A Bright Future with OGSA Data Services

Malcolm Atkinson
Director

www.nesc.ac.uk

7th June 2004
OGSA-DAI

Request to Registry for sources of data about “x”
Registry responds with Factory handle
Request to Factory for access to database
Factory creates GridDataService
Factory returns handle of GDS to client
Client queries GDS with SQL, XPath, XQuery etc
Query results returned XML
OR delivered to consumer as XML

Analyst

Registry
GDSR

Factory
GDSF

Grid Data
Service
GDS

Database
(Xindice, MySQL, Oracle, DB2)

Consumer

GDS interacts with database

SOAP/HTTP
service creation
API interactions
Extensibility

- Data resources
  - Unbounded variety
- Data access languages
  - Established standards
    - With many variants
  - SQL, OQL, semi-structured query, domain languages
- Investment in DBs, DBMSs, File Stores, Bulk stores, ...
  - Not sensible to expect them to change to fit us
- Data Access Models must be extensible
  - Static extension used extensively by OGSA-DAI users

Should extensibility be supported by foundation interfaces?
Move Computation to Data

- **Code scale**
  - Depends on wet-ware
    - No noticeable rate of improvement

- **Data scale**
  - Grows Moore’s Law or Moore’s Law^2

- **Analysis of data**
  - Extracts & derivatives used
    - Often smaller - more value for current investigation

- **Implies move code to data**
  - SQL, Xquery, Java code, ...

- **Extensibility mechanisms used by OGSA-DAIers**
  - Java mobility (e.g. DataCutter), database procedures, ...

Increasingly necessary

Application control or higher-level service decisions

Application control or higher-level service decisions
Integration is Everything

- No business or research team is satisfied with one data resource
- Domain-specialist driven
  - Dynamic specification of combination functions
  - Iterative processes - range of time scales
- Sources inevitably heterogeneous
  - Content, structure & policies time-varying
- Robust & stable steerable integration services
  - Higher-level services over multiple resources
  - Fundamental requirements for (re)negotiation

Federation or Virtualisation preceding integration or kit of integration tools to be interwoven with an application?
Multiple tasks / request

Data Set

Data Set

Ident Type Value
Ident Type Value
Ident Type Value
Ident Type Value
Ident Type Value
Ident Type Value
Ident Type Value
Ident Type Value

1

2

dr

CLIENT API REQUEST SET
Be Direct

- Double Handling costs too much
  - Memory cycles, bus capacity, cache disruption, …
- Double Handling via discs pathologically bad
- Data translation expensive
  - Avoid or compose
- Main memory is not big enough
- Couple generator & consumer directly
  - Data pipe from RAM to RAM
  - Requires coupled computation execution

Breaks down boundaries and merges data, execution & transport requirements.

Demands smart workflow enactment service & foundation services