The Design and Implementation of Grid Database Services in OGSA-DAI

Ali Anjomshoaa[1,2], Mario Antonioletti[1,2], Malcolm Atkinson[2], Rob Baxter[1,2], Andrew Borley[3], Neil P Chue Hong[1,2], Brian Collins[3], Neil Hardman[3], George Hicken[3], Ally Hume[1,2], Alan Knox[3], Mike Jackson[1,2], Amrey Krause[1,2], Simon Laws[3], James Magowan[3], Charaka Palansuriya[1,2], Norman W Paton[4], Dave Pearson[5], Tom Sugden[1,2], Paul Watson[6] and Martin Westhead[1,2]

This paper presents a high-level overview of the design and implementation of the core components of the OGSA-DAI project. It describes the design decisions made, the project's interaction with the Data Access and Integration Working Group of the Global Grid Forum and provides an overview of implementation characteristics. Further details of the implementation are provided in the extensive documentation available from the project web site.

[1] EPCC, University of Edinburgh, James Clerk Maxwell Building, Mayfield Road, Edinburgh EH9 3JZ.
[2] National e-Science Centre, 15 South College Street, Edinburgh EH8 9AA.
[3] IBM United Kingdom Ltd, Hursley Park, Winchester S021 2JN.
[4] Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL.
[5] Oracle UK Ltd, Thames Valley Park, Reading RG6 1RA.
[6] School of Computing Science, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU.

Introduction

The *Open Grid Services Architecture – Data Access and Integration* (OGSA-DAI) project is constructing an efficient Grid-enabled middleware implementation of interfaces and services to access and control data sources and sinks. For the phase of the project just completed, these data sources/sinks were restricted to be relational and XML database management systems (DBMS). The framework, however, has been designed to allow other data sources such as file systems to be accessed through the same interfaces.

The OGSA-DAI-defined services and interfaces, which extend those defined in the *Open Grid Services Infrastructure* (OGSI) specification [1], wrap individual physical data resources so they may be used by higher-level services to provide greater transparency. These data resources may then be easily incorporated within an *Open Grid Services Architecture* (OGSA) framework. The OGSA-DAI services provide the basic primitives to construct sophisticated higher-level services that allow data federation and distributed queries to take place within a *Virtual Organization* (VO)[1].

OGSA-DAI is closely affiliated with the Global Grid Forum (GGF) *Database Access and Integration Services* (DAIS) Working Group (WG)[2]. Input from OGSA-DAI developments has gone to this WG and, in turn, ideas from DAIS have been incorporated into OGSA-DAI. One of the aims of this project is that OGSA-DAI will produce one of the reference implementations of DAIS. This imposes external requirements. Future changes must ensure that OGSA-DAI is properly aligned with the GGF DAIS WG recommendation requirements – once these have stabilized.

In addition to the DAIS coupling, OGSA-DAI also has a requirement to provide UK e-Science projects with a data access and integration middleware distribution to act as a base for their own higher-level services. This also produces requirements that OGSA-DAI can incorporate and feedback to the DAIS WG.

The UK Department of Trade and Industry (DTI) have funded this project under the e-Science Grid Core Programme.

[1] See, for instance, the Distributed Query Processing capabilities established using OGSA-DAI, for more details see the project web site.

[2] http://www.gridforum.org/6_DATA/dais.htm.

More details may be obtained from the project web site:

http://www.ogsadai.org.uk/

Phases and Releases

The OGSA-DAI project was been split into two main phases. Phase I resulted in the production of prototype software and initial design and requirements documents for Phase II. Phase II, which this document describes, was concerned with the design, implementation and documentation of production level software.

OGSA-DAI software releases were staged over the project lifetime to gradually increase the OGSA-DAI functionality and to facilitate the tracking of the rapidly-evolving OGSI and DAIS GGF documents. The release schedule was as follows:

- **Phase II – Release 1 – January 2003**

  The first release covered basic functionality. The intention was to ensure that the base infrastructure was in place, and was demonstrable, in order that more complex functionality could be added in later releases. The design of release 1 was based on draft 5 of the *Open Grid Service Infrastructure* [2].

- **Phase II – Release 2 – April 2003**

  This intermediate release added more functionality (primarily delivery options) to that defined for release 1 and was compatible with the Globus Toolkit 3 (GT3) Alpha 3 release.

  An interim OGSA-DAI release, version 2.5 was made available towards the end of May 2003. This was compatible with the GT3-Alpha 4 release, which was consistent with [1].

- **Phase II – Release 3 – July 2003**

  This final Phase II release included most of the functionality scoped out in Phase I of the project. This release was compatible with the first production release of GT3.

Over 850 copies of the OGSA-DAI releases have been downloaded from the project web site over this time frame. The future evolution of this project is described shortly.

System Overview

The *Open Grid Services Architecture* (OGSA) presents a picture of the Grid where Grid resources and services are represented by instances of Grid services. Grid services, as defined in [3], are stateful service instances supporting reliable and secure invocation, lifetime management, notification, policy management, credential management, and virtualization.

The OGSA-DAI project is developing Grid services that represent data resources, where, by a data resource, we mean any physical or logical entity that is able to source or sink data. These underlying data sources and sinks, together with any associated management infrastructure, are referred to as *physical data resources*. The term *data resource* is then used to represent the aspects and capabilities that are exposed to the Grid. Although this framework potentially has a wider applicability, OGSA-DAI currently only provides interfaces to relational and XML database management systems.

The prime goals of OGSA-DAI were:

- To provide controlled exposure of physical data resources to the Grid.
- To support access to heterogeneous physical data resources through a common interface style while employing the underlying query mechanisms.
- To provide base services that allow higher-level data integration services to be constructed, e.g. distributed query processing and data federation services.
- To leverage emerging Grid infrastructure for security, management, accounting etc.
- To standardise data access interfaces through the GGF DAIS WG.
- To provide a reference implementation of the DAIS specification.

The set of portTypes and Grid services that OGSA-DAI defines expose physical data resources in a controlled manner to the Grid. These then represent client access points for physical data resources.

Grid Services and Data Resources

Being able to expose physical data resources to the Grid through Grid services, as illustrated in Figure 1, is an important requirement if Grids are to be widely adopted. Physical data resource capabilities and content should be advertised, located and accessed using Grid service techniques, e.g. Service Data Elements (SDEs)

and *DAI Service Group Registry* (DAISGR) services.

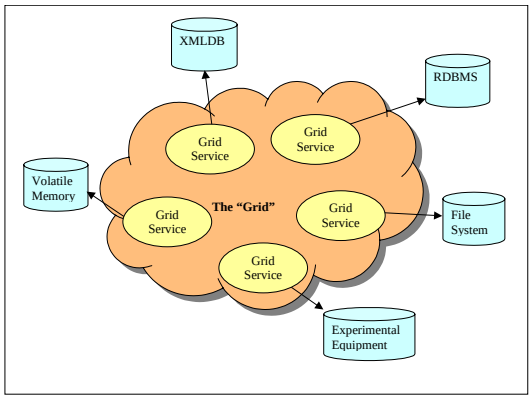

Figure 1: Accessing data resources using Grid services on the Grid

In overview this seems like a satisfactory model. However, when considering the detail it rapidly becomes apparent that physical data resources have inherent structure, associated content and capability meta-data, properties and access mechanisms that must be integrated and exposed to the Grid.

For example, a *Relational Database Management System* (RDBMS) contains databases that, in turn, contain tables. A client may choose to access this structure via a number of different mechanisms, for example, an SQL query, a prepared statement or a call to a stored procedure. A Grid service instance should be able to provide these same access mechanisms to the RDBMS with little overhead.

Grid services must be constructed with the ability to represent the various different internal aspects within a structured data resource. This scenario is illustrated in Figure 2 for the case where the physical data resource is an RDBMS.

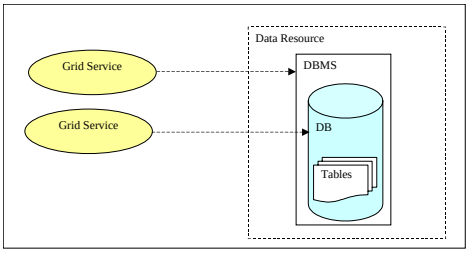

Figure 2: Different levels of Grid service access to a data resource

When considering the number of actions that a Grid service may mediate to a data resource it is clear that the objective of supporting heterogeneous data resources through a consistent interface, while not inventing a new query language, raises challenges. Figure 3 presents the primary mode of operation employed by OGSA-DAI: a Grid service presents some view of a data resource. A query document is submitted to the Grid service, and is evaluated to produce a result document, which is usually returned to the client.

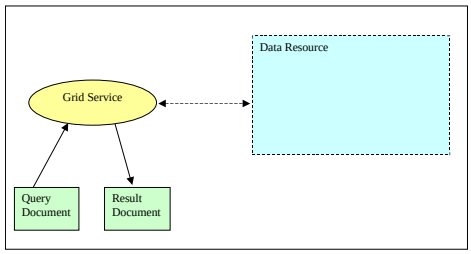

Figure 3: GDS mode of operation

The nature of the query document submitted to the Grid service and the subsequent result document depends on the type of the data resource that the Grid service is configured to represent. For example, a relational database may accept SQL queries while an XML database may accept XPath queries.

Despite the bespoke nature of the queries required to interact with each data resource, the Grid service is able to present a common interface, e.g. the query document system, for managing these queries and for manipulating and delivering any data that is required for/or produced by the query.

OGSA-DAI Service Instances

In order to expose physical data resources to the Grid, OGSA-DAI, by extending the OGSI defined interfaces, have introduced the services described in Table 1.

Service	Description
Grid Data Service (GDS)	Represents a client session with a physical data resource. A GDS is created or instantiated from a GDSF.
Grid Data Service Factory (GDSF)	A GDSF is defined to represent the point of presence of a physical data resource on the Grid. It is through the GDSF service instances that a physical data resource's capabilities and meta-data are exposed.
DAI Service Group Registry (DAISGR)	GDSF instances and capabilities 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 discovery.

Table 1: OGSA-DAI services

Note that currently OGSA-DAI only works with statically-configured data resources. There is no mechanism to dynamically expose data resources once the service container has been started. DAIS is currently scoping out this functionality.

A GDSF instance exposes the capabilities of a physical data resource to the Grid. Any clients that want to interact with the physical data resource have to instantiate a GDS, as depicted in Figure 4. A GDS acts as a session (context holder) with the physical data resource, supporting any inherent activities that can interact with the physical data resource. Data movement to/from the physical data resource takes place as a stream, primarily of XML-structured data. However, the design of the GDS provides the flexibility to develop other modes of operation within the confines of the defined interface.

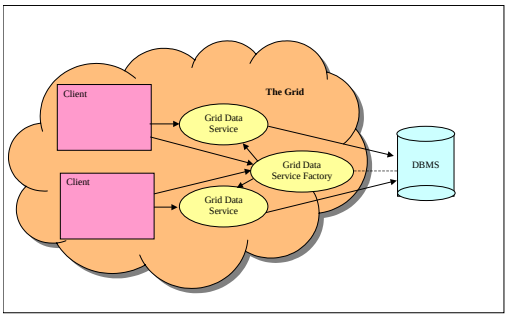

Figure 4: Client access to a data resource through GDSs

A GDS, representing a session with a DBMS, may allow databases/collections to be created and/or removed. A GDS representing a database should allow tables to be created, queried, updated and deleted.

The lifetime of a GDSF will not normally be coupled to the underlying physical data resource that it represents, i.e. if the service instance goes down the underlying physical data resource will not be affected by this change. However, in some instances it might be desirable to couple the lifetime of the service instance with the data it represents. For instance, a GDSF could be used to represent a result set, or other transient data, if this were to be materialized in such a way that it could be used as a data source or sink. In this scenario, the lifetime of the service instance would be tightly coupled to the data it represents – disappearance of one or the other would affect the lifetime of the other. Aspects of this model of a data set are currently being discussed within the DAIS WG. Within the context of OGSA-DAI there is no coupling between the service instance and the underlying physical data resource – if the service disappears the data is still there.

Operations on a Data Resource

GDS interactions with a physical data resource can involve it acting as a data sink, as in the case of an update, and/or a data source, as in the case of a query. The mechanisms for the operations that a client may request are already established and defined by the configuration of the data resource. OGSA-DAI is not defining any new query languages; the GDS is acting as a conduit through which existing query languages may be directed to the physical data resource.

Managing the interaction between a data resource and the Grid involves defining the operations that may be performed on a physical data resource and the data requirements for these operations. In the case of an update operation, data must be delivered to the data source. In the case of a query operation, data may be transported away, via a delivery mechanism, from the data resource.

If the data in question is transported somewhere else in the Grid then a GDSF may be used to represent the data at the destination point. Alternatively, the data may be represented in some other non-Grid-enabled storage system in which case it may be referenced using out-of-band techniques, for example, a URI, or it may be included with the operation request, in the case of an update, or returned with the response in the case of a query.

Consider the case where a GDS is used to request data from a physical data resource:

- If the results are anticipated to be small then the client may request that the data is returned synchronously, i.e. in-lined in the response to the original query. This is unlikely to be satisfactory in most Grid scenarios however, due to the likely data volumes involved.

- Out-of-band delivery mechanisms might be used to transfer data resulting from a query. A new GDSF could then be created against the physical data resource to which results have been delivered - see Figure 5 below[3].

[3] This example assumes that GDSs are able to access file systems, which is not currently in scope for Phase II of the OGSA-DAI project.

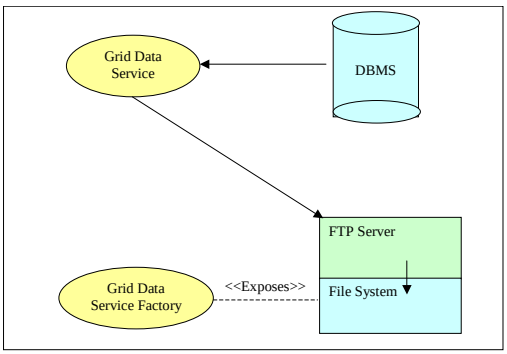

Figure 5: GDS delivery to file system. A GDSF is created to expose the result set. Access to the file system would then be done via GDSs created by that GDSF

- Delivery from one GDS to another may be used as a mechanism for transferring data, as illustrated in Figure 6. The results could then be served by a new GDS.

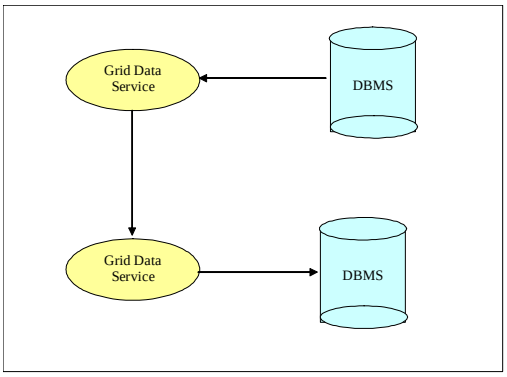

Figure 6: Delivery to a GDS

In this example either:

- The original request is used to configure both service instances, i.e. the request that goes to the GDS is also used to create the second GDS via some GDSF,

Or

- The client must configure each service instance individually.

It is not the intention of OGSA-DAI to build delivery technology or indeed Grid services that represent the data that is being delivered. However, the interface to delivery and the specification of what is to be delivered across a particular interface is of interest to OGSA-DAI.

OGSA-DAI Service Lifecycle

A GDSF must not expose more than one data resource to the Grid. A data resource represents a view of the contents and capabilities of a physical data resource. GDSFs may be configured statically and exposed as persistent services. Configuration of a persistent service requires specifying the database or collection that is to be associated with the instance of the GDSF. GDSFs may advertise themselves in appropriate DAISGRs based on the data resource information they are configured with. The DAISGRs they register with and the data that is registered with each DAISGR are part of a GDSF's configuration.

A client requests a GDSF to create a GDS for a specific data resource, i.e. a data resource advertised by the GDSF. GDSs must be created by GDSFs – there is no support for persistent GDSs. In OGSA-DAI the GDSF cannot process queries directed at the data resource.

DAIS is currently examining frameworks where the GDSF equivalent service may implicitly create a GDS to process queries directed at the GDSF. This is not supported by OGSA-DAI.

A GDSF is created and configured based on the data resource information as specified in a static configuration file. The functions (activities) that a GDS, created by the GDSF, can perform on a data resource are specified in this configuration file. OGSA-DAI has pre-defined a set of functionalities – termed activities. Different activities are available for RDBMS-based data resources as compared to XMLDB-based data resources. These activities are then exposed through an XML document sent to the GDS from the GDSF on creation.

In addition, it is possible for implementers to extend the OGSA-DAI activities to function within the existing OGSA-DAI framework More details of this process are available in the OGSA-DAI user documentation.

The GDS is used by the requesting client and/or advertised in a DAISGR for other clients to use. The GDSF does not automatically register the newly-created GDS – if this is done at all it is viewed as part of the responsibility of the GDS service itself if the client has requested it.

The GDS instance lifetime is controlled using the OGSI-prescribed soft state mechanisms. Each GDS will exist for at least as long as the client-requested lifetime or a GDSF-imposed limit to this, unless the client explicitly destroys the GDS instance. GDS lifetime is not necessarily tied to the lifetime of the underlying data resource. If, for some reason, the data resource becomes unavailable the GDS should report an error to the client when it attempts to access the underlying physical data resource. It is then up to the client to manage the GDS's lifetime. If nothing is done, the GDS will terminate when its allocated lifetime ends.

Figure 7 shows a client interacting with a GDSF in order to create a GDS to access a particular data resource. The client uses the GDS `perform` operation in order to submit requests to the GDS that are likely to result in data being delivered from or to the GDS. When the GDS is no longer required it is destroyed.

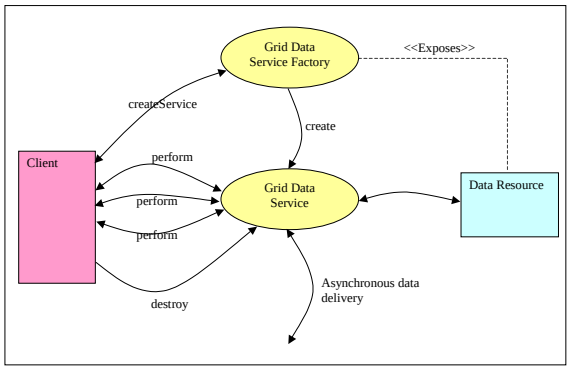

Figure 7: GDS lifecycle

Service Configuration

A physical data resource has been defined to be any physical or logical entity that is able to source or sink data together with any associated management infrastructure. Examples of physical data resource thus are RDBMS, a Database (DB) within a RDBMS, a collection within an XML DBMS or even a directory in a file system.

A GDSF can therefore be statically configured to interact with a physical data resource described by using the above information to create the GDSF representation of the physical data resource.

Meta-Data

Both the GDS and GDSF expose static data from their configuration:

- Meta-data about a data resource's contents.
- Available activities.

This information is "static" as it is presented at the configuration level, initially to the GDSF. However, there is potentially a dynamic aspect to the information. If a DBMS is removed or if a database is added to the DBMS this information should be changed to reflect this. Currently this is a manual configuration task and the GDSF would have to be restarted to take this into account.

Provision has been made in the GDSF configuration file to allow function callbacks to be defined and implementations to be specified that can be used to dynamically generate meta-data content, e.g. the schema for a database. This framework is potentially extensible to allow other dynamic content to be provided.

The GDS may also expose information obtained from the data resource itself, for example, the structure of any data stored. This information can be organised in accordance with the data resource information in the GDS configuration.

Data Integration

Data access has been the primary focus for OGSA-DAI. The wider project scope covers data integration also. In the Grid context, data integration means applying virtualization to present a coherent data resource to the client. This can take many forms. Here are three examples:

1. Choosing a suitable data replica based on some criteria, for example, "closest" or least "expensive".

2. Parallel downloads from data resource replicas or copies to improve performance.

3. Running a query across distributed data resources presented to the user as a single data resource.

In all of these examples some functionality is required over and above the basic access operations. OGSA-DAI, in the current phase of the project, has only attempted to examine example (3) above through the distributed query processor (DQP) work. Currently, the underlying patterns of data access to produce data integration, other than through DQP, must be achieved manually, i.e. using a client directly to direct the work and data flow.

Current Release

The current release of the OGSA-DAI software contains enough functionality for projects to trial the OGSA-DAI services in their own environments. In particular, application developers will be able to evaluate their requirements for data access and integration and begin to develop higher-level services and applications using OGSA-DAI. We hope this will generate feedback, which will be valuable in shaping the direction of future releases.

Future Directions

The focus of the current phase of development has been to develop a production-quality software framework in which atomic GDS operations can be performed, either as single requests or as batches of activities representing a logical unit of work.

The software enables applications to be built without any data access and integration being hard-wired. In consequence, applications built using GDSs should be more flexible, both in terms of what data can be discovered and accessed, and what operations can be formed on data. It is hoped that science communities will define and publish generic GDSs, which expose their data to other users for reuse by other application developers.

The next phase of development, DAIT, will focus on:

- Extending the existing OGSA-DAI functionality on an agreed-priorities basis.
- Providing sufficient flexibility to enable substitution of technology components on a plug-and-play basis.
- Improving the overall quality of software in terms of reliability, performance, and scalability.
- Extending support to more software platforms.
- Composition of higher-level services to establish reusable programming platforms and improved manageability.
- Aligning with DAIS once the standardization process has stabilized.

More information about OGSA-DAI may be obtained from the project web site [4] or by using the contact information on page 1 to get in touch with the OGSA-DAI team. If you have suggestions as to how you would like to see OGSA-DAI evolve or want to use OGSA-DAI within your project then please do get in touch.

References

[1] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Snelling, D., Vanderbilt, P. *Open Grid Service Infrastructure*. Version 1.0.

[2] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Vanderbilt, P. *Grid Service Specification*. Draft 4. October 4th 2002.

[3] Foster, I., Kesselman, C., Nick, J., and Tuecke, S. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems*. http://www.gridforum.org/ogsa-wg/

[4] OGSA-DAI Project website http://www.ogsadai.org.uk