Writing Perform Documents

EPCC, University of Edinburgh
Amy Krause (a.krause@epcc.ed.ac.uk)
Tom Sugden (tom@epcc.ed.ac.uk)

First International Summer School on
Grid Computing, Vico Equense, Italy

Talk Outline

- Introduction and Motivation
- Structure of perform documents
- Database queries and updates
- Parameterisation
- Result transformations
- Delivery mechanisms

Introduction

- Interface to GDS is document-based
- *GridDataService::perform* takes a perform document as its input parameter
- Generates a response document

Motivation

- To access and integrate with data resources using OGSA-DAI:
  - Write perform documents

    OR
  - Write client utilities to generate perform documents

What is a perform document?

- A perform document contains a series of activities for a GDS to perform
  - Query a database
  - Transform the results
  - Deliver the transformed results
- Expressed in XML

XML Structure

- Root element
  - `<gridDataServicePerform>`
- Top-level elements correspond to activities
  - `<sqlQueryStatement>`
    - Performs an SQL query against the data resource
  - `<xslTransform>`
    - Transforms XML data from one structure to another
  - `<deliverToURL>`
    - Delivers data to a URL.

Perform Document Validation

- Each activity element must validate against an activity schema

```
<xsd:complexType name="SQLQueryStatementType">
  <xsd:complexContent>
    <xsd:extension base="gds:ActivityType">
      <xsd:sequence>
        <xsd:element name="sqlParameter"
          minOccurs="0" maxOccurs="unbounded">...
        <xsd:element name="expression"
          minOccurs="1" maxOccurs="1">...
        <xsd:element name="webRowSetStream"
          minOccurs="1" maxOccurs="1">...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Example

```
<?xml version="1.0"?>
<gridDataServicePerform>
  <documentation>
     Performs a simple SQL query and returns the
     results in the response document.
  </documentation>
  <sqlQueryStatement name="statement">
    <expression>
      select * from littleblackbook where id=10
    </expression>
    <webRowsetStream name="statementOutput"/>
  </sqlQueryStatement>
</gridDataServicePerform>
```

Database Queries

- OGSA-DAI R3 includes the following activities for querying databases
  - sqlQueryStatement
    Queries a JDBC compliant database
  - xPathStatement
    Queries an XML:DB compliant database
- Future releases to support
  - XQuery for XML database
  - Querying file systems and files
- Examples provided:
  - <OGSA_DAI>\examples\GDSPerform\JDBC\query
  - <OGSA_DAI>\examples\GDSPerform\XMLDB\xpath

Example: sqlQueryStatement

```
<?xml version="1.0"?>
<gridDataServicePerform>
  <documentation>
    Performs a simple SQL query and returns the
    results within the response document.
  </documentation>
  <sqlQueryStatement name="statement">
    <expression>
      select * from littleblackbook where id=10
    </expression>
    <webRowsetStream name="statementOutput"/>
  </sqlQueryStatement>
</gridDataServicePerform>
```

Example: xPathStatement

```
<?xml version="1.0"?>
<gridDataServicePerform>
  <documentation>
    Performs a simple XPath statement and delivers
    the results within the response document.
  </documentation>
  <xPathStatement name="statement">
    <!--<collection>subCollection</collection>-->
    <!--<resourceId>someResourceId</resourceId>-->
    <!--<namespace prefix="ns">...</namespace>-->
    <expression>/entry[@id&lt;100]</expression>
    <resourceSetStream name="statementOutput"/>
  </xPathStatement>
</gridDataServicePerform>
```

Database Updates

- OGSA-DAI R3 includes the following activities for updating databases
  - sqlUpdateStatement
    Updates a JDBC compliant database using SQL notation
  - xUpdateStatement
    Updates an XML:DB compliant database using XUpdate notation
- Examples provided:
  - <OGSA_DAI>\examples\GDSPerform\JDBC\update
  - <OGSA_DAI>\examples\GDSPerform\XMLDB\xupdate

Database Management

- OGSA-DAI R3 includes the following activities for database management
  - relationalResourceManagement
    Creates and drops databases within a JDBC compliant DBMS
  - xmlCollectionManagement
    Creates, removes and lists collections within an XML:DB compliant DBMS
  - xmlResourceManagement
    Creates, removes and lists the resources contained in collections within XML:DB compliant DBMS
- Examples provided
  - <OGSA_DAI>\examples\GDSPerform\XMLDB\collectionManager
  - <OGSA_DAI>\examples\GDSPerform\XMLDB\resourceManager

Example: xmlCollectionManagement

```
<?xml version="1.0"?>
<gridDataServicePerform>
  <documentation>
    Performs a simple XML:DB collection
    management operation.
  </documentation>
  <xmlCollectionManagement name="statement">
    <!--<collection>subCollection</collection>-->
    <createCollection name="frogSpecimens"/>
    <!--<removeCollection name="frogSpecimens"/>-->
    <!--<listCollections/>-->
    <resourceSetStream name="statementOutput"/>
  </xPathStatement>
</gridDataServicePerform>
```

Transformation Activities

- OGSA-DAI R3 includes the following activities for transforming data
  - xslTransform

    Transforms XML data from one structure to another using an XML Stylesheet Language Transformation
  - gzipCompression

    Compresses data using the GZIP format
  - zipArchive

    Archives and compresses data using the ZIP format
- Examples provided
  - <OGSA_DAI>\examples\GDSPerform\XMLDB\transform

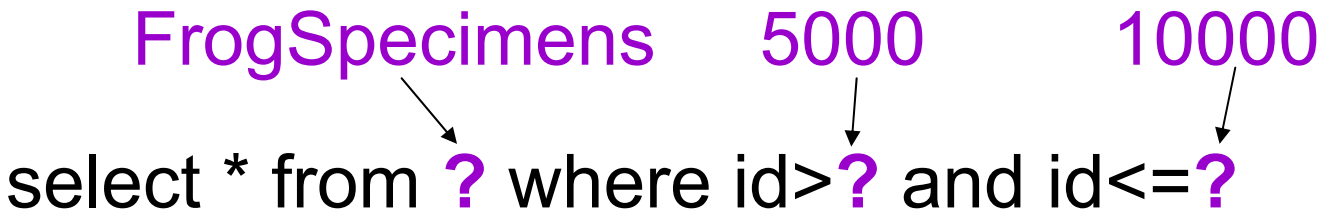
Example: xslTransformActivity

```
<gridDataServicePerform>
  <deliverFromURL name="deliverXSLT">
    <fromURL>somewhere/transform.xsl</fromURL>
    <toLocal name="deliverXSLTOutput"/>
  </deliverFromURL>
  <xPathStatement name="statement">
    <expression>/entry[@id&lt;100]</expression>
    <resourceSetStream name="statementOutput"/>
  </xPathStatement>
  <xslTransform name="transform">
    <inputXSLT from="deliverXSLTOutput"/>
    <inputXML from="statementOutput"/>
    <output name="transformedOutput"/>
  </xslTransform>
</gridDataServicePerform>
```

Transforms the results of a query into HTML

Parameterisation

- Parameterisation allows data to be inserted into a query or update expression



- Currently supported for SQL query and update
  - sqlQueryStatement
  - sqlUpdateStatement

Using Parameters

- *sqlParameter* references the output stream of another activity
  - dataStore activity
  - deliverFrom activity
- references the position of the parameter

```
<sqlUpdateStatement name="statement">
   <sqlParameter position="1" from="dataToInsert"/>
   <expression>
      insert into MyTable values ?
   </expression>
</sqlUpdateStatement>
```

Usings Data Stores

- Parameter values can be specified using the dataStore activity

```
<dataStore name="myDataStore">
  <item>FrogSpecimens</item>
  <item>5000</item>
  <item>10000</item>
  <itemCursor name="myDataStoreOutput"/>
</dataStore>
```

- sqlParameter references the itemCursor name

Example: sqlQueryStatement with Parameters

```
<gridDataServicePerform>
  <dataStore name="myDataStore">
    <item>FrogSpecimens</item>
    <item>5000</item><item>10000</item>
    <itemCursor name="parametersOutput"/>
  </dataStore>
  <sqlQueryStatement name="statement">
    <sqlParameter position="1" from="parametersOutput"/>
    <sqlParameter position="2" from="parametersOutput"/>
    <sqlParameter position="3" from="parametersOutput"/>
    <expression>select * from ? where id>? And id<=?</exprssn>
    <webRowsetStream name="statementOutput"/>
  </sqlQueryStatement>
</gridDataServicePerform>
```

Delivery Activities

- Activity output can be delivered:
  - Synchronously in the response document (default behaviour)
  - Asynchronously using delivery activities
- OGSA-DAI R3 includes the following asynchronous delivery activities:
  - deliverToGDT and deliverFromGDT

    Delivers data to and from a GDS using the Grid Data Transport port type.
  - deliverToGFTP and deliverFromGFTP

    Delivers data between GDS and GFTP location in both directions.
  - deliverToURL and deliverFromURL

    Delivers data between GDS and URL in both directions (ftp, http, https, file)

Delivery Structure

Delivery - Push

- Input Stream: *fromLocal* – references the name of another activity
- Output Stream: *toURL, toGFTP, toGDT* – where to push the data to

Delivery – Pull

- Input Stream: *fromURL*, *fromGFTP*, *fromGDT* – where to pull the data from
- OutputStream: to*Local* – references the name of another activity

Example Query with Asynchronous Delivery to URL

```
<gridDataServicePerform>
  <xPathStatement name="statement">
    <expression>/entry[@id&lt;100]</expression>
    <resourceSetStream name="statementOutput"/>
  </xPathStatement>
  <deliverToURL name="delivery">
    <fromLocal from="statementOutput"/>
    <toURL>
       file://c:/ogsadai/resultsets/newresultset.txt
    </toURL>
  </deliverToURL>
</gridDataServicePerform>
```

Practicals

- Practical exercises to try out:
  - Synchronous request
  - Parameterisation
  - Transformations
  - Asynchronous delivery

http://192.167.1.106:8080/workshop

- Further examples
  - <OGSA_DAI>/examples/GDSPerform