Grid Computing

Open Grid Services Architecture – Data Access and Integration OGSA-DAI

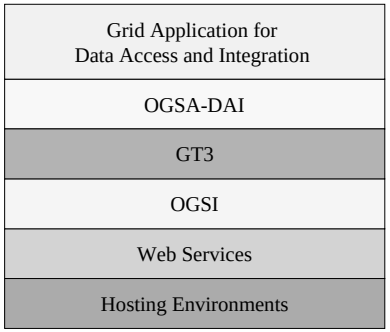
Paul A. Farrell
Fall 2006

The Grid: Core Technologies
Maozhen Li, Mark Baker
John Wiley & Sons; 2005, ISBN 0-470-09417-6

Open Grid Services Architecture-Data Access and Integration (OGSA-DAI)

- Middleware technology to access and integrate different data sources such as relational databases, XML databases, and file systems on the Grid
- Both specification and implementation
- Specification: defines services and interfaces for data access and integration on the Grid
  - extended WSDL portTypes based on the OGSI specification
- Implementation: implements the interfaces in the OGSA-DAI framework based on GT3
  - Aim to incorporate external data resources within OGSA framework and allow access via standard Grid services interfaces
  - supports the registration/discovery of databases and interaction with those databases.

OGSA-DAI



OGSA-DAI portTypes

- A Grid service in the OGSA-DAI is called a *Grid Data Service* (GDS)
- A GDS must implement the *GDSPortType* interface
- *GDSPortType* portType :
  - supports data access, integration and delivery
  - Extends three portTypes, *GridService* portType specified in OGSI, *GridDataPerform* (GDP) portType and *GridDataTransport portType* defined by OGSA-DAI
- *GridDataPerform* portType
  - provides methods for clients to access data sources and retrieve results
  - supports document-oriented interface in which a query request is submitted using Grid Data Service Perform (GDS-Perform) documents to specify the operations on the data sources and a response is returned using Grid Data Service Response (GDS-Response) documents containing information on the results

OGSA-DAI portTypes

- *GridDataTransport* portType
  - provides supports for data transfer between OGSA-DAI services, and between OGSA-DAI client and the OGSA-DAI services
  - allows data to be pushed or pulled
  - derived from *GridService* portType in OGSI
  - provides the following methods
    - PutFully() – transfer a complete set of data.
    - GetFully() – receive a complete set of data.
    - PutBlock() – transfer a block of data, part of a larger batch of data specifying the index of the block.
    - GetBlock() – receive a block of data, part of a larger batch of data specifying the index of the block

OGSA-DAI portTypes

- *GridDataServiceFactory* portType
  - used to implement a Grid Data Service Factory (GDSF), a persistent Grid data service used to create GDSs
- *DAIServiceGroupRegistry* portType
  - used to implement a DAI Service Group Registry (DAISGR). A DAISGR service can be used to register any services that implement one or more OGSA-DAI portTypes

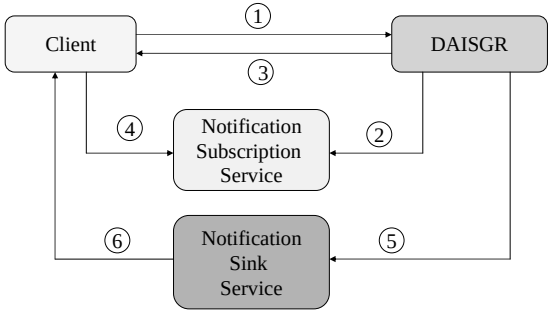
OGSA-DAI Functionality

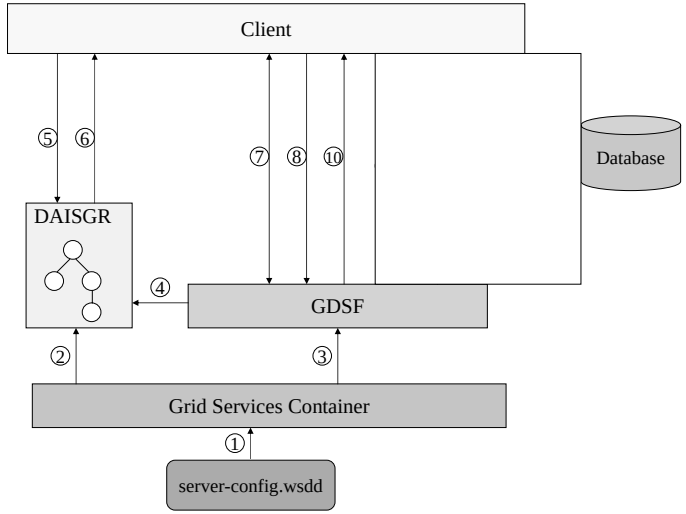
- **Lifetime Management of GDS Instances**
  - **A GDS instance is a transient service created by a GDSF associated with a data source. A GDS instance can be dynamically created and explicitly destroyed.**
- **Service Registration/Unregistration**
  - **A OGSA-DAI service can register itself with a DAISGR via the ServiceGroupRegistration::Add() method. A DAISGR can also be registered in another DAISGR.**
  - **By querying a DAISGR, clients can discover OGSA-DAI services that offer particular services or capabilities or manage particular data sources.**
  - **Registered services in a DAISGR can also be unregistered via the ServiceGroupRegistration::Remove() method.**
- **Service Discovery**
  - **A client can query a DAISGR via the GridService::FindServiceData() method to discover an OGSA-DAI service meeting its requirements.**
  - **A client can then query the OGSA-DAI service directly and also query the OGSI ServiceGroupEntry (SGE) service managing the OGSA-DAI service's registration. There are three purposes for a client to query a DAISGR:**
    - **To discover a GDSF to create GDS instances for specific applications.**
    - **To query an OGSA-DAI service instance to retrieve its state information.**
    - **To query the OGSI ServiceGroupEntry associated with an OGSA-DAI service to retrieve the OGSA-DAI service's registration.**

OGSA-DAI Functionality

- OGSA-DAI Service Notification
  1. A client uses the NotificationSource::Subscribe() method to subscribe to a DAISGR to specify the events of interest for notifications. In the subscription, it specifies the location of a notification sink service which implements the *NotificationSink* interface.
  2. The DAISGR creates a notification subscription service which implements the *NotificationSubscription* interface to manage the subscription.
  3. The DAISGR informs the client of the identity of this NotificationSubscription service.
  4. The client queries the notification subscription service via GridService::FindServiceData() method to manage its subscription, e.g., its lifetime management.
  5. Once the DAISGR has some changes in its state, it will notify the notification sink service via the NotificationSink::DeliverNotification() method.
  6. The notification sink service will send the notification messages to the client

OGSA-DAI Service Notification





1. Start a Grid service container, which reads a server-config.wsdd file. The server-conf.wsdd file allows the Grid service container to access information on services to be deployed and to map between service names and the associated classes and XML elements and associated Java classes.
2. The Grid service container creates a persistent DAISGR based on the GSH specified in the server-config.wsdd file.
3. The Grid service container creates a persistent GDSF based on the GSH specified in the server-config.wsdd file.
4. The GDSF registers itself in the DAISGR with the ServiceGroupRegistration::Add() method.
5. The client queries the DAISGR using the GridService::FindServiceData() method. The client selects a registered GDSF.
6. The DAISGR returns the GSH of a selected GDSF.
7. The client can query the service data elements of the GDSF to retrieve its configuration information.
8. The client calls the GDSF Factory::createService() method to create a GDS instance.
9. The GDSF creates the GDS instance.
10. The GDSF returns the GSH of the GDS instance to the client.
11. The client queries the service data elements of the newly created GDS instance using the GridService::FindServiceData() method to establish its configuration and the schema that describes the GDS-Perform documents that may be submitted through the GridDataPerform::perform() method.
12. The client submits a GDS-Perform document to the GDS instance.
13. The GDS instance accesses a database to get some data, and generate a GDS-Response document.
14. The GDS instance returns the GDS-Response document to the client.
15. The client destroys the GDS instance via GridService::Destroy() method.

OGSA-DAI and DAIS

- The Database Access and Integration Services (DAIS) working group within the GGF seeks to promote OGSA compliant standards for Grid database services
  - does not seek to develop new data storage systems, but to make such systems more readily usable
- OGSA-DAI is a collaborative programme involving the Universities of Edinburgh, Manchester and Newcastle, with IBM and Oracle
  - objective is to produce open source database access and integration middleware that meets the needs of the UK e-Science community
  - scope includes the definition and development of generic Grid data services providing access to, and integration of data, held in relational database management systems, as well as semi-structured data held in XML repositories