Experiences of Designing and Implementing Grid Database Services in the OGSA-DAI project

Mario Antonioletti[1], Neil Chue Hong[1], Ally Hume[1], Mike Jackson[1], Amy Krause[1], Jeremy Nowell[2], Charaka Palansuriya[1], Tom Sugden[1] and Martin Westhead[1]

[1]EPCC, University of Edinburgh, James Clerk Maxwell Building, Mayfield Road, Edinburgh EH9 3JZ, UK.

[2]UK Grid Support Centre/EPCC, James Clerk Maxwell Building, Mayfield Road, Edinburgh EH9 3JZ, UK.

Abstract

This paper describes the experiences of the OGSA-DAI team in designing and building a database access layer using the OGSI and the emerging DAIS GGF recommendations. This middleware is designed for enabling other UK e-Science projects that require database access and providing the basic primitives for higher-level services such as Distributed Query Processing. OGSA-DAI also intends to produce one of the required reference implementations of the DAIS specification once this becomes a proposed recommendation and, until then, scope out their ideas, provide feedback as well as directly contributing to the GGF working group. This paper enumerates the issues that have arisen in tracking the DAIS and OGSI specifications whilst developing a software distribution using the Grid services model; trying to serve the needs of the various target communities; and using the Globus Toolkit OGSI core distribution. The OGSA-DAI software distribution and more details are available from the project web site at http://www.ogsadai.org.uk/.

1 Introduction

The *OGSA - Data Access and Integration* (OGSA-DAI) project started in February 2002. The work undertaken was split into two main phases. Phase I concentrated on developing prototypes and designs that fed into the second phase. Phase II was dedicated to producing three main releases of the OGSA-DAI software. The first production release of the code – Release 3 – was made in July 2003 shortly after the *Globus Toolkit 3.0* (GT3) release. Code development was undertaken by two small EPCC and IBM UK teams, working at opposite ends of the UK, with design input coming from the other team members.

The goals for OGSA-DAI were:

- To serve the UK e-Science community by providing middleware allowing controlled exposure of data resources to an OGSI-compliant Grid.
- To support access to heterogeneous data resources (currently relational and XML databases) through common interfaces using the existing underlying query mechanisms.
- To provide the base services to allow higher-level data integration services to be constructed, e.g. Distributed Query Processing[1] (DQP) and Data Federation services.
- To provide input into the GGF standardisation of data access interfaces through the GGF *Database Access and Integration Services* (DAIS) WG and to provide a reference implementation of the resulting DAIS recommendation.

OGSA-DAI continues to develop, adding support for more *Database Management Systems* (DBMSs), providing more functionality, contributing to the DAIS GGF WG, as well as extending the existing framework by incorporating interfaces to file systems.

2 Why use Grid Services?

The OGSI GGF recommendation [1] introduced Grid services, built as an extension to Web services, offering

[1] See http://www.ogsadai.org.uk/dqp.

benefits for data access and integration services over non-OGSI Web Services solutions. These benefits include:

- **Service Data Elements**: allow the state of a service to be exposed and provide a standard interface for getting and setting service properties, for example whether a database query is in progress or complete. Higher-level services can then use these to query the underlying database state, e.g. a DQP service could find out the database schema or the server load through the service's SDEs.
- **Dynamic Service Creation:** allows service instances to be created on demand according to need. Transient service instances are useful for managing aspects of an interaction, e.g. to provide context for a database connection, or to monitor a computing job.
- **Lifetime Management:** can be used to ensure that services that are no longer required are cleaned up (so resources are not wasted) as well as providing lifetimes for notification subscriptions, cache lifetimes and other time-constrained services and resources.
- **Service Groups:** facilitate the provision of registries that can be used by clients to identify service instances meeting application-specific requirements. This allows the assignment of unique global ids for data resources, location independence and ultimately true virtualisation of data.

3 OGSA-DAI Overview

OGSA-DAI extends OGSI by adding the following portTypes:

- `GDSPortType` – a derivation of:
  - `GridDataPerform` – provides the main end point for accessing a data resource.
  - `GridDataTransport` – used to push/pull data from/to a service.
- `GridDataServiceFactory` – extends the OGSI `Factory` portType to create OGSA-DAI services implementing the `GDSPortType`.
- `DAIServiceGroupRegistry` – facilitates management of a group of OGSA-DAI services.

These port types are, in turn, used to construct the base services, described in **Table 1**, that provide the core functionality of OGSA-DAI.

Service	Description
Grid Data Service (GDS)	Provides the access end point for a client and holds the client session with that data resource. A GDS is created by a GDSF.
Grid Data Service Factory (GDSF)	A GDSF is defined to represent the point of presence of a data resource on a Grid. It is through a GDSF that a data resource's capabilities and meta-data are exposed.
DAI Service Group Registry (DAISGR)	GDSFs may be located on the Grid through the use of a DAISGR with which GDSFs may register to expose their capabilities and any meta-data to aid service/data discovery.

Table 1: OGSA-DAI services

In the OGSA-DAI view of the world, DAISGRs and GDSFs are persistent services. They are instantiated out of band when the Grid service-hosting container is started. Each GDSF currently only exposes one database (either XML or relational). If a container has to expose more than one database, this must be done via separate GDSFs.

On creation a GDSF may register its capabilities and any meta-data with a DAISGR that clients can subsequently use

to discover the services that the GDSF offers. Currently OGSA-DAI does not constrain the type of meta-data that may be registered by a GDSF – OGSA-DAI is awaiting guidance to come from DAIS and/or other GGF WGs on these matters.

The standard OGSA-DAI scenario has a client contacting a DAISGR first to locate GDSFs that meet its requirements. The client then accesses suitable GDSFs directly to find out more about their properties and the data resources they represent. The client may then ask one of these GDSFs to instantiate a GDS to allow access to the data resource it represents. Accessing the data resources is achieved by sending the GDS a GDS-Perform document via the `GridDataPerform::perform` operation of the GDS. This document contains a collection of *activities*, analogous to those defined in the GGF7 DAIS specification [2], which are linked together within the same GDS-Perform document. The document is processed by a component of the GDS known as the *engine*. Data can thus be delivered to/from the client (or third party) according to the instructions contained in the perform document.

Activities provide an extensible way of adding functionality to GDSs. An XML Schema defines the syntax of activities and an external implementation, in Java, is associated with each activity. These can be dynamically loaded into the GDS engine as required, although for simplicity in the current implementation the activities that a GDS supports are loaded at creation time and are specified in a GDSF configuration file read at start-up time.

Some of the activities that OGSA-DAI has pre-defined are shown in **Table 2**.

Activity	Description
Relational Type Activities	
`sqlQueryStatement`	Perform an SQL Query statement.
`sqlUpdateStatement`	Perform an SQL Update statement.
XML Type Activities	
`XPathStatement`	Perform an XPath statement against an XML database.
`xUpdateStatement`	Perform an XUpdate statement against an XML database.
Delivery Activities	
`deliverTo/FromURL`	Deliver results to a user-specified URL.
`deliverTo/FromGDT`	Deliver results to another service that implements the `GridDataTransport` portType.
`deliverTo/FromGFTP`	Deliver results using GridFTP.
Transformation Activities	
`gzipCompression`	Compress results into a GZIP stream.
`xslTransform`	Perform an XSL transformation against an XML document.

Table 2: OGSA-DAI activities

Activities serve as the main extension point of OGSA-DAI. Developers can embed their own application-specific activities by providing their own XML schema and an implementation that can be used by the GDS engine. More information on writing activities can be found in [3].

3.1 Security

OGSA-DAI security is based on the message-level security model of GT3. It is straightforward to configure message-level security using GT3. Security restrictions can be applied to an OGSA-DAI service as a whole or on an operation-by-operation basis.

The security model of the underlying DBMS is used to enhance the security of OGSA-DAI. The configuration files of OGSA-DAI allow a *role map* to be specified. This role map allows specific Grid credentials to be mapped to corresponding database roles that are used to access the database. A GDS then accesses the database using this database role. It is therefore possible to grant data access permissions that are specific to each

individual OGSA-DAI user or to specific groups of OGSA-DAI users.

3.2 Transport

It is important for the OGSA-DAI transport model to ensure that no unnecessary data movement takes place between a GDS and any consumer. Data is streamed through the engine, although asynchronous and block delivery modes are also supported to allow marshalling of services for co-operative scheduling.

Third party delivery can be achieved using standard non-OGSA protocols such as GridFTP, HTTP and FTP as well as a simple implementation of a `GridDataTransport` activity that allows data to be pushed/pulled from a service directly.

4 Implementation

4.1 Tracking Specifications

Since the start of the OGSA-DAI project the OGSI specification has gone through 29 iterations, rapidly accelerating towards the final proposed recommendation status achieved at GGF 8. Similarly, the DAIS specification has changed radically for each of its GGF outings since GGF 6. Trying to provide a consistent framework with such rapidly moving targets has presented unique problems.

A series of design documents were used to track and anticipate changes in the specification documents as well as sheltering developers from the immediate impact of these changes. This also allowed the OGSA-DAI team to provide valuable input into the DAIS recommendation process.

4.2 Using the Globus Toolkit 3

Early development using technology previews, Alpha and Beta releases of GT3 proved challenging due to the prototypical nature of these products and limited coverage in the associated documentation.

Development using the current GT3, however, has proved to be far more straightforward and intuitive. This reflects the advanced state in which GT3 now exists and the improved support for service developers and users provided by the software and its associated documentation.

4.3 Tooling

The Eclipse IDE was used to develop the main OGSA-DAI codebase. This was chosen because of its ability to integrate well with JUnit for automated unit testing and ANT for automating the build process.

The use of these products – as well as generic tooling targeted at the Web services community – for Grid service development proved to be advantageous. For instance, XMLSpy and WebSphere were productively used during the project for XML Schema validation and development.

4.4 Platforms

The Apache Tomcat/Apache Axis/Globus GT3 services stack was used due to its acceptance as the *de facto* OGSI implementation. The software was tested on Microsoft Windows 2000/XP, Redhat Linux 8, IBM AIX and Sun Solaris 8 to give a representative spread of installed platforms.

MySQL, IBM DB2, Oracle and Apache Xindice were similarly chosen to give a representative selection of relational and XML databases in common usage. Although the software has only been tested against these databases, it has also been shown to operate with Microsoft Access and Microsoft SQL Server, eXist and IBM Cloudbase.

4.5 Testing

Testing of Grid services requires a comprehensive test plan that operates on a number of levels. For OGSA-DAI the test plan is as follows:

- At the lowest level, each software component contains a comprehensive set of unit tests that uses the JUnit framework. All unit tests are run and checked daily.

- Higher-level tests (client side/system tests) include tests for data operation primitives such as:
  - SQL/XPath/XUpdate statement execution,
  - Tests for their support tools (e.g. transformations, compression etc.),
  - Grid-level tests such as submission of GDS-Perform documents with various configurations (e.g documents with multiple activities, parameters, different delivery targets etc.) and use of the `GridDataTransport` portType.

The tests are contained in an automated system test framework (built using ANT and Java) and are run whenever a stable build is produced. Like the JUnit framework, the system test framework produces a report after each run listing the tests it ran and highlighting any failures.

There are added challenges in testing Grid service implementations, in particular when schema or source code changes make existing test code obsolete. Client-side libraries may help alleviate this problem, as well as helping uptake amongst developers and users alike.

4.6 Benchmarking and scaling

Performance of OGSA-DAI, and the Grid Services stack, is crucial to the acceptance of the software. The Information Systems Institute (ISI) at the University of Southern California provided benchmarking results from early releases of the OGSA-DAI software which were used to improve the performance of subsequent releases, in particular through a BlockReader/Writer framework which allows activities to stream data from one to another.

Initial runs show that OGSA-DAI and the Grid Services stack introduce a fixed but significant overhead which scales as JDBC with increasing result size until memory limitations begin to cause exceptions in the Tomcat hosting environment.

Further benchmarking activities with Release 3.0 are currently taking place at the Computing Science Department of Indiana University, the Computing Science department of the University of Manchester, and internally at EPCC. It is hoped to publish these results and the benchmarking suites when work is complete.

4.7 Interoperability

Implementation of a GDS client in C# that runs on the Microsoft .NET framework has proved straightforward. This client interacts with OGSA-DAI with no compatibility issues. A simple OGSA-DAI GDS demonstrator available for a Microsoft ASP.NET-based implementation of OGSI has also been developed. These activities were part of a related project, MS.NETGrid[2].

4.8 Teamworking

Many Grid projects are based around diverse, geographically distributed teams encompassing different organisations. There are particular challenges to be met in such projects but the OGSA-DAI project has found various technologies (CVS, IRC, Twiki, AccessGrid, NetMeeting, tele-conferencing) to be helpful in bringing the distributed team members together in a collaborative working environment.

5 Current Status of Project

As the project ends its second phase, it has built up a wide-ranging set of users who have been evaluating the software.

The follow-on DAIT project will continue development of the software for the next two years.

5.1 Downloads

At the time of writing, there have been over 900 downloads of the OGSA-DAI distribution, with over 200 downloads of the latest version, from all over the world.

[2] See http://www.epcc.ed.ac.uk/ogsanet/.

5.2 Training and documentation

OGSA-DAI courses have been given at the e-Science Institute in Edinburgh, GGF and the 2003 Grid Summer School. These courses have not only served to disseminate information about OGSA-DAI but have also presented an opportunity to canvass users about the functionality and direction of OGSA-DAI.

Extensive documentation is provided with the distribution and via the project web-site as well as the archives of the OGSA-DAI users mailing list. This is an important way to maintain and extend the OGSA-DAI user community and information is available at a variety of levels from installation guides to programming tutorials.

5.3 Support

Support for OGSA-DAI is provided by the UK Grid Support Centre[3] via the OGSA-DAI website. Queries have been successfully answered from users around the world. These questions range from general installation and configuration issues to complex queries regarding the client side access to OGSA-DAI.

5.4 Future Plans

The OGSA-DAI project is entering a third phase of development, which seeks to turn evaluators into implementers.

As OGSA Grid standards and technologies mature (e.g. the new GridFTP, CAS), the project will incorporate these into OGSA-DAI. OGSA-DAI currently is being extended to extend the framework for message-level security, which users can customise to enforce specific security mechanisms.

Added functionality and an emphasis on improved performance and scalability will be undertaken at the same time as a shift in emphasis to higher-level data integration services built on the basic data access building blocks.

Interoperability with other implementations, e.g. from the MS.NETGrid and edikt::eldas[4] projects will help to drive the acceptance of the DAIS recommendations and also grow the OGSA-DAI user community.

6 Conclusions

The OGSA-DAI project is an example of a large project spanning both industry and academia. The project focuses on developing Grid services in the rapidly evolving area of Grid computing. The difficulties of such a project can be significant, but these can be overcome because of the existence of clear draft standards, useful tooling and the experience of the existing Web services community.

The next stage of development will be even more interesting as the OGSI, DAIS and other standards stabilise and become accepted.

Acknowledgements: This work is supported by the UK e-Science Grid Core Programme, whose support we are pleased to acknowledge.

References

[1] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Snelling, D., Vanderbilt, P. *Open Grid Services Infrastructure version 1.0 (Draft 33)*. Open Grid Services Infrastructure WG, Global Grid Forum, June 27th 2003. See http://www.gridforum.org/ogsi-wg/.

[2] Chue Hong, N., Krause, A., Malaika, S., McCance, G., Laws, S., Magowan, J., Paton, N.W., Riccardi. G. *Grid Database Service Specification* – 16th February 2003.

[3] Hicken, G. *How to Write an Activity,* (OGSA-DAI-USER-UG-ACTIVITY-v1.0), July 2003. Available from http://www.ogsadai.org.uk/docs/current/OGSA-DAI-USER-UG-ACTIVITY.pdf.

[3] See http://www.grid-support.ac.uk.

[4] See http://www.edikt.org/eldas/.