Replica Update Propagation for Grid Data Resources

Marek Ciglan, Ladislav Hluchy
Institute of informatics, Slovak academy of sciences
Email: {marek.ciglan, hluchy.ui} @savba.sk

Abstract

Data replication is a well established concept in the grid community to increase data availability and fault-tolerance in the grid environment. However, there is a lack of software tools for maintaining the consistency among distributed data resources suitable for the heterogeneous nature of the grid. In this paper, we try to address this gap by introducing service for data update propagation among distributed data replicas. Our replica update service provides basic functionality for higher level grid consistency services and virtualizes the underlying data resources. We discuss the possibility of integration of replica update service with legacy software, using legacy software functionality and, increasing the data availability by means of data replication.

1. Introduction

Resource sharing nature of grid technology encourages and provides means to enhance the cooperation in the scientific communities. The concept of the computing grid arose from the need to share computing power, mostly for the jobs that use read only data sets as input (data output from scientific experiments). Thus, the data management tools for grid computing was primary designed to manage read-only data sets. Data replication concept was adopted as a way to increase the data availability and provide fault-tolerance. The grid technology is gaining popularity among wider scientific communities and the importance of updatable data resources becomes evident. Tools that would allow data replication and updating of those data could increase the cooperativeness and ease the data sharing within the community. Services for grid data consistency handling are needed to achieve this goal. Much work on the data consistency was already done within relational databases community. However, the grid environment posses additional challenges; such as the heterogeneity of data resources, the need for virtualization and the necessity of relaxed consistency models.

In this paper, we present a service for replica update propagation, which can serve as the base block for higher-level consistency services.

2. Replica update propagation service

OGSA-DAI [2] is currently most advanced tool for data access and integration for grid data resources, compliant with emerging GGF standards [3]. This makes this middleware environment the best choice for building higher level data-related grid services. OGSA-DAI constitutes a middleware layer over diverse types of data resources – relational databases, xml databases and file resources. It provides WSRF [1] standard compliant web service interface for accessing the system functionalities. Our Replica Update proPAGATION service (RUPAGATION) use OGSA-DAI framework as the operational environment. It is build as a set of modules, which can be plugged in OGSA-DAI data services.

The aim of RUPAGATION is to form the base stone for higher level grid consistency services, which will provide automatic and autonomous synchronization of replicated data resources in the distributed, heterogeneous grid environment. RUPAGATION provides basic functionality for consistency services. It is able to catch and cache updates submitted to specified data resources, propagate updates to the other replica sites and apply updates on data replicas.

RUPAGATION can provide its functionality for all types of resources supported by OGSA-DAI. This means that if we have an implementation of a consistency model that is build on top of RUPAGATION service, we can use the same implementation to ensure consistency for a large number of relational databases, xml databases and file resources. This virtualization provided for consistency services is, in our opinion, the most important aspect of this work.

RUPAGATION was decoupled to several modules with clearly defined functionalities which can be deployed separately, according to the system requirements. From the technical point of view, modules are set of OGSADAII activities.
RUPAGATION is composed of following modules: Update catcher, update manager, update transfer and update applier module.

Update catcher module is the first interaction point for incoming update statements. It is responsible for caching those commands for later processing by the update manager module. After receiving each update statement, module sends a notification message to update manager, which, consequently, can trigger an update operation.

Update manager component is responsible for creating new update set from newly arrived updates. It is also responsible for maintaining previously created update sets and for collecting information about replicas consistency states. Update manager defines an interface for interaction with consistency services, implementations of consistency models.

Update transfer module is a chain of data delivery activities (mostly provided by OGSA-DAI middleware) that is able to transfer data files (update sets) to remote grid node operating RUPAGATION service. The transferred updates are registered within appropriate Update Applier module.

Update applier module is module responsible for maintaining information about state of a data resource replica (e.g. which updates was already applied on the replica, timestamp information for each update etc.) and for executing updates on replica stored in underlying data resource.

3. RUPAGATION usage

On the top of RUPAGATION, we have built simple consistency service for primary copy update model. One replica of a data source is labeled to as a primary copy. For each non-primary replica, only read operations are allowed for users or applications. Update operations are applied only on primary replica, changes performed on the primary replica are then propagated and applied on the other replicas. Update manager module notifies the simple consistency service of each incoming update. This naïve consistency service provides us with the behavior similar to Read-One-Write-All (ROWA) consistency model [4].

To test the implementation, we have integrated RUPAGATION (using described simple consistency model) with a legacy application - Metadata Catalog Service (MCS) [5]. The calls for JDBC update statements in MCS code was replaced by invocation of update operation on (remote) primary replica data resource using web service interface of OGSA-DAI data service. Each deployed copy of MCS performed read operations on local database, the update operations was send to primary replica copy. Update propagation service than applied those updates to all deployed MCS's data replicas.

The time efficiency of update operations decreased, but the load of read-only operation was distributed among the deployed copies of MCS.

Two notable observations from this experiment are:
• the solution is suitable for applications which perform large amount of read-only requests and only relatively few write operations
• we can turn (possibly non-collaborative) legacy application into an distributed application with consistent replicated data among deployed instances.

4. Future work

Presented software is (in the time of writing the paper) still in prototype stage. In current implementation, the update catcher module is available only for relational data resource. Support for XML databases and file resources is planned in near future. We also plan to implement several consistency models on the top of RUAGATION system.

5. Conclusions

In this paper, we have presented a set of software modules that form an update propagation service for relational data resources in the grid environment. Described software modules are implemented as a collection of activities pluggable into OGSA-DAI framework. Proposed system is intended to form a base for higher-level services for grid data consistency handling.

We have also described a possibility to integrate proposed update propagation system with legacy applications and we have highlighted possible benefits of this approach.

6. References