Integration of WS-DAIR into gLite-AMGA

A. Javadzadeh Boloori, KTH

B. Koblitz, CERN

OGF23, Barcelona

June 2end, 2008

www.eu-egee.org

Information Society and Media

Overview

- AMGA Metadata Catalogue
- WS-DAIR compatible interface for AMGA
- Features
- Evaluation: Avian flue drug discovery application
- Future plans and comments
- Conclusion

AMGA- Introduction

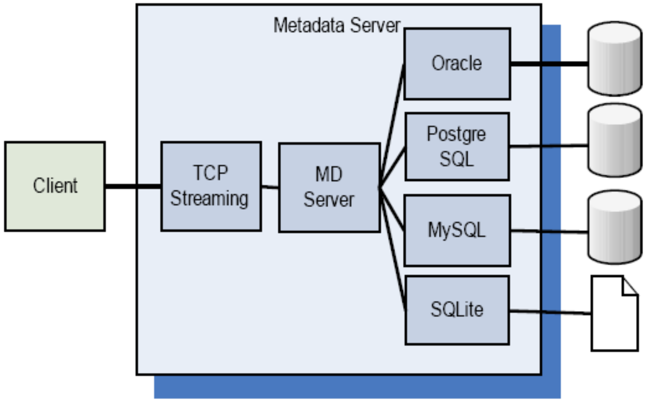
- **AMGA is Metadata catalogue of EGEE's gLite 3.1 MW**

(Used as a file catalogue, job submission, ...)

- **AMGA brings Grid-Idea to relational Dbs**
  - AMGA hides DB differences
  - AMGA allows replication and (some) federation of data
  - AMGA has fine-grained access control to entries based on ACLs and uses VOMS authentication

AMGA- Introduction

- **To achieve this, AMGA has**
  - Proprietary (but well documented) TCP-Streaming protocol (WAN performance)
  - Proprietary SQL inspired query language
    - Hides DB differences
    - Allows access control layer

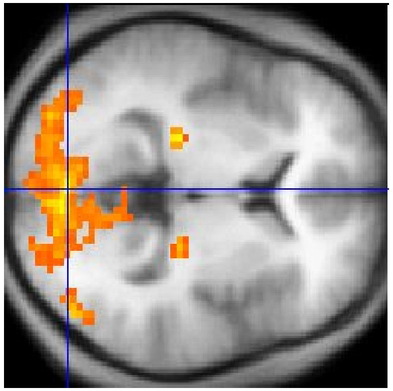


Examples

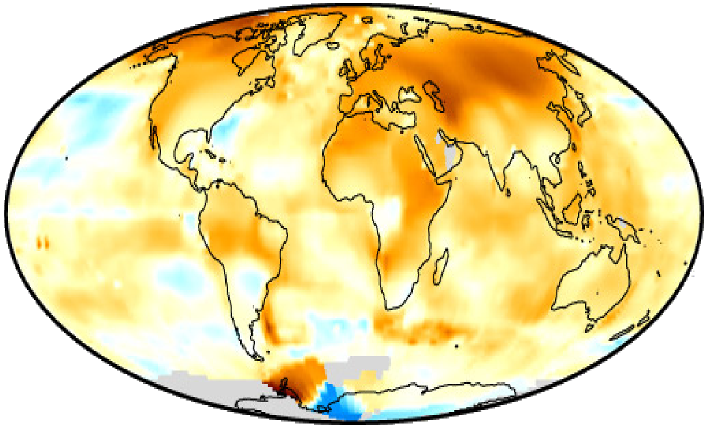
```
createdir /jobs
addattr /jobs jobStatus int
addentry /jobs/job1 jobStatus 0
updateattr /jobs jobStatus 1 jobID>100
selectattr /DLibrary:FileName /DLAudio:Author /DLAudio:Album
        '/DLibrary:FILE=/DLAudio:FILE and like(/DLibrary:FileName, “%.mp3")‘
```

EGEE Experiences With AMGA

Medical Data Management

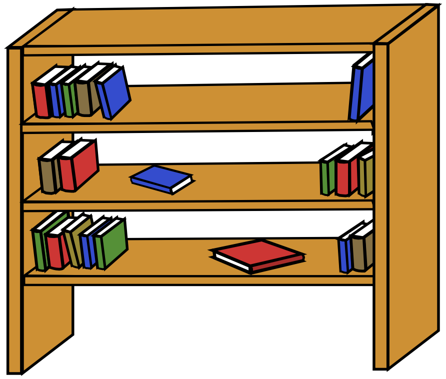


Climate Research



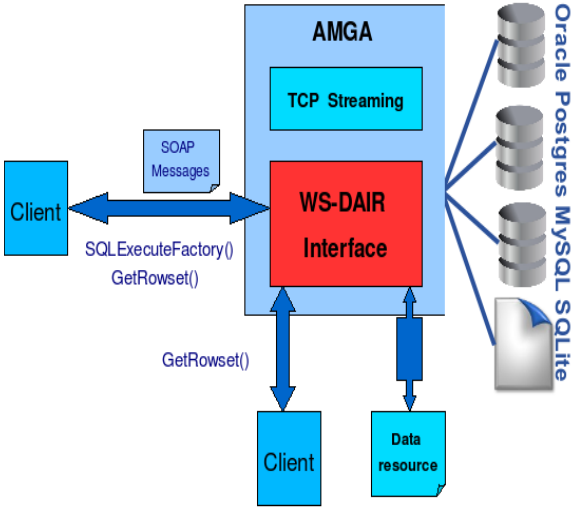
High Energy Physics

Digital Library



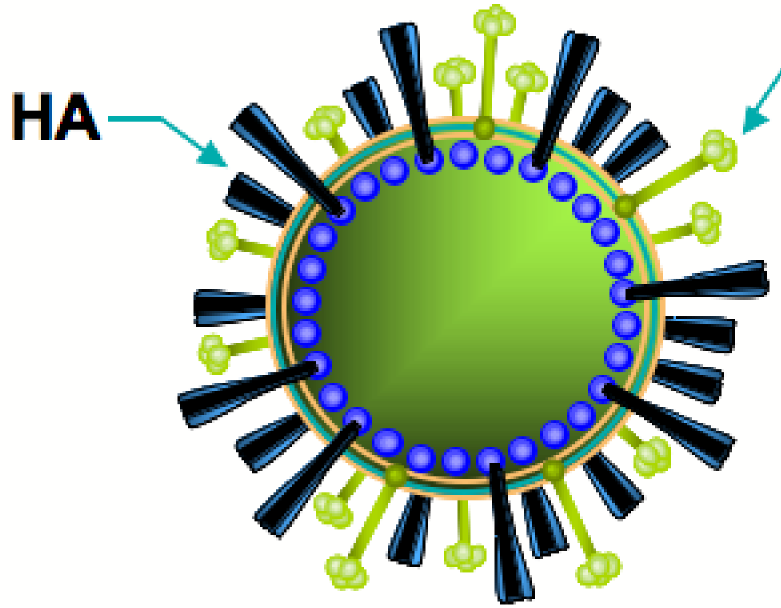
AMGA WS-DAIR Interface

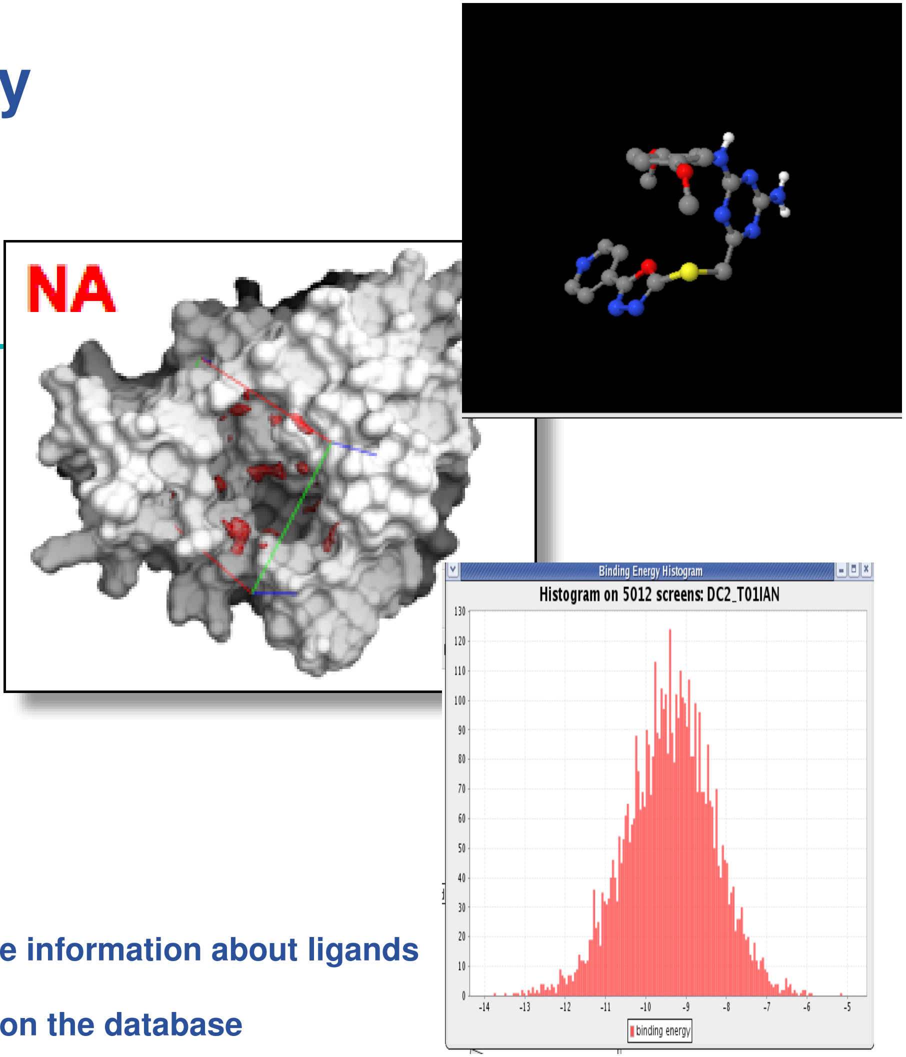
- **Implemented in C++ using gSoap**
- **Features**
  - Integration with database access services adapting WS-DAIR
  - Sending result sets in chunks, using iterations
  - Caching result sets
  - Scrolling in result set, in any direction
  - Third party delivery of results
  - PropertyDocuments, as in-memory data
  - Standard encoding of data (Java WebRowSet)



Evaluation

- **Avian Flue Drug Discovery**





-The application uses AMGA Metadata Catalogue to store information about ligands

-Direct and indirect access patterns were used to query on the database

-Various queries of different size, up to 300,000 records have been tested

Direct Access

- ## Message samples

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv=
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<SQLExecuteRequest xmlns=
"urn:metadata.server.amga.glite.org">
<DataResourceAbstractName>metadata
</DataResourceAbstractName>
<DatasetFormatURI>uri:com.java.xml.ns.jdbc.webrowset
</DatasetFormatURI>
<SQLExpression Language="uri:AMQL">
<Expression> selectattr  /wisdom/results/autodock/ligand_id
 /wisdom/results/autodock/ligands:name 'ligand_id &gt 301275;'
</Expression>
</SQLExpression>
</SQLExecuteRequest>
</soapenv:Body>
</soapenv:Envelope>
```

SQLExecuteRequest

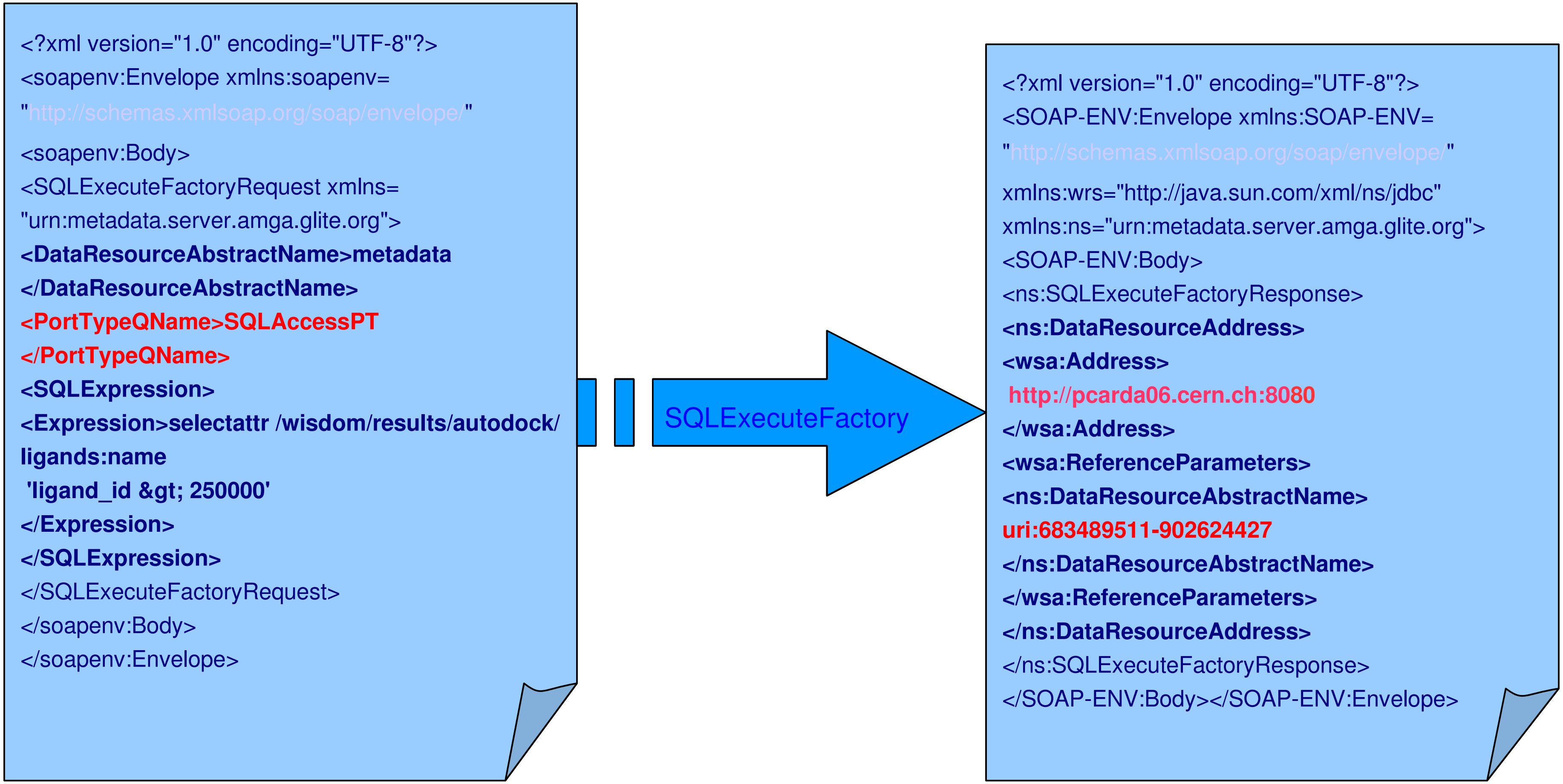
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
 "http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:SOAP-ENC=
 "http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:wrs="http://java.sun.com/xml/ns/jdbc"
 xmlns:ns="urn:metadata.server.amga.glite.org">
<SOAP-ENV:Body>
<ns:SQLExecuteResponse>
<ns:SQLDataset xsi:type=
"ns:SQLDatasetType">
<ns:DatasetFormatURI>
uri:com.sun.java.xml.ns.jdbc.webrowset
</ns:DatasetFormatURI>
DATA
<ns:SQLDataset>
<ns:SQLExecuteResponse>
```

```
<wrs:data>
<wrs:currentRow>
<wrs:columnValue>301117
</wrs:columnValue>
<wrs:columnValue>cdi_G669-02
</wrs:columnValue>
</wrs:currentRow>
<wrs:currentRow>
<wrs:columnValue>301142
</wrs:columnValue>
<wrs:columnValue>cdi_G669-01
</wrs:columnValue>
</wrs:data>
```

SQLExecuteResponse

Indirect Access I

- ## Message samples



```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv=
"http://schemas.xmlsoap.org/soap/envelope/"
<soapenv:Body>
<SQLExecuteFactoryRequest xmlns=
"urn:metadata.server.amga.glite.org">
<DataResourceAbstractName>metadata
</DataResourceAbstractName>
<PortTypeQName>SQLAccessPT
</PortTypeQName>
<SQLExpression>
<Expression>selectattr /wisdom/results/autodock/
ligands:name
 'ligand_id &gt; 250000'
</Expression>
</SQLExpression>
</SQLExecuteFactoryRequest>
</soapenv:Body>
</soapenv:Envelope>
```

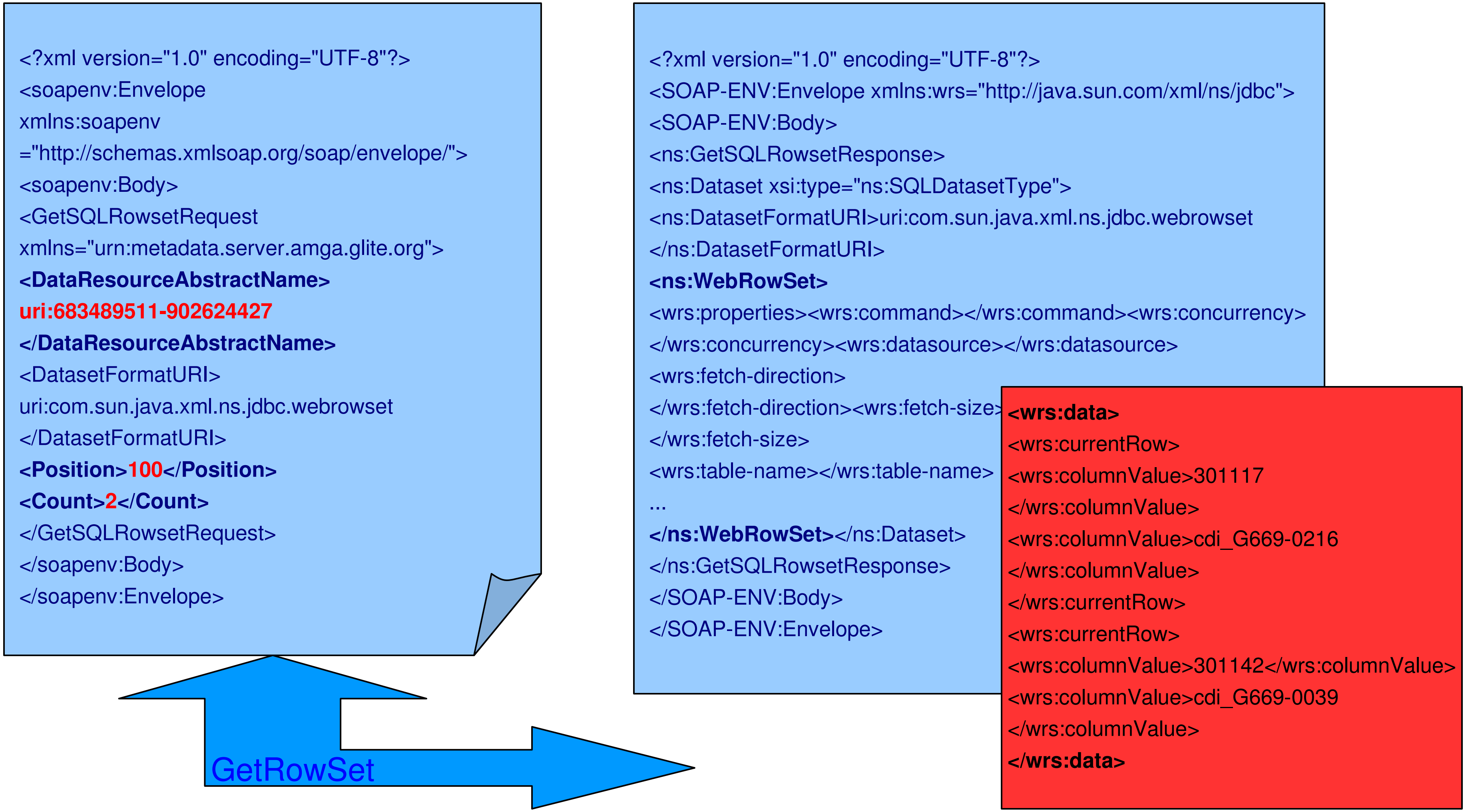
SQLExecuteFactoryRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wrs="http://java.sun.com/xml/ns/jdbc"
xmlns:ns="urn:metadata.server.amga.glite.org">
<SOAP-ENV:Body>
<ns:SQLExecuteFactoryResponse>
<ns:DataResourceAddress>
<wsa:Address>
 http://pcarda06.cern.ch:8080
</wsa:Address>
<wsa:ReferenceParameters>
<ns:DataResourceAbstractName>
uri:683489511-902624427
</ns:DataResourceAbstractName>
</wsa:ReferenceParameters>
</ns:DataResourceAddress>
</ns:SQLExecuteFactoryResponse>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
```

SQLExecuteFactoryResponset

Indirect Access II

• Message samples



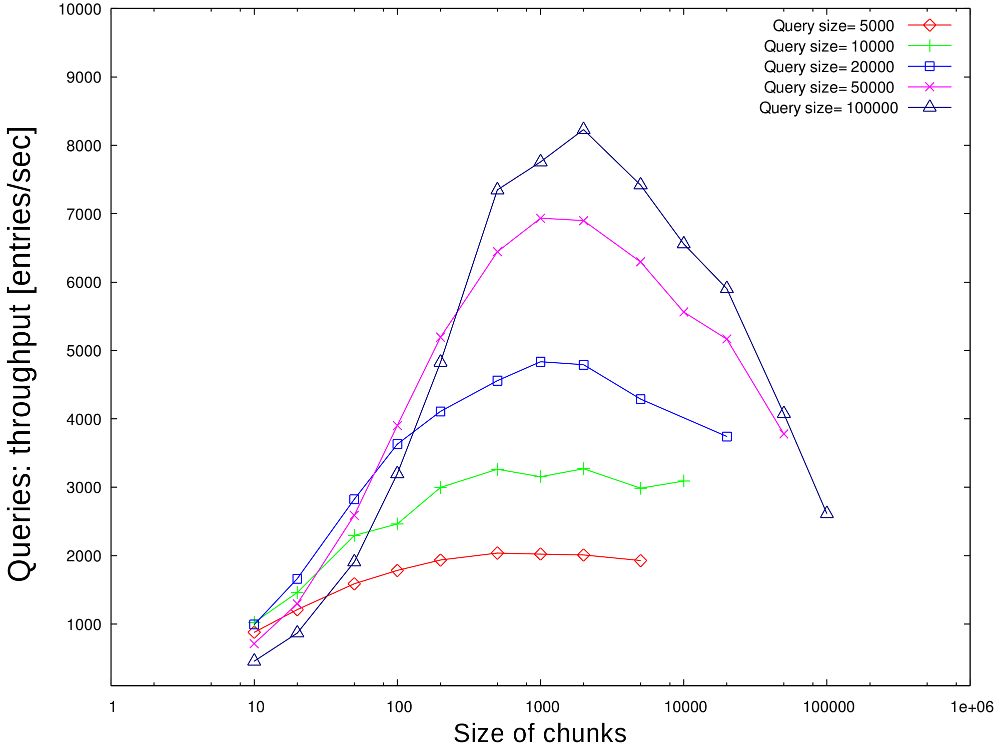
```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv
="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<GetSQLRowsetRequest
xmlns="urn:metadata.server.amga.glite.org">
<DataResourceAbstractName>
uri:683489511-902624427
</DataResourceAbstractName>
<DatasetFormatURI>
uri:com.sun.java.xml.ns.jdbc.webrowset
</DatasetFormatURI>
<Position>100</Position>
<Count>2</Count>
</GetSQLRowsetRequest>
</soapenv:Body>
</soapenv:Envelope>
```

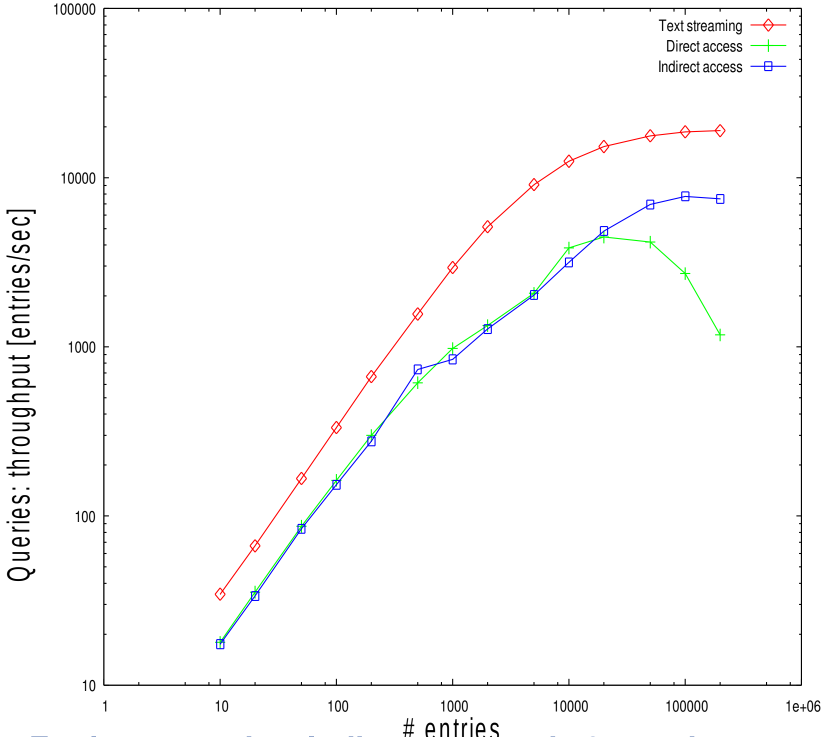
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:wrs="http://java.sun.com/xml/ns/jdbc">
<SOAP-ENV:Body>
<ns:GetSQLRowsetResponse>
<ns:Dataset xsi:type="ns:SQLDatasetType">
<ns:DatasetFormatURI>uri:com.sun.java.xml.ns.jdbc.webrowset
</ns:DatasetFormatURI>
<ns:WebRowSet>
<wrs:properties><wrs:command></wrs:command><wrs:concurrency>
</wrs:concurrency><wrs:datasource></wrs:datasource>
<wrs:fetch-direction>
</wrs:fetch-direction><wrs:fetch-size>
</wrs:fetch-size>
<wrs:table-name></wrs:table-name>
...
</ns:WebRowSet></ns:Dataset>
</ns:GetSQLRowsetResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<wrs:data>
<wrs:currentRow>
<wrs:columnValue>301117
</wrs:columnValue>
<wrs:columnValue>cdi_G669-0216
</wrs:columnValue>
</wrs:currentRow>
<wrs:currentRow>
<wrs:columnValue>301142</wrs:columnValue>
<wrs:columnValue>cdi_G669-0039
</wrs:columnValue>
</wrs:data>
```

Enabling Grids for E-sciencE

Influence Of Chunk Size





For large queries, Indirect access is 3 to 7 times faster than direct access.

TCP streaming is 2 to4 times faster than indirect access and 4 to 20 times faster than direct access.

- Chunk-size of 1000 is sweet-spot for this type of query over large range of query sizes

Comments

- WS-DAIR requires all data to be retrieved out of DB(in order to support third-party delivery) GetSQLRowSet, GetTuples. This is memory intensive and raises consistency issue (not easy to implement)

- Using JavaWebRowset as encoding of data pushes overhead , due to unnecessary tags (Column, Row)

Conclusion

- New WS-DAIR compatible interface for AMGA is another step towards interoperability and extensibility of relational database access on the grid
- The work is still in progress
- AMGA Web Site: http://cern.ch/amga

Future Plans

- **Interoperability tests**
- **Support SQL standard**
- **Adapt other web service standards**
  - WS- Security
  - WS-Transactions

Questions

?

The End

Thank you!