

Distributed Query Processing with OGSA-DQP

Principles and Architectures for
Structured Data Integration: OGSA-DAI as an example

ISSGC06 (Ischia, Italy)

17 July 2006

Slides thanks to Steve Lynden
Amy Krause
EPCC

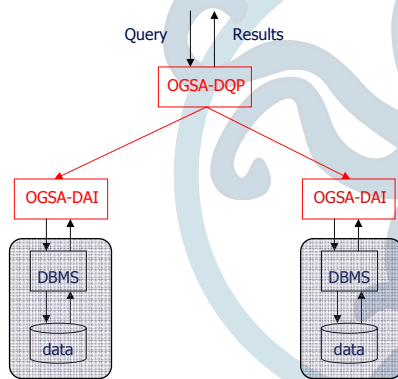


Introduction

epcc

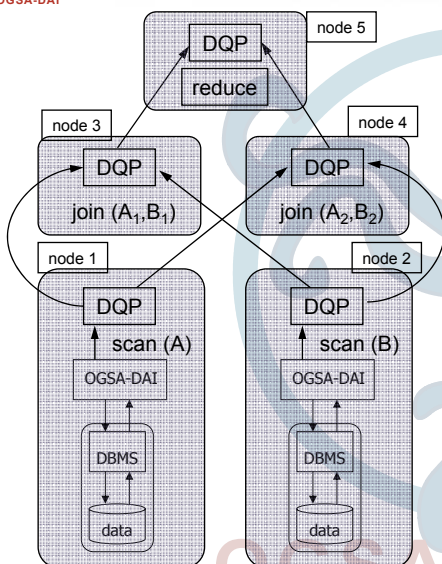
- OGSA-DQP is a service based distributed query processor
- It evaluates queries over distributed data sources wrapped by OGSA-DAI
- It is built using OGSA-DAI extensibility points
- People involved:
 - University of Manchester
 - Steven Lynden, Alvaro Fernandes, Rizos Sakellariou, Norman Paton
 - University of Newcastle
 - Jim Smith, Arijit Mukherjee, Paul Watson
 - OGSA-DAI
- Prototype release 3.0 available from the OGSA-DAI website
- Release 3.1 will available soon

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



- OGSA-DQP uses a middleware approach.
- It can be seen as a mediator over OGSA-DAI wrappers.
- Effectiveness: “leave to it to orchestrate your services”;
- Usability: “use it as an OGSA-DAI data service”.

OGSA-DAI



- OGSA-DQP queries can be evaluated across multiple nodes by DQP services deployed on those nodes
- Operators can be parallelised, e.g. a join can be executed across two nodes
- OGSA-DQP compiles, optimises and schedules queries for execution across available nodes
- An OGSA-DQP query is separated into a number of *partitions*, each of which encapsulates an individual service's role in the query evaluation

OGSA-DAI



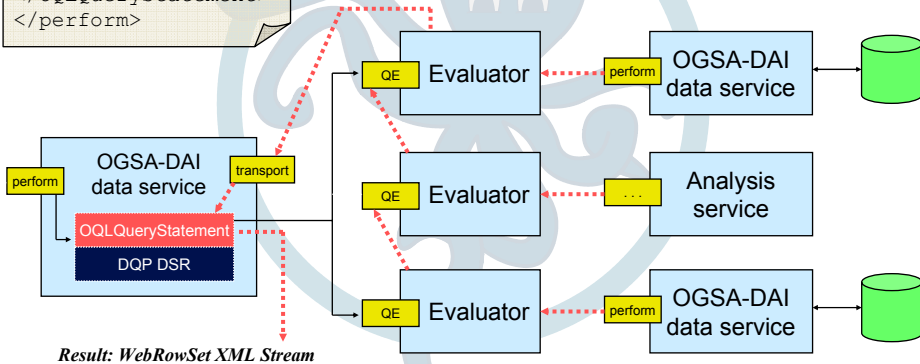
- Given two DBMSs and one analysis tool (i.e., a Web service):
 - **goTerm** : a GO Gene Ontology table within a MySQL DB, exposed by an OGSA-DAI data service
 - **protein** : a protein sequence table within a MySQL DB, exposed by an OGSA-DAI data service
 - **Blast** (sequence alignment scoring Web service);
- We want to obtain alignment scores for a sequence against proteins of a certain kind
- The user submits a single query referencing data stored at multiple sites.
- The author of the query need not be aware of how/where data is stored.
- Queries are written in Object Query Language (OQL):

```
select p.proteinId, Blast(p.sequence)
from protein p, goTerm t
where t.termId = 'GO:0005942' and p.proteinId=t.proteinId
```



- DQP evaluator services:
 - Are plain Web services
 - Implement the QueryEvaluation port type:
 - **evaluate** – the input is a query plan partition which is subsequently executed
 - **receiveData** – allows the evaluator to receive data from other evaluators
- OGSA-DAI extensions:
 - DQP resource – a resource which encapsulates a distributed query infrastructure: DQP evaluator services, OGSA-DAI data services etc. Implemented as a data resource accessor.
 - OQL query statement activity – enables the submission of a query in Object Query Language (OQL)
 - DQP factory activity – enables the creation and configuration of DQP resources.

```
<perform>
<OQLQueryStatement>
<expression>
OQL query
</expression>
</OQLQueryStatement>
</perform>
```



- OGSA-DQP is a service based distributed query processor that is:
 - Exposed as a service
 - Implemented as an orchestration of services
- It provides an example of how the OGSA-DAI extensibility points can be used...
 - The activity extensibility points are used
 - New data resource accessors are implemented
 - Dynamic resource deployment is used during configuration to create new resources
- Benefits:
 - OGSA-DAI manages activity concurrency – we didn't need to write concurrent code
 - OGSA-DQP can take advantage of the host of delivery options provided by OGSA-DAI
 - OGSA-DQP is insulated from multiple platforms (WS-I, WSRF) by OGSA-DAI