NextGRID & OGSA Data Architectures: Example Scenarios

Stephen Davey, NeSC, UK

ISSGC06 Summer School, Ischia, Italy
12th July 2006
Contributors & Acknowledgments

This presentation is based on work by

- Stephen Davey *et al.*, “OGSA Data Scenarios”

- Allen Luniewski, Dave Berry *et al.*, “OGSA Data Architecture”

With additional thanks to

- NextGRID Architecture WP1, OGSA Data Working Group.
Introduction - Aim & Scope

These slides cover the following:

- Example Data Scenarios
  - Data Storage
  - Data Replication
  - Data Staging
  - Data Pipelining

- Data Components & Architectural Context
  - NextGRID Data Architecture
  - OGSA Data Architecture
Data Scenarios

- Purpose of the Scenarios
  - Example scenarios of a generic nature to accompany the OGSA Data Architecture document.
  - Not a use case document generating requirements for the OGSA Data Architecture.
  - Instead provides illustrations of how the components and interfaces described in the OGSA Data Architecture document can be put together in a selection of typical data scenarios.
Scenarios done so far …

- **Data Storage** – store file data in a Grid Data Service and retrieve it later.
- **Data Replication** – maintain a replica of data at a different location (for availability or performance).
- **Data Staging** – the movement of data in preparation for the performing of operations on or with this data.
- **Data Pipelining** – connect the output from one service to the input of another.

To be covered next week:

- **Data Integration** – bringing the data that you require together from disparate sources. [See OGSA-DAI sessions 26, 27].
- **Personal Data Service** – the organising of an individual’s data to allow them access to it from many different locations. [See sessions 32, 33; myGrid etc.].
- **Data Discovery** – discover data; register data/metadata. [See Ontologies & Semantic grids sessions 32, 33].
Data Storage Scenario

- Use Case 1: Writing a file into storage
  1. The customer requests file storage space on the Data Storage Service to which the file can be written.
  2. The customer requests a file name (SURL) from the Data Storage Service for the given space to write a file. The Data Storage Service returns a valid SURL.
  3. Using the file name, the client requests a file URL (reference) with some specific parameters (protocol, security tokens, etc) with which the file can be actually written. The Data Storage Service returns a valid Transfer URL (TURL). The TURL may also be an Access URL (i.e. for POSIX access as opposed to transfer).
  4. The customer makes use of the service that supports the requested protocol to actually write the file into the given space on storage using the TURL. This may be through:
     a) The Data Storage Service directly,
     b) or the Data Access Service,
     c) or the Data Transfer Service.
  5. The customer notifies the storage at the end of the operation that the write is complete. Data Storage Service acknowledges completion.
Data Storage – Writing a file

1. Request file space.
2. Get file name (SURL).
3. Get Transfer URL (TURL) or Access URL.
   4a. Write file.
   4b. Write file.
   4c. Write file.
5. Notify of completion.
Data Storage Scenario 2

Use Case 2: Make data available online.
The customer has the file names for a set of files in a given space and requires that these files should be available online.

1. The files are made available online by the Data Storage Service.
2. The data are read through an appropriate interface, such as the Transfer Service.
3. The online attribute of the files may expire and they can be retired to nearline storage.
Data Storage – Make online

1. Make online.
2. Read files.
3. Retire to nearline.

Customer

Data Storage Service

Nearline Storage

Online Storage

Transfer Service

Storage Devices
1. A data resource is registered with a replicating data service (details such as creation time, access control, etc. would also be included) and replication service enters the data resource into a replica catalogue.

2. The replication service uses a data transfer service to move copies of this data to different locations and tracks which data is kept where.

3. Clients access the catalogue to find the data resource, or to return a list of resources that satisfy certain Quality of Service (QoS) requirements.

4. Clients then access the stores either directly or indirectly.

5. Changes to the data are notified to the replication service.

6. Updates then occur between the data services to synchronize the replicas.
Data Replication – 1

1. Register data
2. Transfer copies
3. Find data
4. Access data
5. Notify
6. Update

Data Storage 1
Data Storage 2

Data Service 1
Data Service 2

Replication Service
Registry Service

Customer 1
Customer 2
Data Replication – 2

1. Register
2. Transfer copies
3. Find data
4. Access data
5. Notify
6. Update

Customer 1

Data Service
Replication Service
Replica Catalogue Service

Customer 2

Data Service 1
Data Storage 1

Data Service 2
Data Storage 2
Data Staging Scenario

1. Customer 1 submits a parameter space exploration job to the Parameter Space Exploration Service.

2. An optimized copy (bulk load) of the boundary conditions data is made from the Parameter Space Exploration Service to the Simulation Service, utilising a Data Service to assist in the extraction and transfer of the data. This step would actually have 3 parts:
   a) Firstly, storage space needs to be reserved through the Simulation Service with the corresponding EPR for the storage being returned to the Parameter Space Exploration Service.
   b) Secondly, the Parameter Space Exploration Service queries the Boundary Conditions database for the relevant data.
   c) Finally the Data Service bulk loads the boundary condition data to the Simulation Service.

3. The Simulation Service sets up the results database.
Data Staging Scenario (cont.)

4. From the parameter set the simulation jobs are generated and sent to the Simulation Service. Each of the jobs will take parameters from the parameter set database and then read the boundary condition data from the local copy of the boundary conditions database.
5. Results from the Simulation Service are stored in the results database.
6. On completion of all the generated jobs the Simulation Service’s local copy of the boundary conditions database is deleted.
7. Queries (or jobs) are used to get derivatives from the results database.
8. The simulation service returns the derived data to the consumer.
9. On completion of all queries the simulation service deletes the results set database.
Data Staging

1. Submit job.
2a. Get EPR for storage & CPUs.
2b. Query relevant boundary conditions.
2c. Bulk load boundary condition data.
3. Set up Results DB.
4. Generated jobs from parameter set.
5. Store results.
6. Delete boundary condition data.
7. Query results set.
8. Return derived data.
9. Delete Results DB.

Customer 1

Parameter Space Exploration Service

Parameter Set

Simulation Service

Results Set

Boundary Conditions

Data Service 1

Data Service 2

Boundary Conditions (copy)
Data Pipelining Scenario

1. Customer 1 (Designer) submits a rendering job to the Rendering Service.
2. Completed animation is stored to a common storage device.
3. Rendering Service transfers the completed animations (data) to the Visualization Service using the Data Transfer Service.
4. The Visualization Service displays the animations to the customers (Designer & Reviewer) in an agreed format.
Data Pipelining

1. Submit job.
2. Store results.
3. Transfer results.
4. Return results.

Customer 1

Customer 2

Rendering Service

Data Transfer Service

Data Service

Completed Animations

Visualisation Service
Summary of Data Components

- Capabilities that can be provided by the data architecture include:
  - **Data transfer**
    - infrastructure for transferring data between services and/or resources.
  - **Data access**
    - methods of accessing data, whether that data is stored locally or remotely.
  - **Data location management**
    - staging, caching and replicating data resources.
  - **Data federation**
    - integrating multiple data resources so that they can be accessed as if they were a single resource.

- **Data description**
  - The types of data (both simple and compound) under consideration and how those types are specified.

- **Policies**
  - quality of service (QoS), protocols and coherency conditions
Basic structure of a data architecture

Client APIs (non-OGSA) / Other services

Transfer

Lookup

Registries

Sink/Source

Description

Sink/Source

Description

Access

Other Data Services

Data Management

Managed Storage

Storage Management

Storage

Access

Client APIs (non-OGSA) / Other services

Managed Storage

Storage Management

Data Management

Other Data Services

Transfer Protocols

Key:

- **Interface**: An API or service calling an interface
- **Service**: A service using a resource
- **Resource**: Transfer of data between resources

From: “The Open Grid Services Architecture, Version 1.6”
Architectural Context

- NextGRID data architecture
  - Within framework provided by OGSA WSRF Base Profile (and built on Web Services)
    - provides the default messaging layers and service specification languages
    - management of distributed resources
    - addressing
    - notification of events

- Naming
- Registries and resource discovery
- Security & Trust
- Policies and agreements
NextGRID Interactions

- Registry
  - Register / Update / Query
  - Query
  - Get token assertions
  - Orchestration
    - Invoke
    - Resolve
    - Generate / Verify
    - Get tokens
    - Trust and Security
      - Get token assertions
    - Naming and Addressing
      - Get token assertions
    - Monitor / Control
      - Negotiate SLA
        - Get token assertions
      - SLA Management
        - Get token assertions
      - Administer policy
  - Functional
    - Get token assertions
    - Register / Update
Questions?

- Data Scenarios
  - Data Storage
  - Data Replication
  - Data Staging
  - Data Pipelining
- Data Architecture & Context