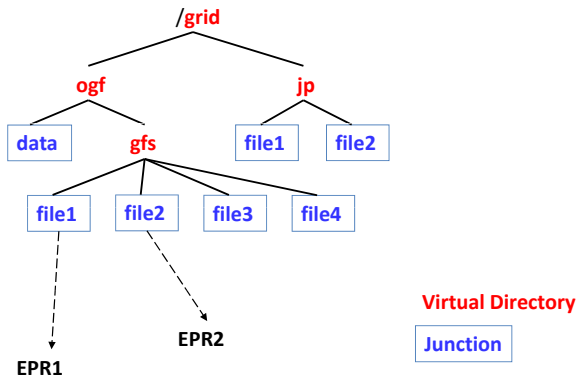

Figure 1: RNS: Hierarchical namespace management.

This paper describes the use of metadata in RNS to manage files distributed in different kinds of Data Grids. We use two kinds of Data Grids: iRODS (The Integrated Rule-Oriented Data System)[2, 3] and Gfarm (Grid Data Farm)[4]. The example of this paper involves sharing large files using an application based on SAGA (A Simple API for Grid Applications)[5, 6] to span the environment with iRODS and Gfarm. We already showed how to manage distributed files with RNS in such heterogeneous Data Grids[7]. That paper presented an easier way to share the metadata about each file between the sites of different research organizations for ongoing use. This paper uses the information for the physical file locations and the MD5 checksum values to enhance the reliability of the files.

This paper gives an overview of RNS in Section 2. Section 3 describes how to access distributed files by using RNS. An overview of the SAGA is presented in Section 4. In Section 5 we describe the current reliability to manage a large file and in Section 6 we describe how to enhance the reliability. Section 7 summarizes some of the tests we used to determine the effectiveness of our approach and we then conclude with a summary.

2. RNS Overview

RNS, which was introduced in GFD101, offers a simple standard way of mapping names to endpoints within a Grid or distributed network[1]. As shown in Figure 1[8, 9], RNS provides hierarchical namespace management with name-to-resource mapping[8]. RNS has two fundamental types of entries: virtual directories and junctions. RNS virtual directories represent non-leaf nodes in a hierarchical RNS namespace tree. RNS junctions link references to the existing resources into the hierarchical RNS namespace. All compliant RNS implementations must have a valid WSAddressing[10] EPR (Endpoint Reference).

The prototype implementation of RNS is available from Osaka University[9] and the University of Tsukuba[11]. We use the API of the RNS prototype in our implementation. The RNS servers and clients of the application are communicating with XML messages using SOAP (Simple Object Access Protocol)[12] as defined in GFD101, and also each RNS directory or junction entry can contain its own XML messages as metadata.

3. Access to Distributed Files With RNS

As our example of distributed file access, we assume that the large file data*A* consists of several file pieces stored in different kinds of Data Grids. To manipulate data*A*, application users need to retrieve all of the distributed pieces by using a Grid application.

A Grid application needs the physical locations of the existing resources in the Data Grids. The required additional information can be attached to each RNS entry because RNS entries can include XML metadata. We have defined XML attribute expressions for each RNS entry to handle all of the physical locations. An RNS virtual directory contains RNS junctions that have the physical location and the checksum value for each piece of the file. The EPR