The AMGA WS-DAIR Implementation

Sunil Ahn[1], Taesang Huh[1], Soonwook Hwang[1]

[1] KISTI, 245 Daehak-ro, Yuseong-gu, Daejeon, Korea
{siahn, tshuh, hwang}@kisti.re.kr

Abstract. AMGA is one of first two reference implementations with which interoperability was tested for WS-DAIR standard ratification. This paper presents a reference implementation model of the WS-DAIR interface based on AMGA. AMGA is designed for a metadata catalog service, which allows the implementation of the WS-DAIR interface much simpler and more efficient in terms of performance and scalability.

Keywords: AMGA, WS-DAI, Grid, metadata catalog, Grid data access, interoperability

1 Introduction

OGF (Open Grid Forum) DAIS (Data Access and Integration Services) working group [1] has developed the WS-DAI (Web Service Data Access and Integration) family of specifications [2], which defines web service interfaces to data resources. The WS-DAI [3] is a data independent specification, which is extended to several data dependent specifications, such as WS-DAIR (Data Access and Integration – Relational Realization) [4] and WS-DAIX (Data Access and Integration – XML Realization) [5].

AMGA (ARDA Metadata Grid Application) [6-7] is an official metadata service for EGEE [8], being part of the gLite [9] middleware software stack being developed for the EGEE project. Metadata service is considered to be one of integral parts of Data Grids where it is common to see millions of files spread all over the geographically distributed environments because it provides an efficient mechanism to publish data with descriptive information (i.e. metadata) about files and to locate the files using the metadata as well.

The WS-DAIR interface has been implemented for AMGA, which allows a seamless integration of AMGA into the DAIS framework of OGF standardized Grid data access services. The WS-DAIR interface for AMGA will greatly improve the extensibility and interoperability with other Data access services. It allows users to build interoperable applications in a programming language of their choice by using standard interfaces to access data resources. In addition, data resource services that are newly introduced to the Grid will be readily accessible with existing clients.

This paper presents a reference implementation model of the WS-DAIR interface based on AMGA. OGSA-DAI and AMGA are first two reference implementations which had tested their interoperability for WS-DAIR standard ratification [10], but their designs and implementations have several differences. While OGSA-DAI is designed for general purpose database accessing on Grid, AMGA is basically designed for a metadata catalog service or very simple database accessing on Grid. It is not necessary to provide all the capabilities of general purpose databases for the AMGA service, which allows the implementation of the WS-DAIR interface for AMGA much simpler and more efficient in terms of performance and scalability.

2 Design and Implementation

The implementation provides three data services defined in WS-DAIR; SQLAccess, SQLResponse and SQLRowset. The SQLAccess data service handles direct data access, in which a result of a query is returned directly. A SQLExecuteFactory operation creates a new data resource from a query, and returns its reference. The SQLResponse data service deals with indirect data access, in which newly created resources are accessed by the returned reference. The SQLRowset data service also serves indirect data access, but it manages one dataset and allows a resource accessed per tuple basis while SQLResponse manages multiple datasets and allow a resource accessed per dataset basis. All these three data services are actually served by a single standalone daemon to avoid unnecessary communications among services.

WS-DAIR specifies that the SQLRowSetFactory operation creates a new data resource through a SQLResponse data service again and the SQLRowset data service han-dles indirect data access on a data resource derived from a SQLResponse data service. However in case of the AMGA WS-DAIR implementation there is no difference between a resource managed by SQLResponse and a resource managed by SQLRowset, so that it is not necessary to create a new data resource on the SQLRowSetFactory operation call. That is because metadata searching in AMGA does not create multiple datasets in any cases, while general database query may create multiple datasets in some cases such as calling a stored procedure. Therefore SQLResponse and SQLRowset data services share the same data resource.

Calling the SQLExecuteFactory operation causes an internal database view created and the created view is mapped to a directory structure of AMGA. This mapping allows a user to access newly created data resource through the previously existing AMGA interface.

This simplified implementation model on data services and data resources helps the implementation to minimize overhead caused by the WS-DAIR interface. In addition, it allows a user to have a consistent view on data in both interfaces; the WS-DAIR interface and the existing AMGA interface.

The figure 1 shows major components of the AMGA WS-DAIR implementation. The connection handler module plays a role of handling concurrent access from multiple users. The user's request is inserted into the queue and multiple threads retrieve requests turn by turn and process them. The authentication module plays a role of authenticate users based on their certificates, VOMS (Virtual Organization

Management System) [11] information or ID/password. The authorization module examines whether a user has proper permissions on the accessed tables based on user ID, group ID, permissions and ACL (Access Control List).

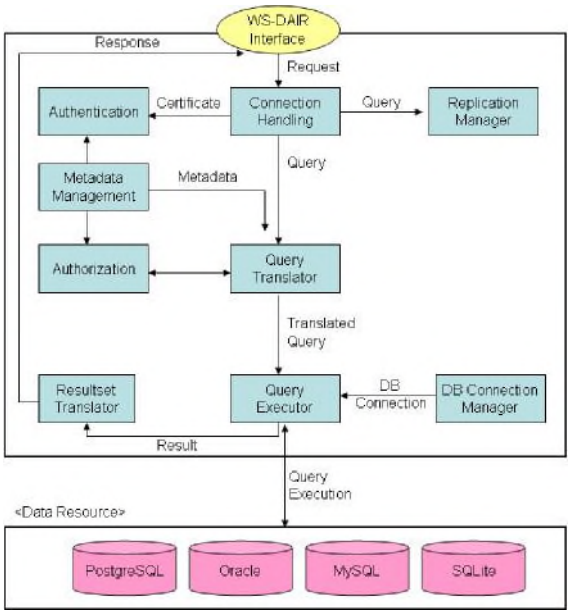

Fig. 1. Components of the AMGA WS-DAIR implementation

The query translator module parses user's query and the translated query is executed by the query executor module. And the result-set translator module returns the queried results after changing to the form that a user requests. The query translator plays a role of solving database heterogeneity. It translates a user's query to the form which the underlying DMBS understands.

The database connection module manages database connections which are shared by multiple threads. The replication module manages queries requiring changes on database and it transfers them to other slave nodes and let them executed as they are. The replication feature makes data replicated in many places to enhance performance as well as availability of data resource on the Grid. The metadata manager plays a role of managing metadata on users' metadata, such as access permissions per table, list of users, list of groups, directory to table mapping and so on.

To implement a web service, we used a C++ gSOAP toolkit which produces skeletons and stubs automatically. The WSDL files given at the WS-DAIR specification were used almost without modification and used binding was document/literal.

3. Conclusion

Interoperability of data accessing is one of the most important issues in many Grid projects, and the WS-DAIR specification was defined especially for relational databases. In this paper we presented a reference implementation of the WS-DAIR specification based on AMGA. It provides very simplified WS-DAIR implementation model on data services and data resources, so that it is possible to minimize overhead caused by the WS-DAIR interface. It also allows a user to have a consistent view on data in both inter-faces; the WS-DAIR interface and the existing AMGA interface.

References

1. DAIS Working Group, "Wiki page - OGF Data Access and Integration Services," April 2003, [Online]. Available: http://forge.gridforum.org/sf/projects/dais-wg/.
2. M. Antonioletti and A. Krause, "The WS-DAI Family of Specifications for Web Ser-vice Data Access and Integration," ACM SIGMOD, vol. 35, no 1, pp. 48-55, 2006.
3. M. Antonioletti, et al., "Web Services Data Access and Integration (WS-DAI) Speci-fication Version 1.0," OGF GFD.74, 2006.
4. M. Antonioletti, et al., "Web Services Data Access and Integration–The Relational Realisation (WS-DAIR) Specification Version 1.0," OGF GFD.76, 2006.
5. M. Antonioletti, et al., "Web Services Data Access and Integration–The XML Reali-sation (WS-DAIX) Specification Version 1.0," OGF GFD.75, 2006.
6. AMGA, "AMGA: The gLite Grid Metadata Catalogue," Sep. 2005. [Online]. Available: http://amga.web.cern.ch/amga.
7. N. Santos and B. Koblitz, "Metadata Services on the Grid," Nuclear Instruments and Methods in Physics Research, vol. 559, no. 1, pp. 53-56, 2006.
8. F. Gagliardi, et al., "Building an Infra-structure for Scientific Grid Computing: Status and Goals of the EGEE Project," Philosophical Transactions: Math., Physical and Engineering Sciences, vol. 363. pp. 1729-1742, 2005.
9. gLite, "Lightweight Middleware for Grid Computing," Mar. 2009, [Online]. Available: http://glite.web.cern.ch/glite/.
10. S. Lynden, M. Antonioletti, M. Jackson and S. Ahn, "WS-DAI and WS-DAIR Im-plementations - Experimental Document," OGF GFD.160, 2009.
11. R. Alfieri, et al., "VOMS, an Authorization System for Virtual Organizations," LNCS, vol. 2970, pp. 33-40, 2004.