# A WS-DAIR Interface to the AMGA Metadata Catalogue

*B. Koblitz, CERN*

*with A. Boloori, KTH Stockholm*

*OGF 23: Workshop on Databases on the Grid*

*Barcelona, June 4th 2008*

# Overview

- **What is AMGA? Why adopt WS-DAIR?**
- **Overview of WS-DAIR**
- **The WS-DAIR implementation for AMGA**
  - Implementation and Features
  - Evaluation
- **Future plans and comments**
  - Native SQL!
- **Conclusions**

# What are AMGA and WS-DIAR?

- **AMGA is Metadata catalogue of EGEE's gLite 3.1 MW (Metadata is relationally structured data for Grid jobs stored in databases)**
- **WS-DAIR is new OGF standard for access to relational DB's on the Grid**
- **Both, the AMGA service and the WS-DAIR interface address (some) problems of DB access on the Grid:**
  - Authentication (Grid-Proxy certificates, VOMS)?
  - Logging, tracing?
  - Connection pooling?
  - Replication of Data?
  - Performance of WAN access?
  - ... the Grid idea?

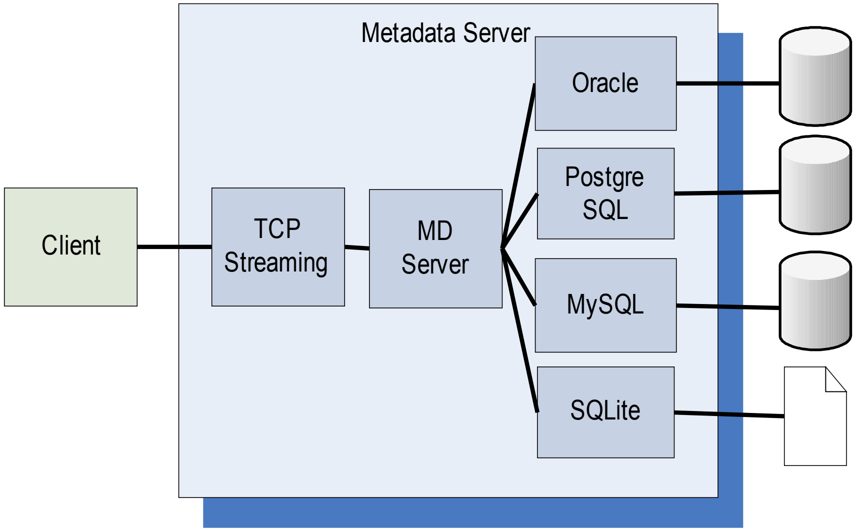
# Objective: WS-DAIR integration

- **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
- **But AMGA uses**
  - Propietary (but well documented) TCP-Streaming protocol (WAN performance)
  - Proprietary SQL inspired query language
    - Hides DB differences
    - Allows access control layer
    - Very simple to use for non-experts

➔ **Integration of WS-DAIR in AMGA will make AMGA a relational data source in a WS-based environment!**

# What can AMGA do?

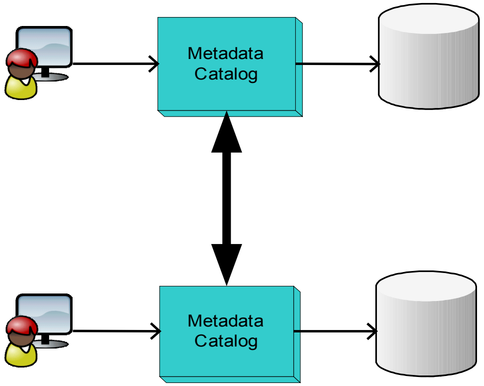
- **AMGA features:**
  - Streamed Bulk Operations
  - Supports single calls, sessions & connections
  - SSL security with grid certs
  - Own User & Group management + VOMS
  - PostgreSQL, Oracle, MySQL, SQLite backends
  - Can access existing DBs
  - Clients in C/C++, Java, Python, Perl, PHP



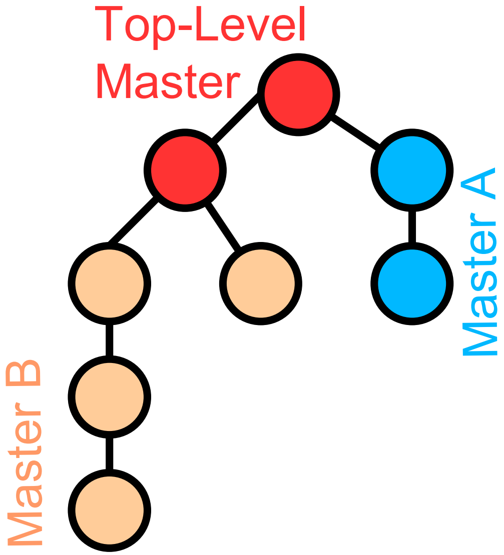
- **Simple Metadata query language:**
  - Supports complex SQL-like queries: joins, SQL functions
  - Abstracts DB data types
  - Checks access permissions per table/row via ACLs

# Replication in AMGA

- **AMGA integrates replication of metadata**
  - Asynchronous replication: Ideal for WAN
  - Master replication
  - DBs are consistent (transactions supported)
  - But: Not all DBs necessarily in same state!



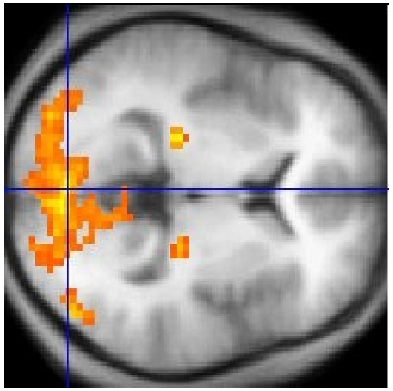
- **Replication makes use of hierarchical table structure**
  - Global table tree
  - Different masters (writer) for sub-trees
  - Only one master per table!



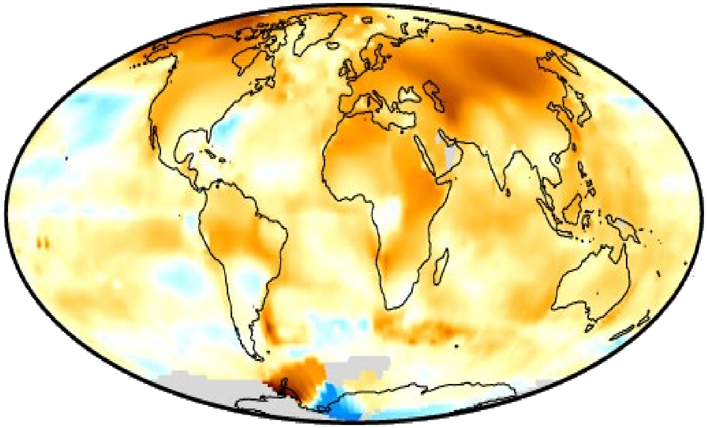
- **Top-level master**
  - controls users/groups
  - hold information about participating Dbs
- **Health-e-Child has dozens of AMGA instances, replicating medical information to/from Hospitals**

# EGEE experience with Metadata

## Medical Data Management

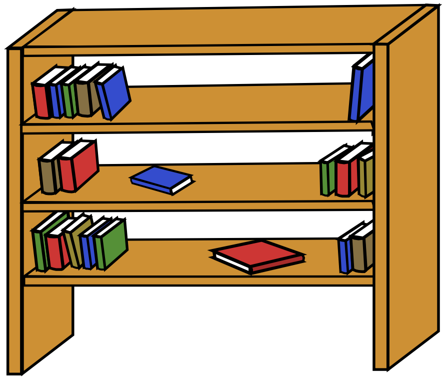


## Climate Research

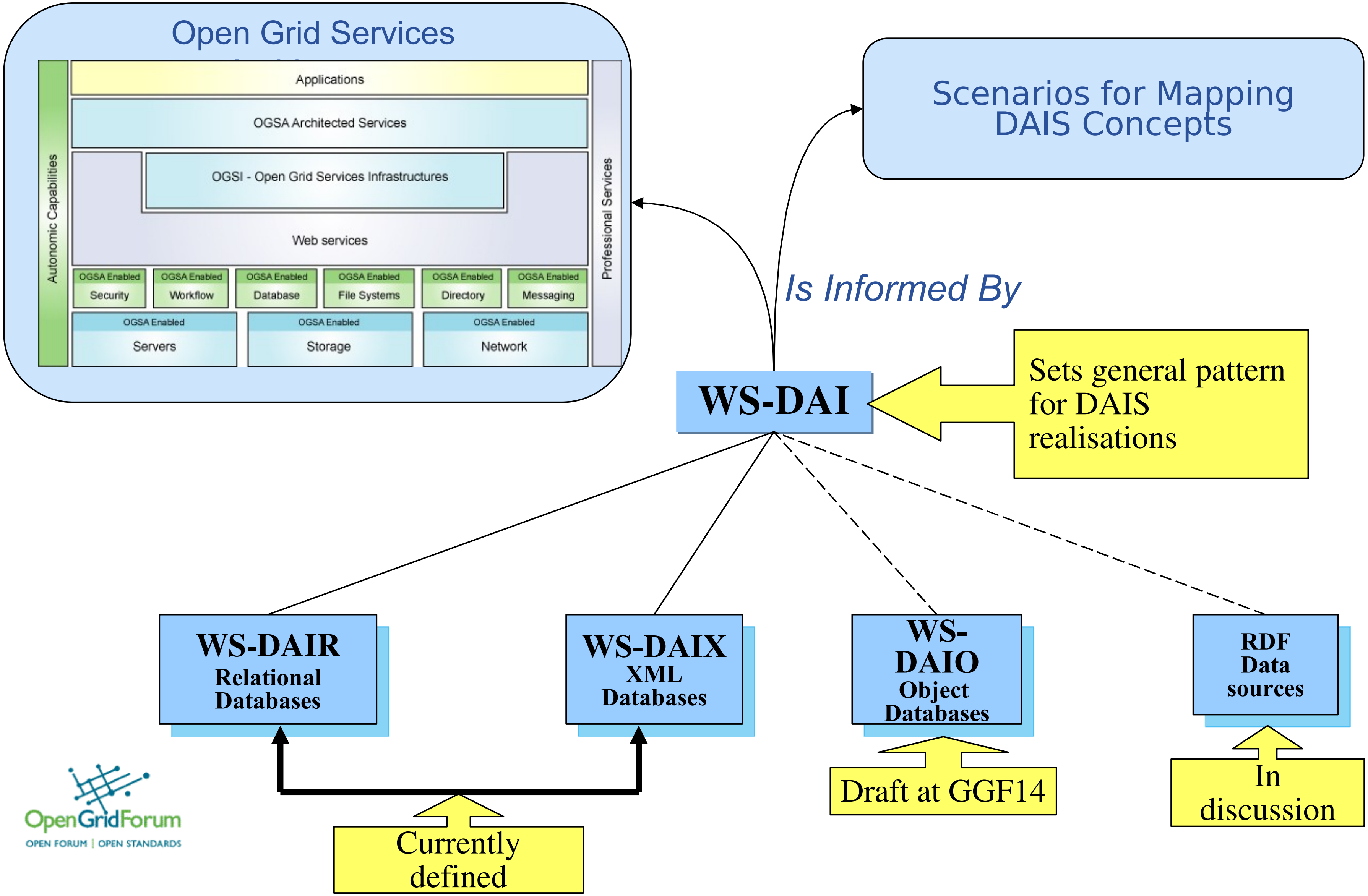


## High Energy Physics

## Digital Library



# WS-DAIR: Overview



Scenarios for Mapping DAIS Concepts

*Is Informed By*

**WS-DAI**

Sets general pattern for DAIS realisations

**WS-DAIR** **Relational Databases**

**WS-DAIX** **XML Databases**

**WS-DAIO** **Object Databases**

**RDF Data sources**

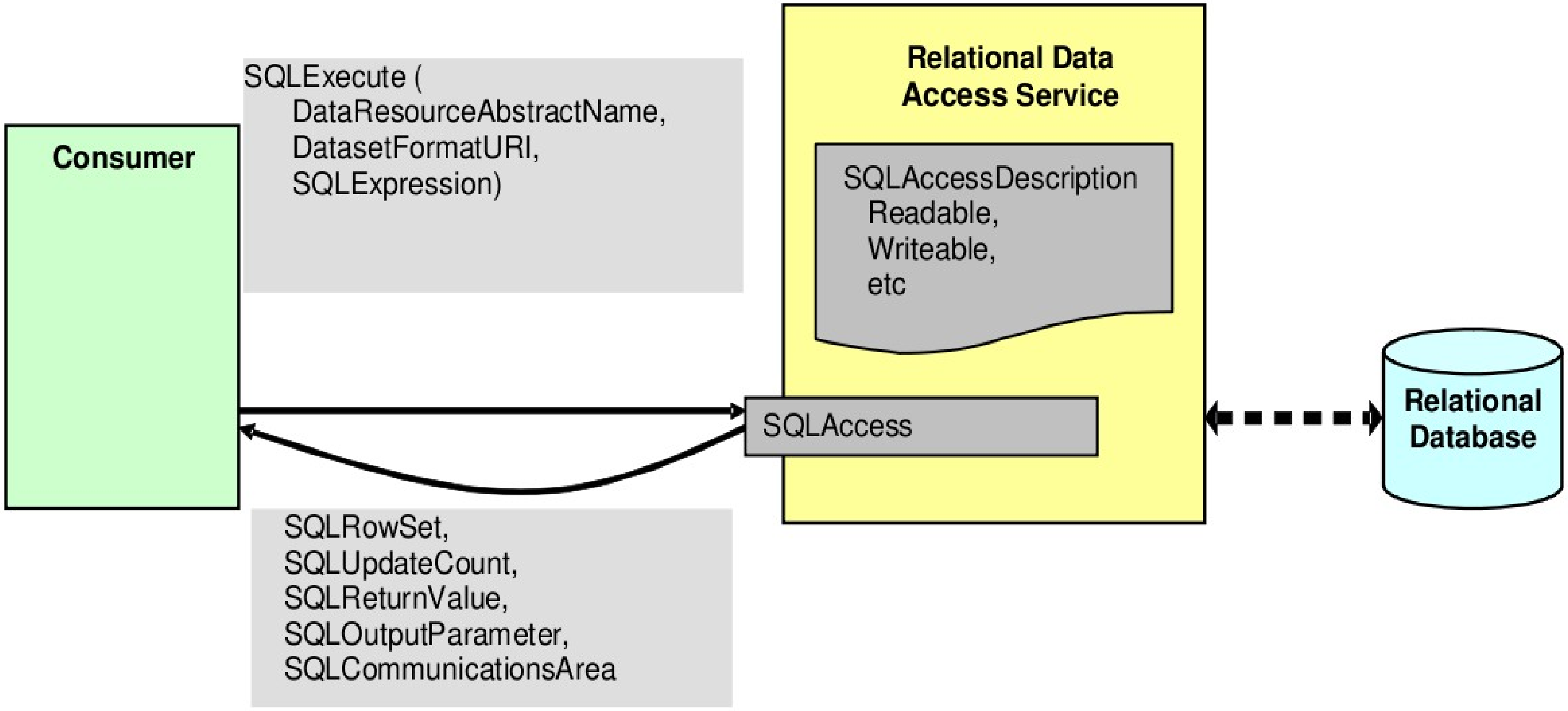
Currently defined

Draft at GGF14

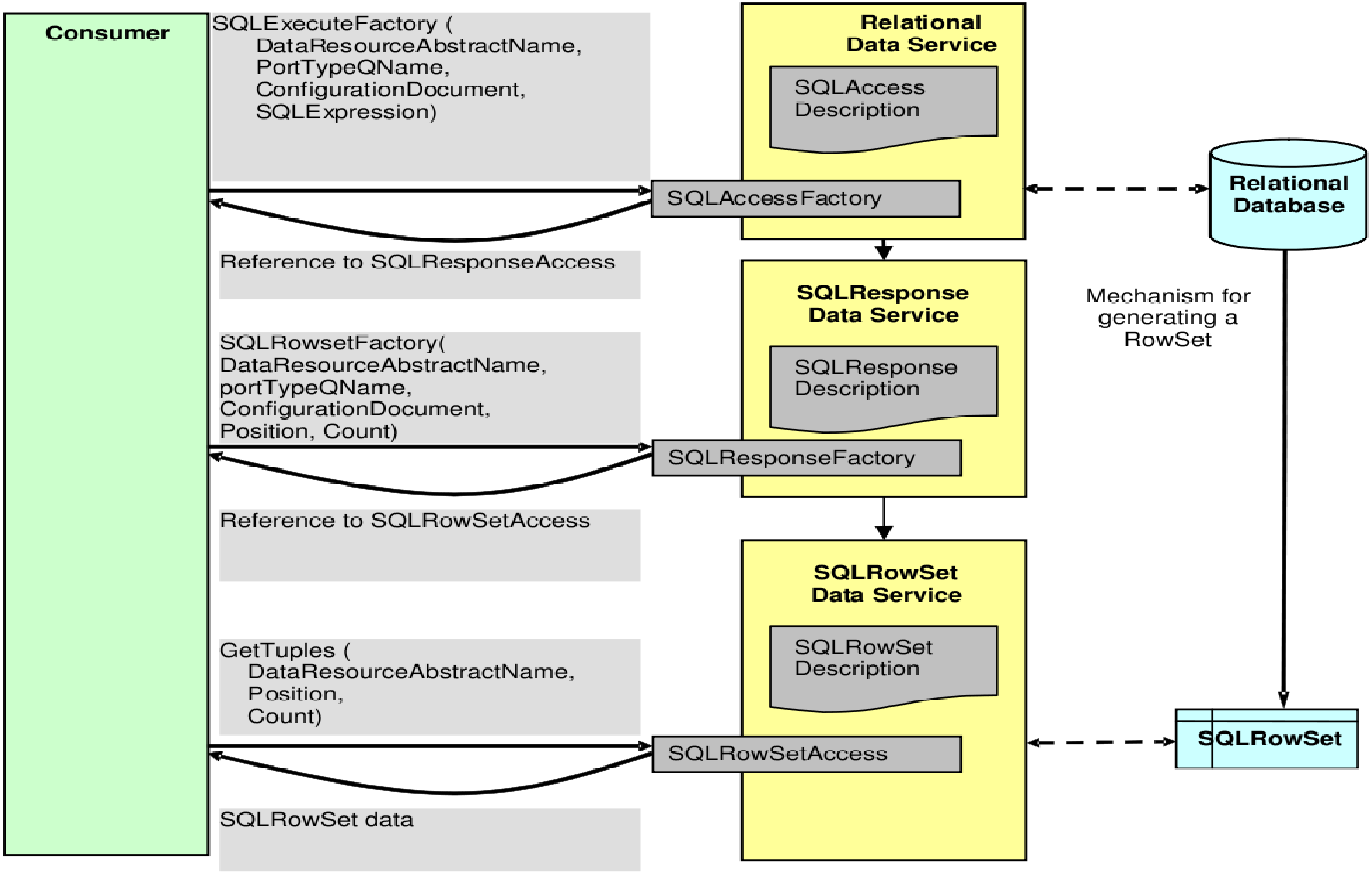
In discussion

# Direct Access

- **Short queries (INSERT, UPDATE, DELETE and brief SELCTS) answered via SQLAccess service:**

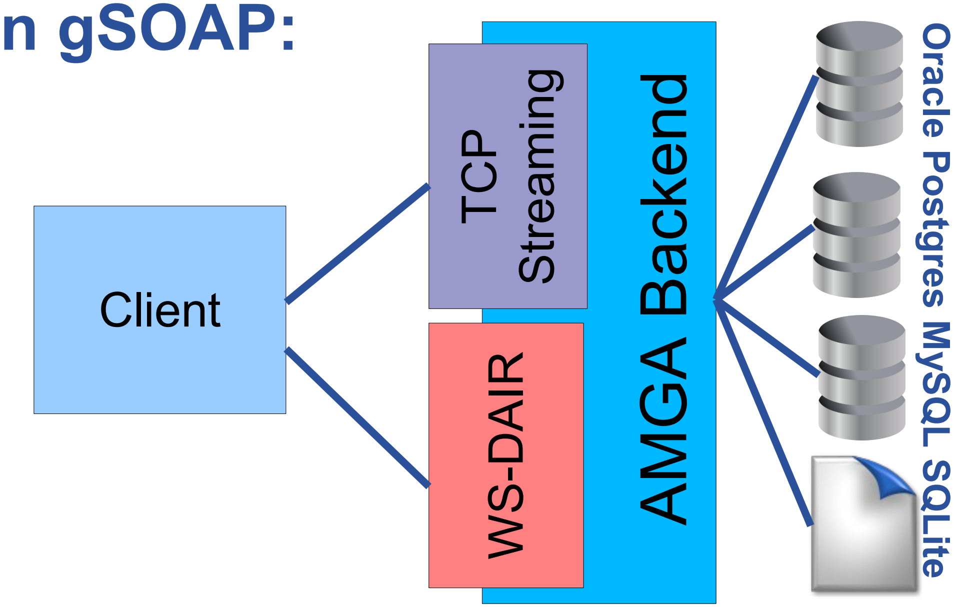


# Indirect Access



# AMGA-DAIR Integration

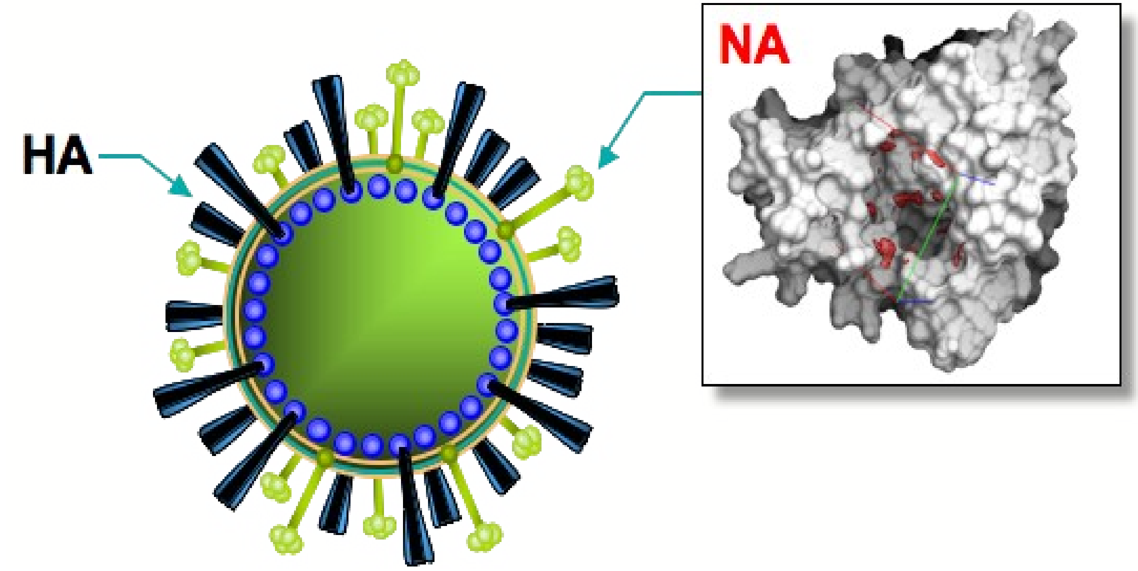
- WS-DAIR interface replaces the existing SOAP interface, written in gSOAP:

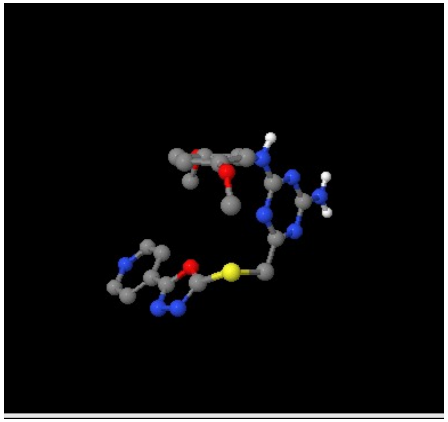


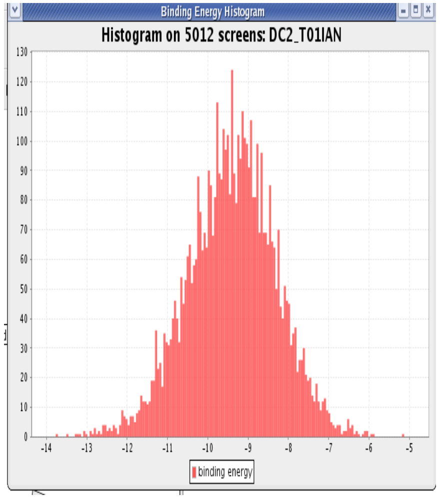
- Implementation of direct access and indirect access done. Data returned in Sun's WebRowSet specification.
- All queries still use AMGA's metadata query language.
- Clients written in Java (Axis) and C++ (gSOAP), WSDL had to be hand-optimized for gSOAP

# Evaluation: Avian Flu Project

- AMGA used as repository for Avian Flu Drug Discovery project



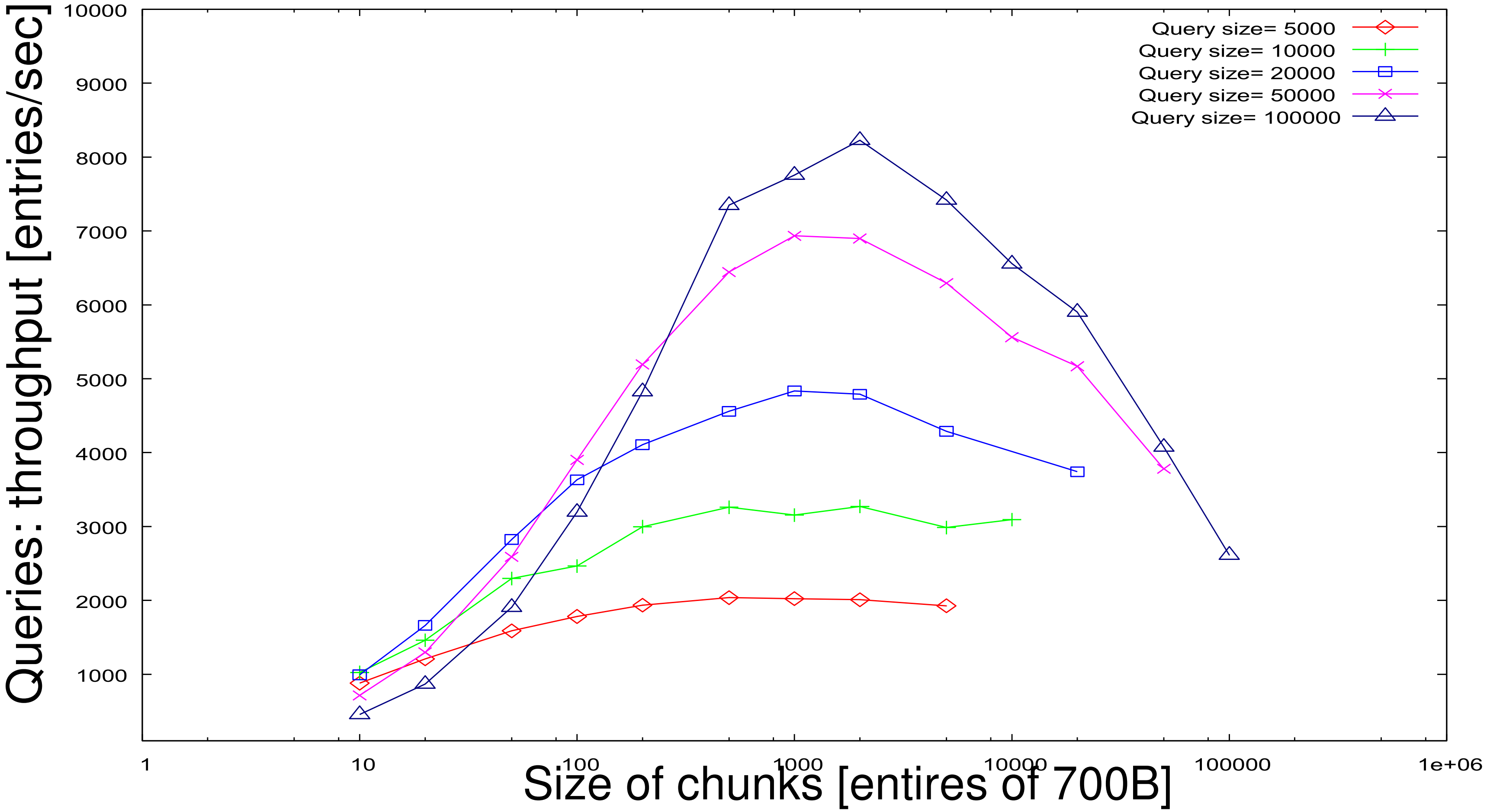




- In silico matching of compounds against NA surface molecule
- WS-DAIR interface tested against real repository with 300k results of docking simulations, using real queries
- All tests done on a LAN

# Indirect Access

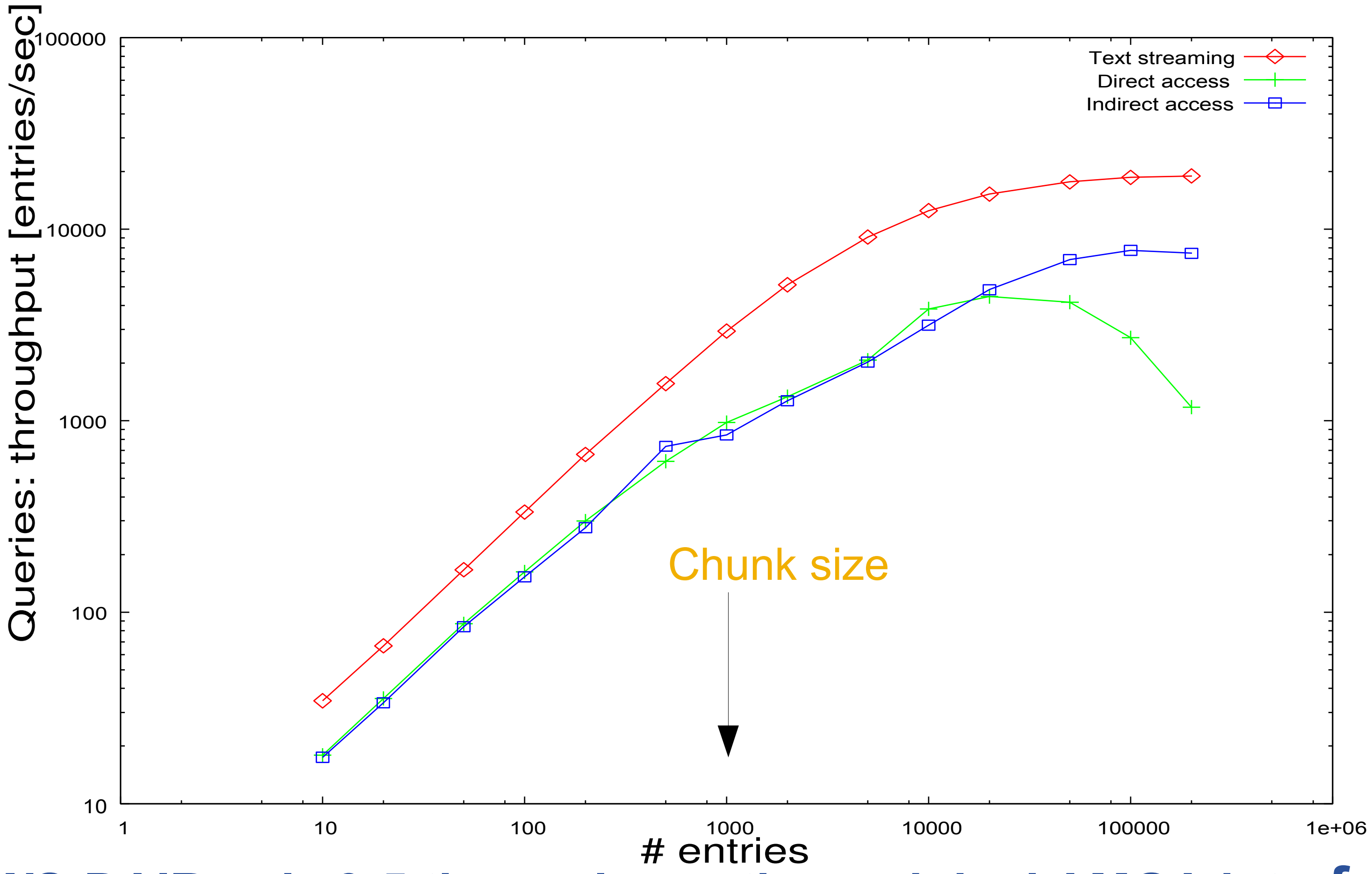
- **Influence of chunk size for indirect access speed:**



- **Chunk size relatively uncritical, large queries favoured**

# Direct Access vs Indirect Access

- **Throughput on LAN, optimized chunk sizes**



**WS-DAIR only 3-5 times slower than original AMGA interface**

# XML Response

- **AMGA'**

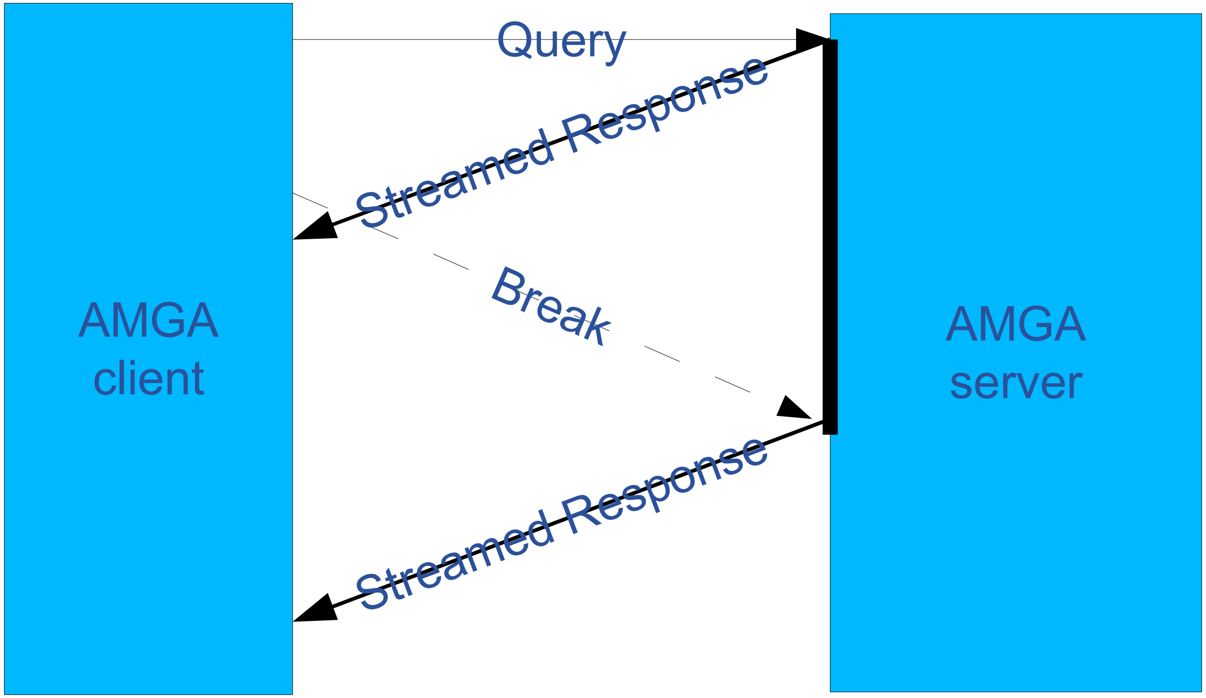
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wrs="http://java.sun.com/xml/ns/jdbc"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

   <ns:WebRowSet>
    DATA
   </ns:WebRowSet>
  </ns:SQLDataset>
 </ns:SQLExecuteResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Payload/message size ration ~1/8 for large messages

```
 <data>
    <currentRow>
      <columnValue> firstrow  </columnValue>
      <columnValue> 1 </columnValue>
     </currentRow>
     <currentRow>
      <columnValue> secondrow </columnValue>
      <columnValue> 2 </columnValue>
     </currentRow>
     <currentRow>
</data>
```

# Streaming Access

- AMGA's original mechanism streams data back with the possibility to interrupt the streaming
- Flow control is provided by the server and clients buffers:



- Only 1 round-trip plus time to stream data.

# Some Questions Raised

- **Had to hand-optimize WSDL for gSOAP**
  - Saw interoperability issues due to bugs
  - Need to do interoperability tests with other implementations
- **Several parameters strongly influence performance**
  - Direct vs indirect access, which to choose
  - Chunk size controlled by the client, how is the right one chosen?
- **Payload to message size ratio is very small < 1/8**
  - What happens when bandwidth limited
  - Does compression help?
- **We will need to investigate carefully the behaviour on the WAN**

# Future Plans

- **Established a new collaboration with Kisti in Korea, new developments are coming:**
  - Support for SQL queries, while keeping AMGA's security features with ACLs for tables/rows
  - Usable through WS-DAIR and text streaming interface
  - Connection pooling DB backend for AMGA
- **Start interoperability tests within DAIS WG**
- **Give feedback to DAIS on standard**
- **New EGEE working group on Grid DB access starting up:**
  - Use AMGA as means to evaluate WS-DAIR
  - Performance analysis

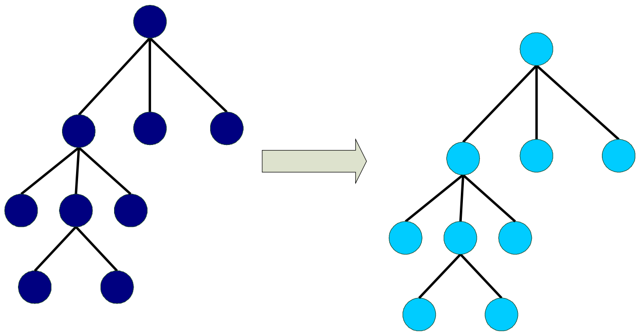
# Summary

- **A WS-DAIR compatible interface makes AMGA a component that can be used within the OGSA FW**
- **Good experiences with implementation of WS-DAIR:**
  - Implementation in AMGA was straight-forward
  - Separation in Direct and Indirect access very useful
- **AMGA can contribute several features to the OGSA infrastructure:**
  - Access control, VO integration
  - Replication
  - Experiences with high-performance applications
- **Still some things missing in AMGA: SQL support**
- **DAIR: http://www.gridforum.org/documents/GFD.76.pdf**
- **AMGA Web Site: http://cern.ch/amga**

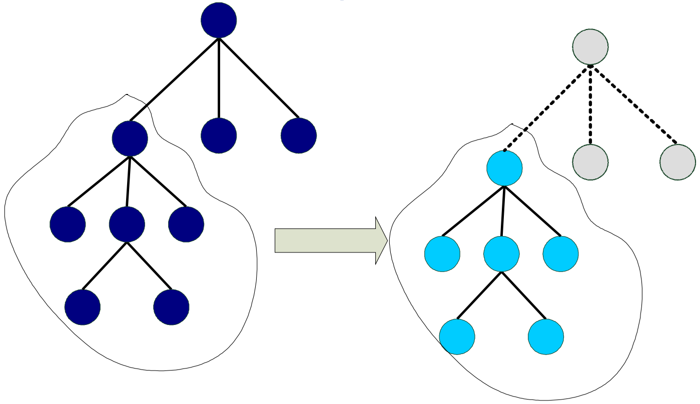
# Replication & Federation Modes

- **AMGA replication makes use of hierarchical concept:**

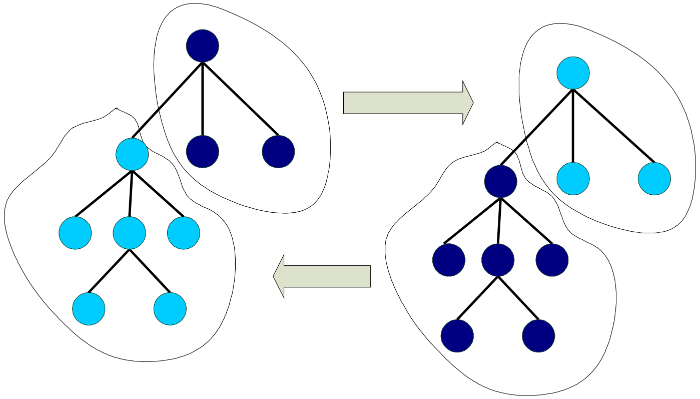
Full replication



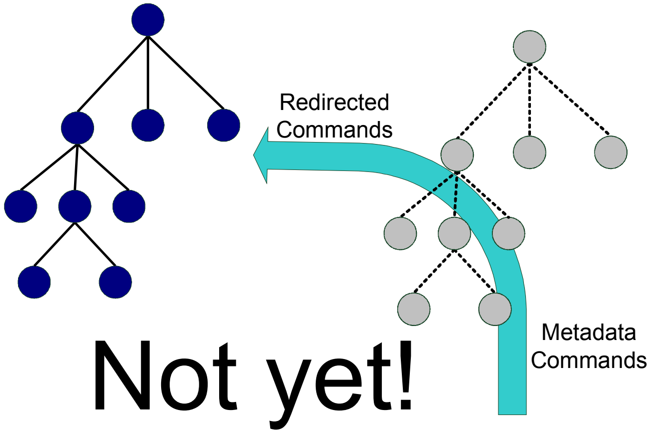
Partial replication



Federation

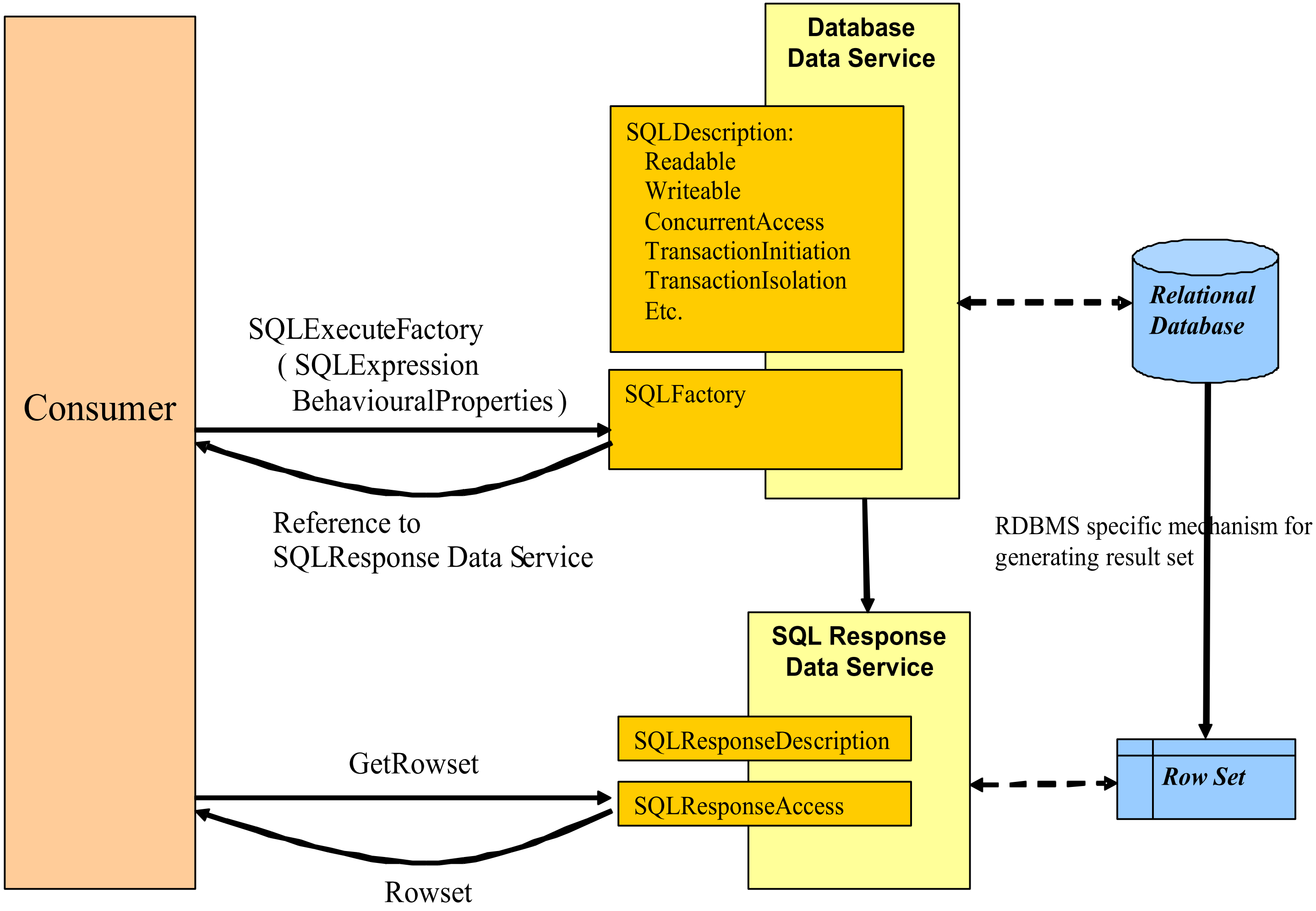


Proxy



Not yet!

# Performance

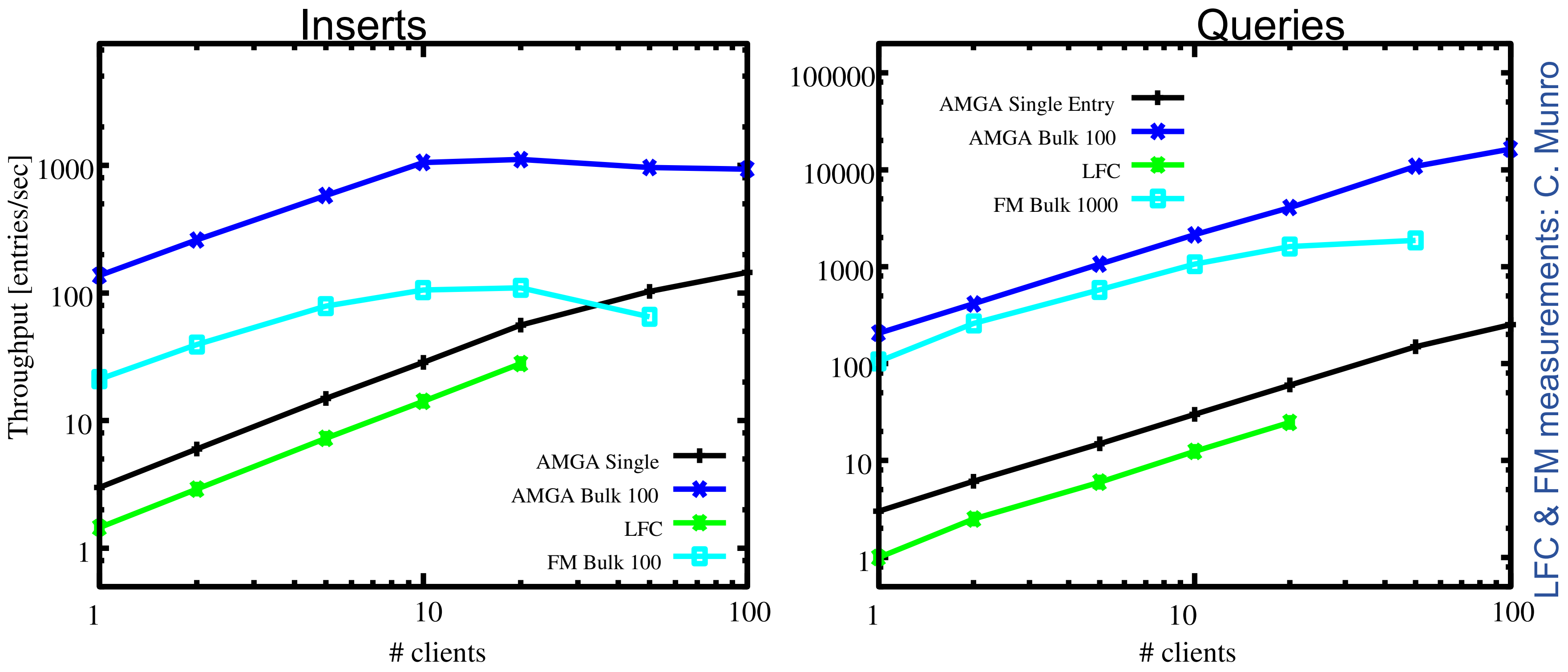


## Indirect Access

# Health-e-Child

- **EU-funded project to connect paediatricians across countries, led by Siemens**
- **Scalable Grid-based tool for sharing images and studies to support decision making**
- **Privacy laws in different countries govern data:**
  - Anonymization
  - Encryption of sensitive data
  - Very strict access restrictions based on roles
  - Limitations on cross-border transfers of data
- **Metadata (names) often more sensitive than real data (images)**
  - Metadata anonymized, private part (names, personal information) striped across USB-Key and a special database
- **AMGA used as a tool to combine metadata gathered in different hospitals via replication**
  - Tightly control replication to restrict data leaving site
  - Replicated metadata allows federated view on complete ensemble

# WAN Performance

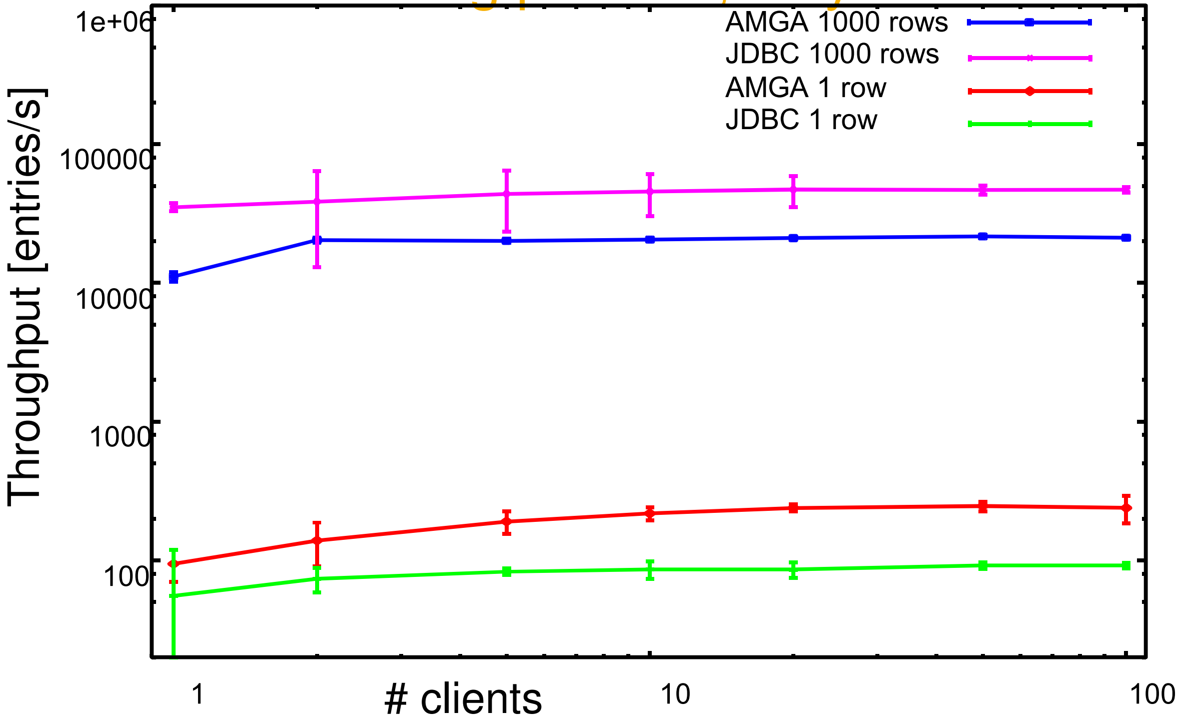


LFC & FM measurements: C. Munro

- **Comparison with FC protocols, connection from Taiwan:**
  - 300ms latency dominates performance
    - Reduce round-trips with sessions or holding connections
  - (Streamed) bulk operations vital for WAN performance

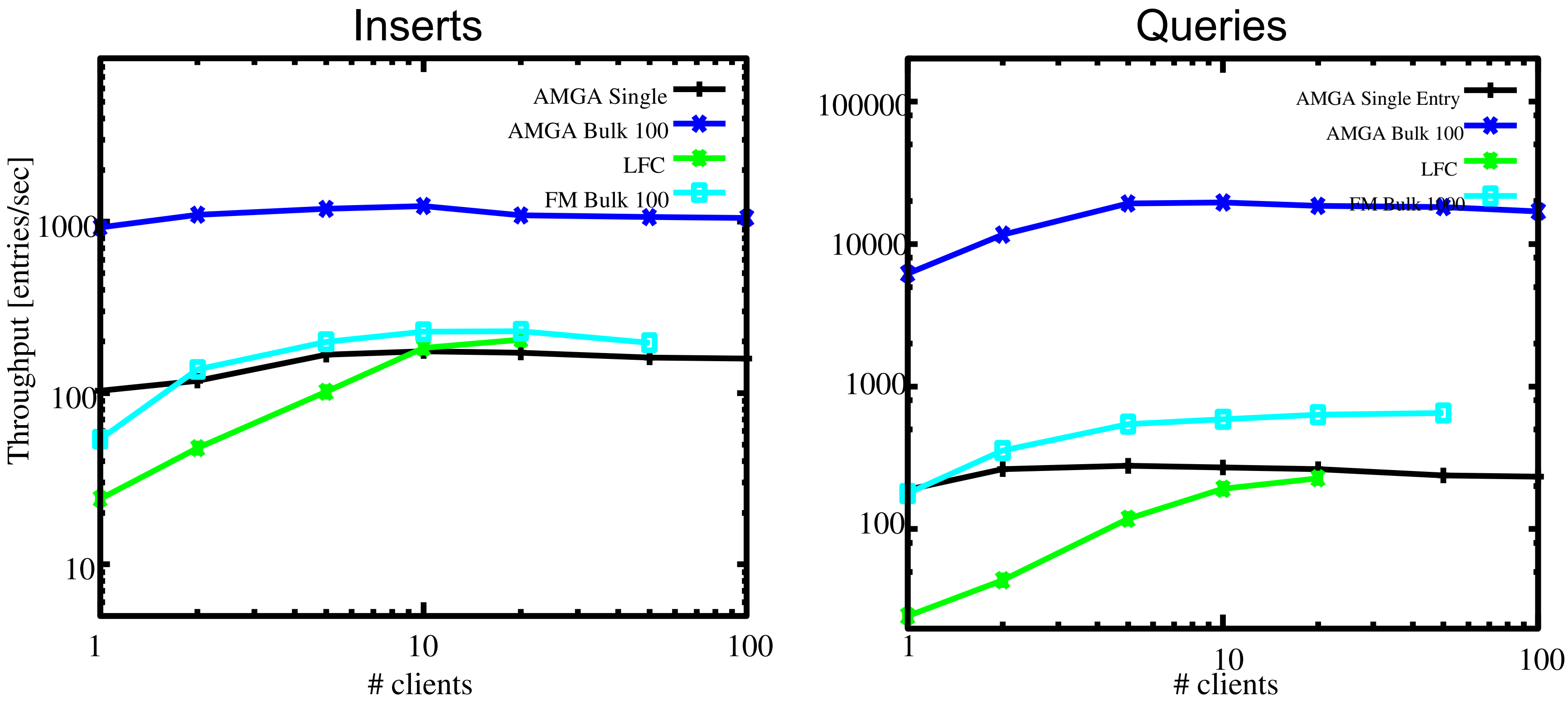
# Performance

- **Performance required to be comparable to direct DB access by HEP applications**
  - Lean C++ Implementation
  - Fast TCP text streaming protocol, very fast SSL sessions



Throughput comparison between AMGA and direct access via JDBC reading same table on a LAN

# LAN Performance



LFC & FM measurements: C. Munro

- **Protocol comparison with LFC and FiReMan catalogues:**
  - Authentication with X509 Certs, SSL connections
  - LFN/GUID pairs inserted, query for GUID of LFN, Oracle DB
  - AMGA scales very well up to 100 concurrent client
  - Streamed bulk inserts/queries are very fast!

Measurements 2005