The gLite AMGA Metadata Catalogue

B. Koblitz, CERN

with S. Ahn, A. J. Boloori, T. Calanducci, C. Cherubino, S. Hwang, N. Kim, S. Scifo

EGEE Conference, Istanbul, September 2008

www.eu-egee.org

Information Society

Overview

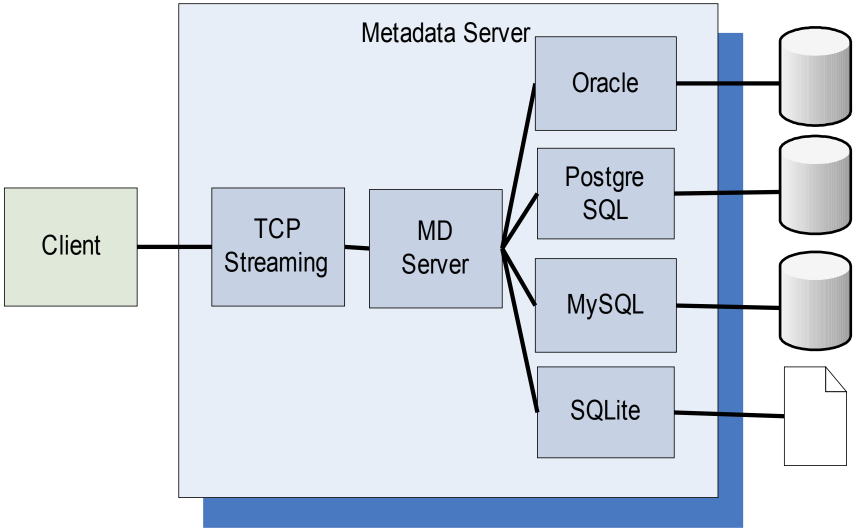
- **Introduction: Metadata on Grids**
- **AMGA: the gLite Metadata catalogue**
- **New developments in AMGA 2.0**
  - WS-DAIR Web-Services Front-end
  - SQL-Queries
  - Multi-Threaded DB backend
- **AMGA use cases**
  - LHCb Logging and Bookkeeping (Very large DB)
  - Medical metadata: Health-e-child (Replication)
  - Task managment: In silico drug design with WISDOM (very many reads / writes)

Metadata in EGEE

- **Metadata is information about**
  - data stored in files
  - Tasks (workflow management)
  - ...
- **Metadata is relationally structured**
- **gLite-AMGA provides metadata catalogue interface**
  - Schema (aka table, think directory)
    Has hierarchical name and list of attributes `/prod/events`
  - Attributes (aka columns)
    Have name and storage type, Interface handles types as strings
  - Entry (aka row)
    Live in a schema, assign values to attributes
  - Query: SELECT ... WHERE ... clause in SQL
- **AMGA 1.9 introduces native SQL support**
  - Intended for usage in APIs

AMGA Implementation

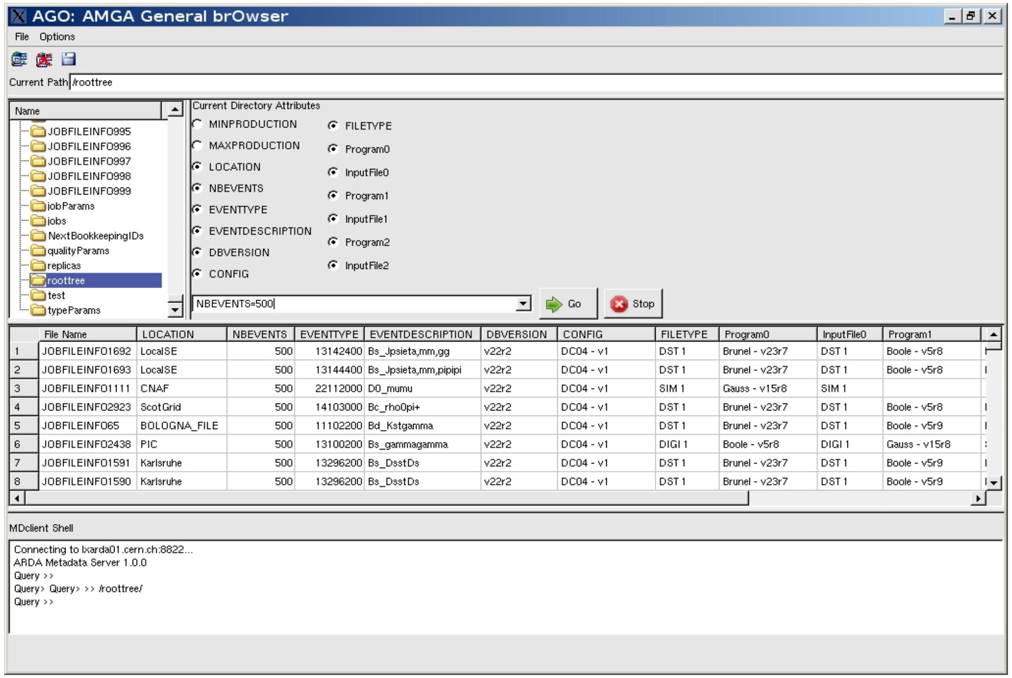
- **AMGA Implementation:**
  - SOAP and Text frontends
  - Streamed Bulk Operations
  - Supports single calls, sessions & connections
  - SSL security with grid certs, support for VOMS
  - Own User & Group management + VOMS
  - PostgreSQL, Oracle, MySQL, SQLite backends
  - Access existing DBs



- **All queries are parsed by AMGA:**
  - AMGA understands security aspects (access permissions)
  - Queries are translated into the respective DB SQL dialect
  - Abstracts DB data types

AMGA Clients & APIs

- **AMGA Clients (for setup, administration)**
  - Shell-like client
  - Graphical Browser (Python)



Current Path /roottree

Current Directory Attributes

○ MINPRODUCTION
○ MAXPRODUCTION
● LOCATION
● NBEVENTS
● EVENTTYPE
● EVENTDESCRIPTION
● DBVERSION
● CONFIG
● FILETYPE
● Program0
● InputFile0
● Program1
● InputFile1
● Program2
● InputFile2

NBEVENTS=500

	File Name	LOCATION	NBEVENTS	EVENTTYPE	EVENTDESCRIPTION	DBVERSION	CONFIG	FILETYPE	Program0	InputFile0	Program1
1	JOBFILEINFO1692	LocalSE	500	13142400	Bs_Jpsieta,mm,gg	v22r2	DC04 - v1	DST 1	Brunel - v23r7	DST 1	Boole - v5r8
2	JOBFILEINFO1693	LocalSE	500	13144400	Bs_Jpsieta,mm,pipipi	v22r2	DC04 - v1	DST 1	Brunel - v23r7	DST 1	Boole - v5r8
3	JOBFILEINFO1111	CNAF	500	22112000	D0_mumu	v22r2	DC04 - v1	SIM 1	Gauss - v15r8	SIM 1	
4	JOBFILEINFO2923	ScotGrid	500	14103000	Bc_rho0pi+	v22r2	DC04 - v1	DST 1	Brunel - v23r7	DST 1	Boole - v5r8
5	JOBFILEINFO65	BOLOGNA_FILE	500	11102200	Bd_Kstgamma	v22r2	DC04 - v1	DST 1	Brunel - v23r7	DST 1	Boole - v5r9
6	JOBFILEINFO2438	PIC	500	13100200	Bs_gammagamma	v22r2	DC04 - v1	DIGI 1	Boole - v5r8	DIGI 1	Gauss - v15r8
7	JOBFILEINFO1591	Karlsruhe	500	13296200	Bs_DsstDs	v22r2	DC04 - v1	DST 1	Brunel - v23r7	DST 1	Boole - v5r9
8	JOBFILEINFO1590	Karlsruhe	500	13296200	Bs_DsstDs	v22r2	DC04 - v1	DST 1	Brunel - v23r7	DST 1	Boole - v5r9

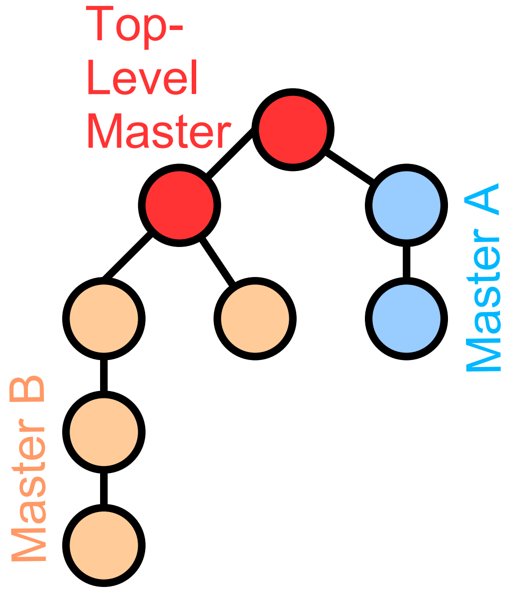
MDclient Shell

```
Connecting to lxarda01.cern.ch:8822...
ARDA Metadata Server 1.0.0
Query >>
Query> Query> >> /roottree/
Query >>
```

- **Many Programming APIs**
  - Diverse user community requested/provided C/C++, Java, Python, Perl, PHP
- **SOAP interface**
  - WS-DAIR compatible, tested with gSOAP and Axis toolkits

Replication in AMGA

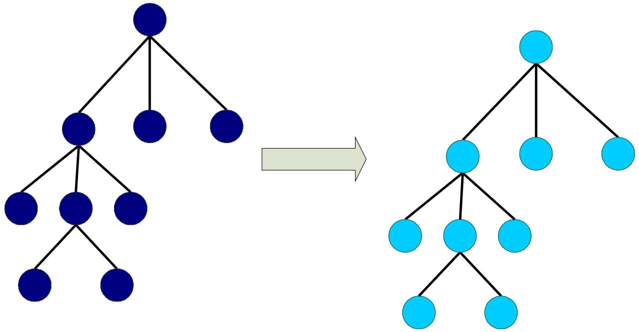
- **AMGA integrates replication of metadata**
  - Asynchronous replication: Ideal for WAN
  - DBs are consistent (transactions supported)
  - However: Not all DBs necessarily in same state
- **Replication makes use of hierarchical table structure**
  - Global table tree
  - Different masters for sub-trees
  - Only one master per table!
  - Writes only allowed on master.
- **Top-level master**
  - controls users/groups
  - hold information about participating DBs



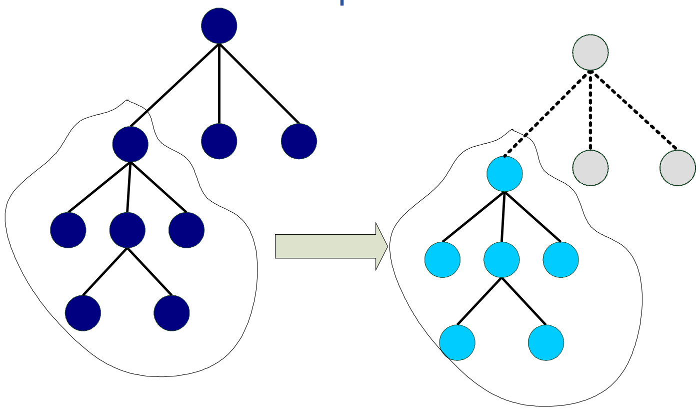
Replication & Federation Modes

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

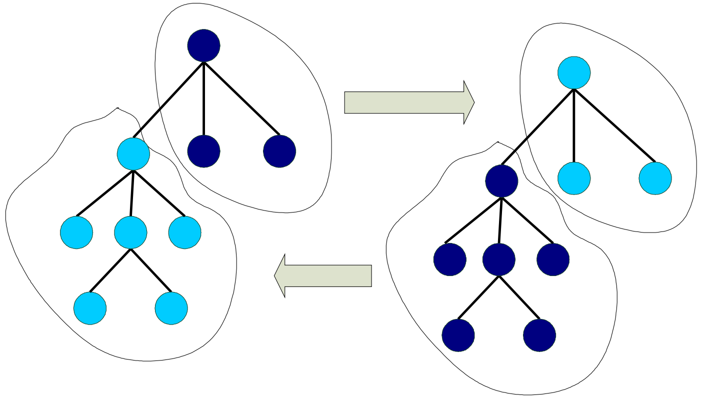
Full replication



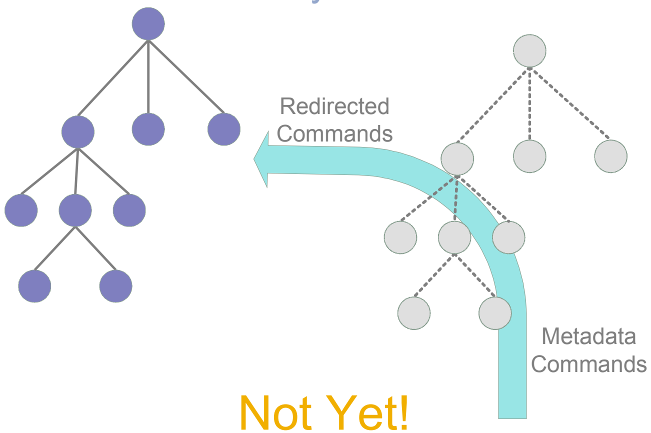
Partial replication



Federation



Proxy

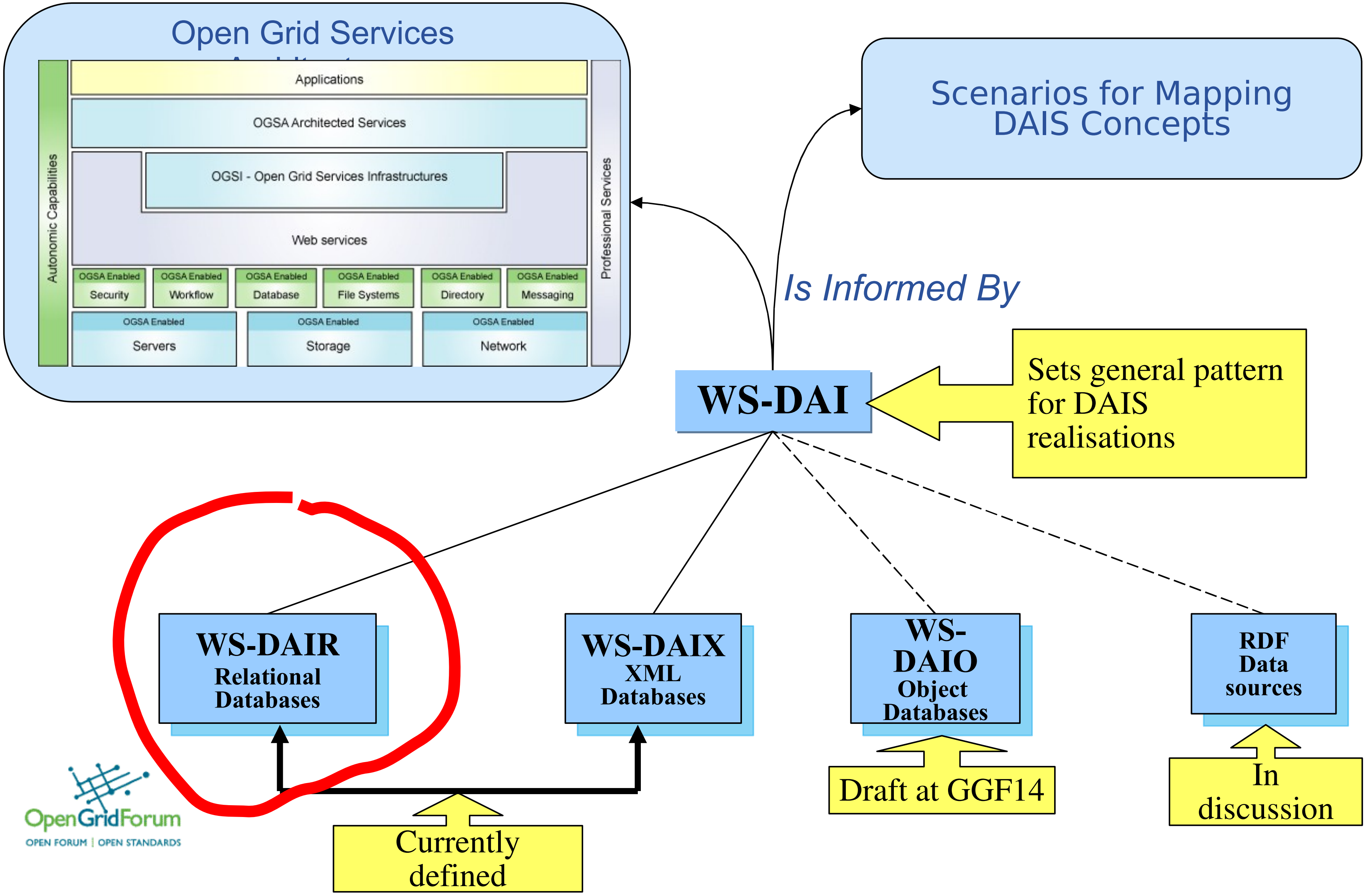


Not Yet!

AMGA 2.0

- **Currently preparing AMGA 2.0 release**
  - To be released later this year
  - Stability of WS-DAIR server needs to be improved before releasea
- **Feature-complete 1.9 technology preview available**
  - More flexible DB schema
  - Support for the import of existing relational tables
  - WS-DAIR frontend
  - Native SQL support
  - Multi-threaded DB backend with connection pooling
-

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

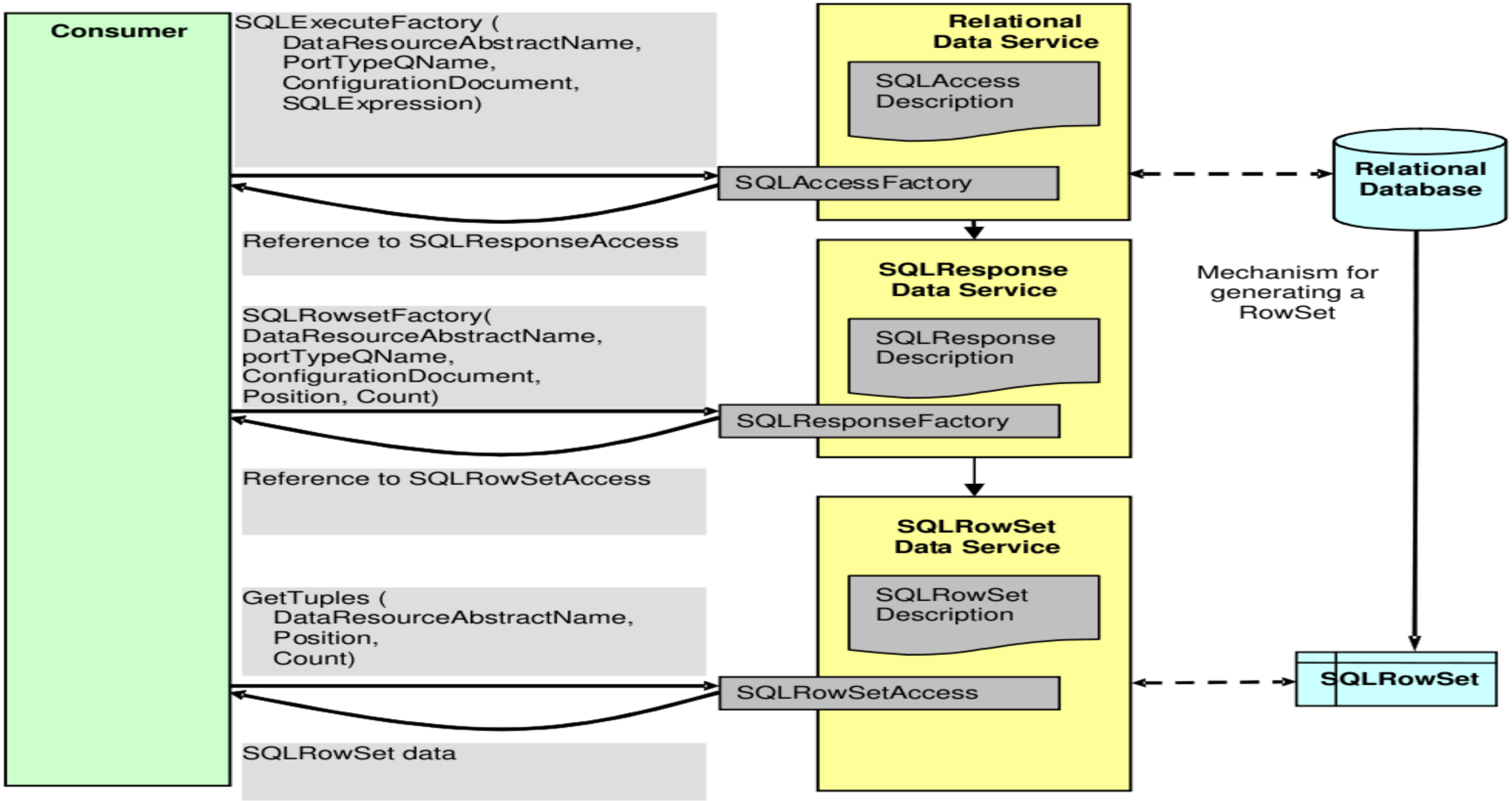
Currently defined

Draft at GGF14

In discussion

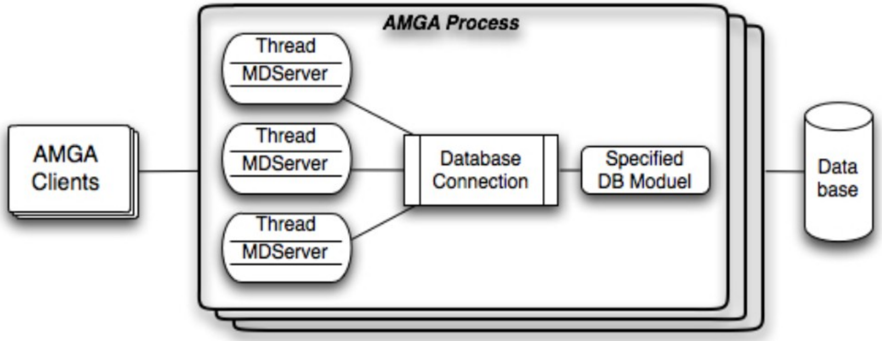
Indirect Access

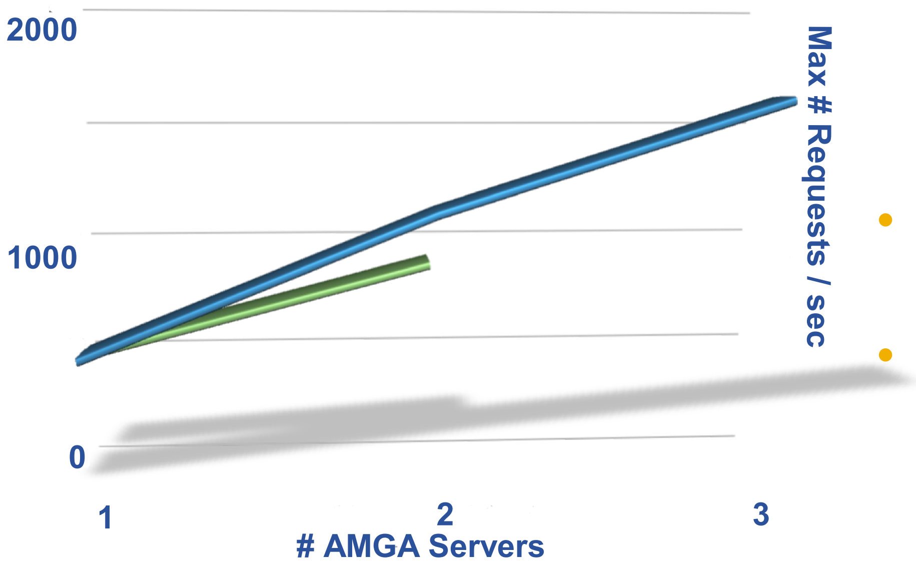
- **Short queries (INSERT, UPDATE, DELETE and brief SELCTS) are answered via the SQLAccess**
- **Large results via indirect access:**



New Multithreaded Backend

- AMGA 1.3 used one process per client connection
- Processes allow control of misbehaving clients
- But processes cannot share DB connections
  - Limits # concurrent clients





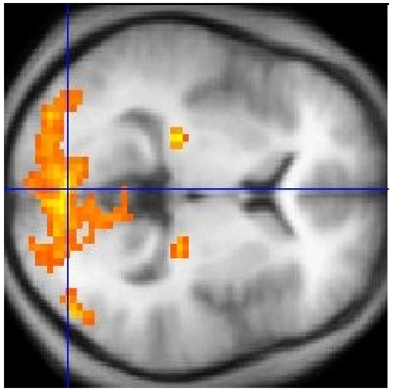
- AMGA 1.9: Multiple threads sharing a DB connection encapsulated in multiple processes
- Threads allow better resource usage on DB
- WISDOM measured 1500 Queries/s, more than the required 5million Queries / h peak needed for data challanges

SQL Queries

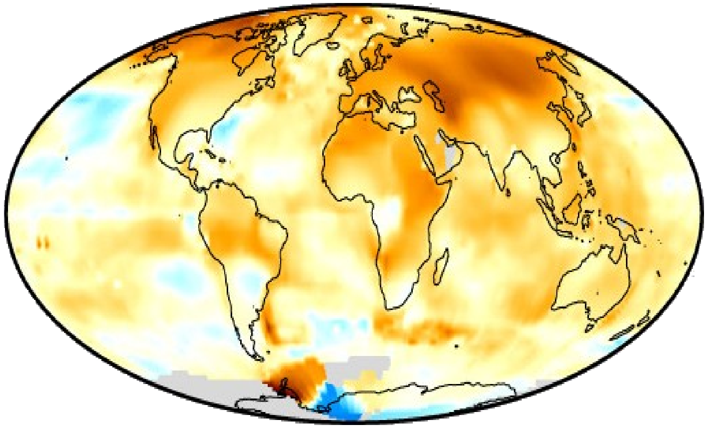
- **AMGA allows queries now in native SQL**
  - Support for entry level SQL 92 done
  - Some 92 intermediate level supported
  - SELECT, UPDATE, INSERT, DELETE
- **All queries are subject to AMGA access restrictions**
  - All queries are parsed by AMGA,
    AMGA "understands" security implications
  - Posix ACLs for tables/entries
- **SQL 92 query is translated into backend DB dialect**

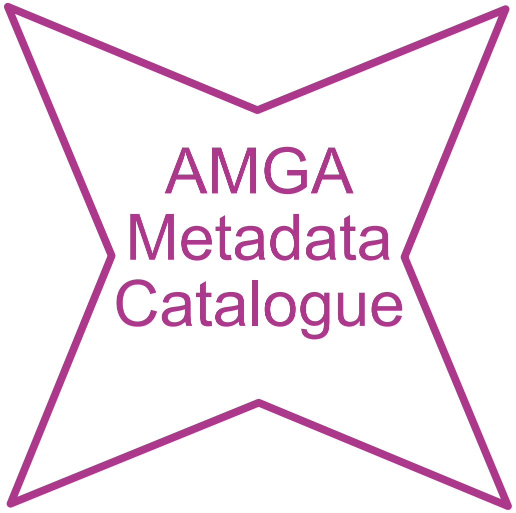
EGEE experience with Metadata

Medical Data Management



Climate Research

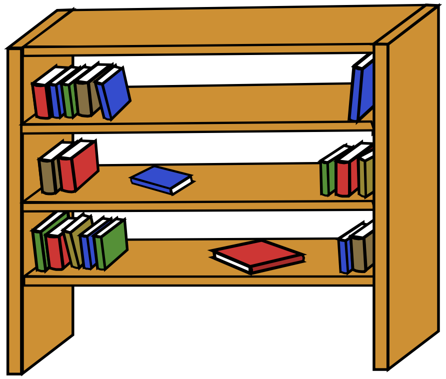




AMGA
Metadata
Catalogue

High Energy
Physics

Digital Library



HEP: LHCb L & B

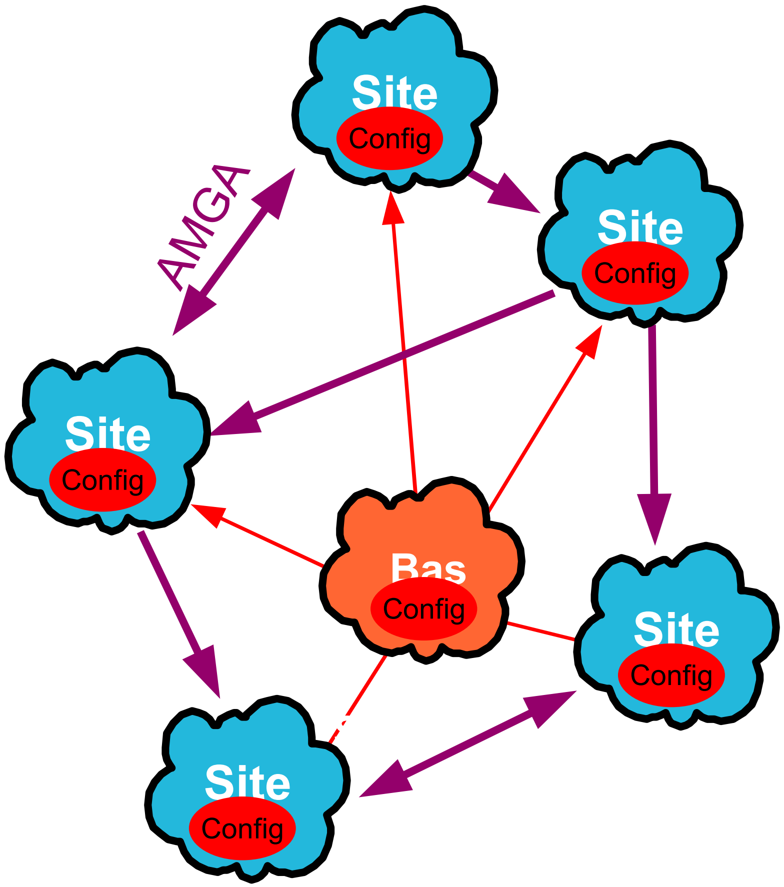
- **LHCb uses AMGA to centrally store the entire file provenance information from jobs processing the data**
  - 100 Million entries required (successfully tested!)
    150GB data
  - 100 000 entries/day insert rate expected
  - 10 entries/second read-rate
- **Main challenges are reliablity, performance and size**
  - Use ORACLE RAC server as backend
  - Production software access via Java (JSP)
  - User (read) access: Python (inc. browser)

Replication in Health-e-Child

- EU-funded project to allow practitioners to share medical data
- Several dozens of hospitals providing case-data
- Central server with credentials for participatingsites and users (replication mandatory)
- Data replicated from site to site on demand
- New sites need to be registered in base AMGA server
- 'Automount' mechanism for joining sites

```
/configuration
/siteA/patients
/siteB/patients
...
/common/patients
```

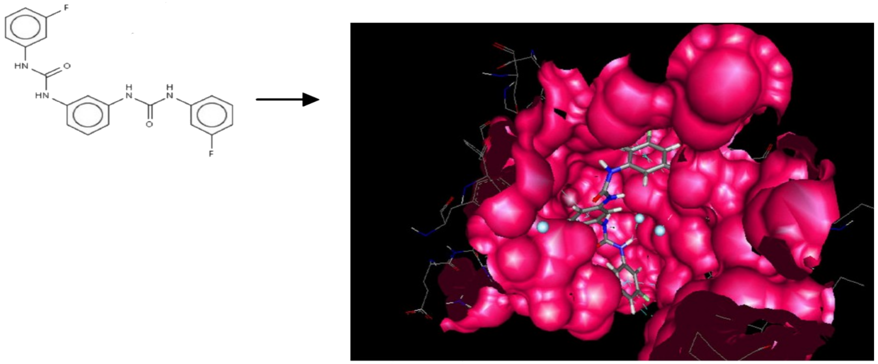
Union View



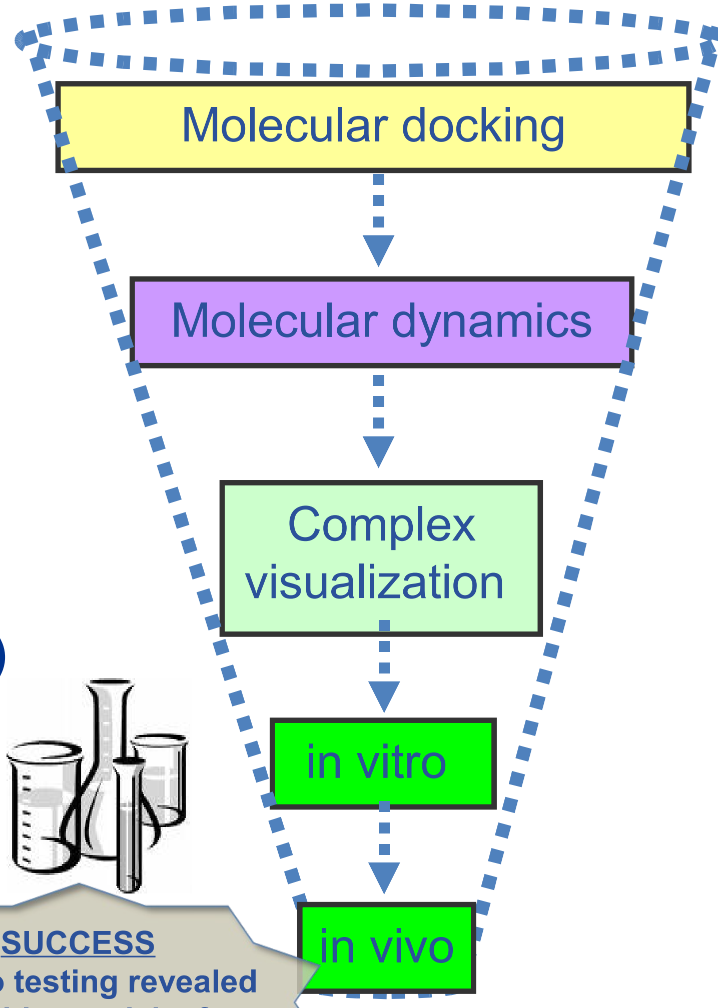
Visit Demo Stand!

Wisdom: Finding Malaria Drugs

- AMGA used as result repository and workflow manager to find Malaria drugs
- In silico matching of compounds against NA surface molecule



- Second Data Challenges on malaria (2006) 140 million dockings in 10 weeks
- Average throughput 80,000 dockings/h = 800k AMGA requests/h
- Threaded AMGA developed for peak rates (5Mio queries / s)



SUCCESS
In vitro testing revealed inhibition activity for compounds selected

Summary

- **AMGA provides Grid Layer to relational databases:**
  - Abstraction of different DB vendors
  - Efficient LAN/WAN access
  - Fast X509 Grid security, VOMS integration
  - Rich set of features: Transactions, Views, Sequences, complex Joins....
- **AMGA is building block for distributed databases:**
  - Asynchronous replication
- **AMGA widely used in EGEE community and beyond**
- **AMGA 2.0 brings new features:**
  - Native SQL support
  - Very scalable backend
  - WS-DAIR compatible frontend

Basic Concepts

- **Schema (aka table, think directory)**
  - Has hierarchical name and list of attributes `/prod/events`
- **Attributes (aka columns)**
  - Have name and storage type, Interface handles types as strings
- **Entry (aka row)**
  - Live in a schema, assign values to attributes
- **Query**
  - SELECT ... WHERE ... clause in SQL-like query language

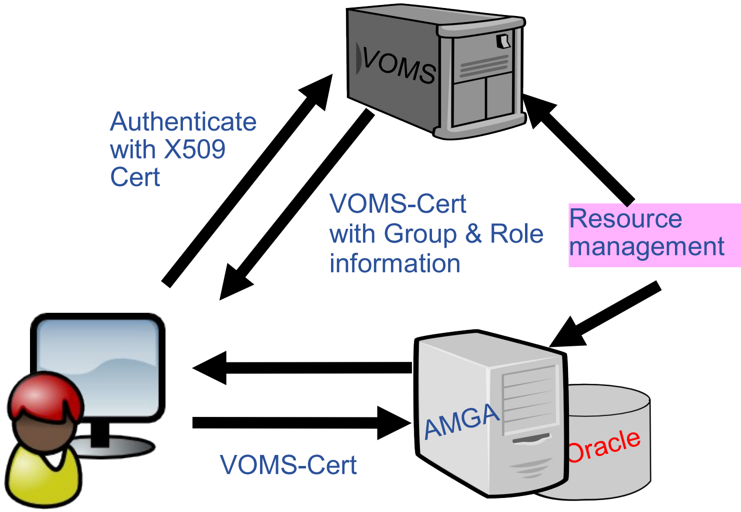
Examples

```
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")‘
```

- **AMGA 1.9 supports native SQL!**

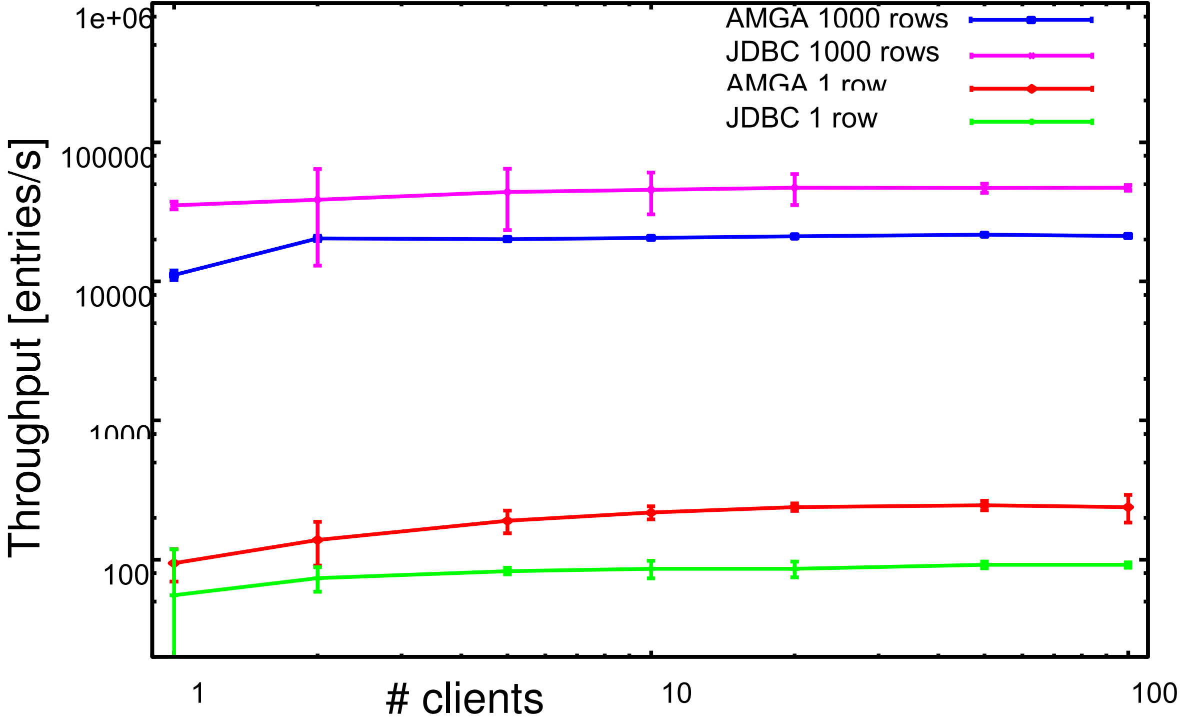
Security Concepts

- **AMGA provides high performance SSL connection**
  - SSL connections are Optional
  - SSL sessions
  - Authentication based on Password, X509 Cert, Grid Proxy, VOMS roles
  - Built-in group-management like AFS or via VOMS
- **Fine grained access control**
  - Posix-ACLs and Unix permissions for entries and collections
  - Attribute level security done through views


VOMS
Authenticate with X509 Cert
VOMS-Cert with Group & Role information
Resource management
AMGA
Oracle
VOMS-Cert

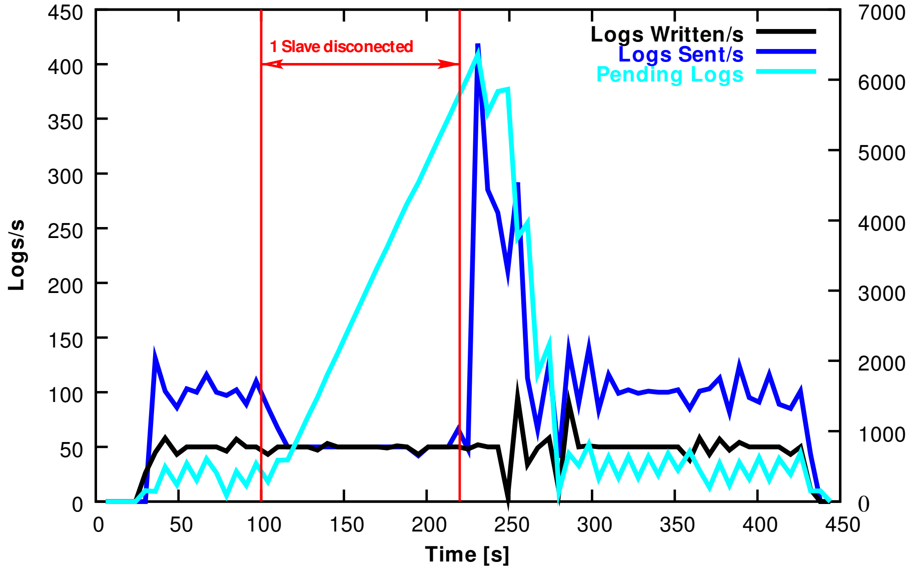
Performance

- **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

Replication Benchmarks

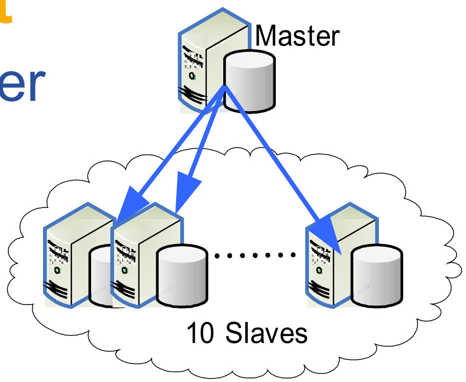


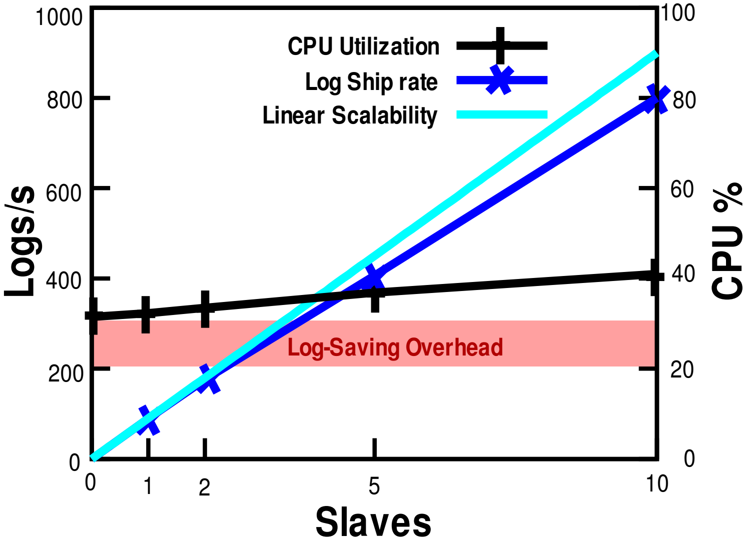
AMGA keeps logs for disconnected slaves

Reconnected slaves are brought up-to-date automatically

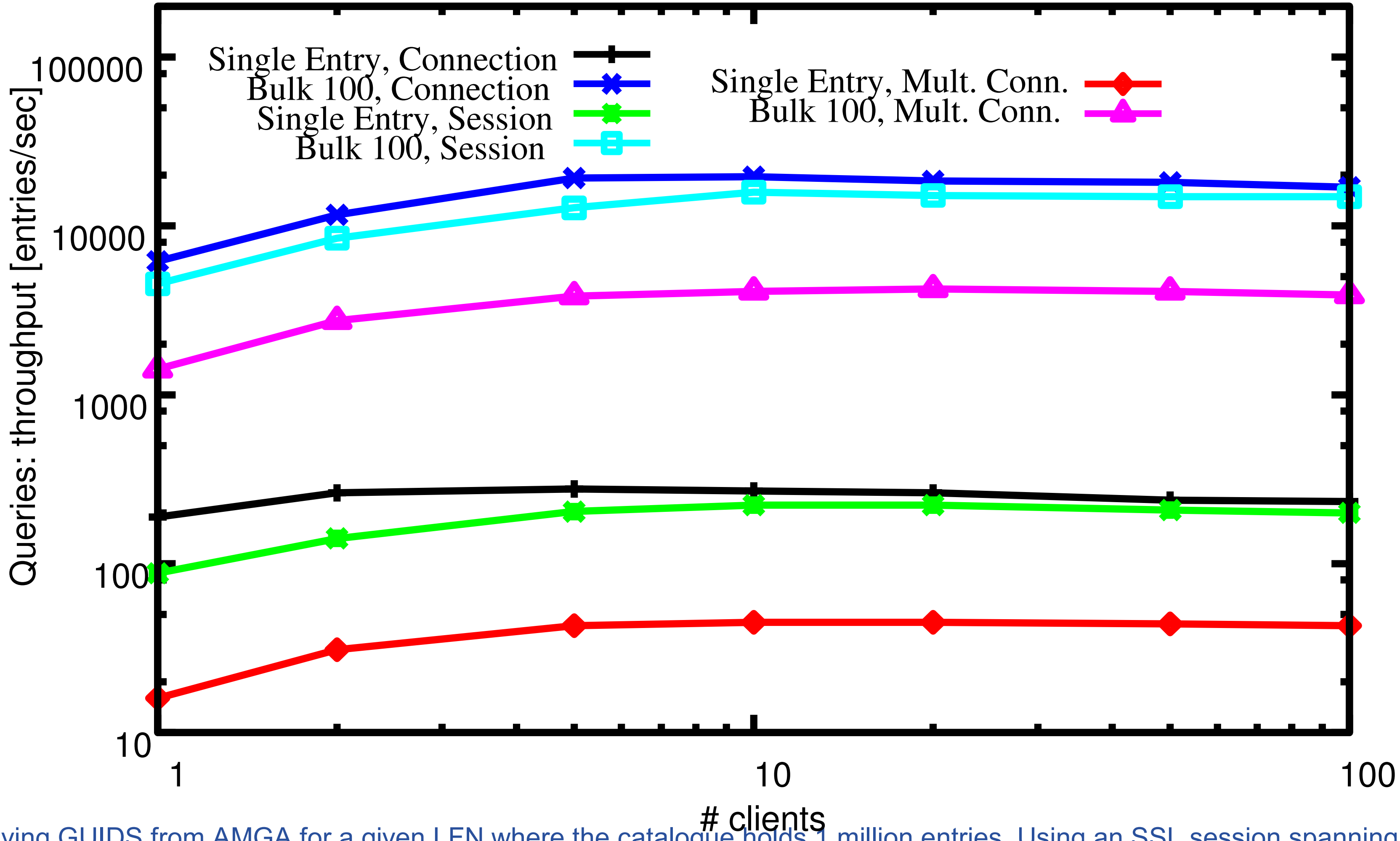
Fast recovery

- **Scalability test**
  - Setup: 1 master 10 slaves
  - Inserts at 90/s
  - 10% CPU overhead for 10 slaves



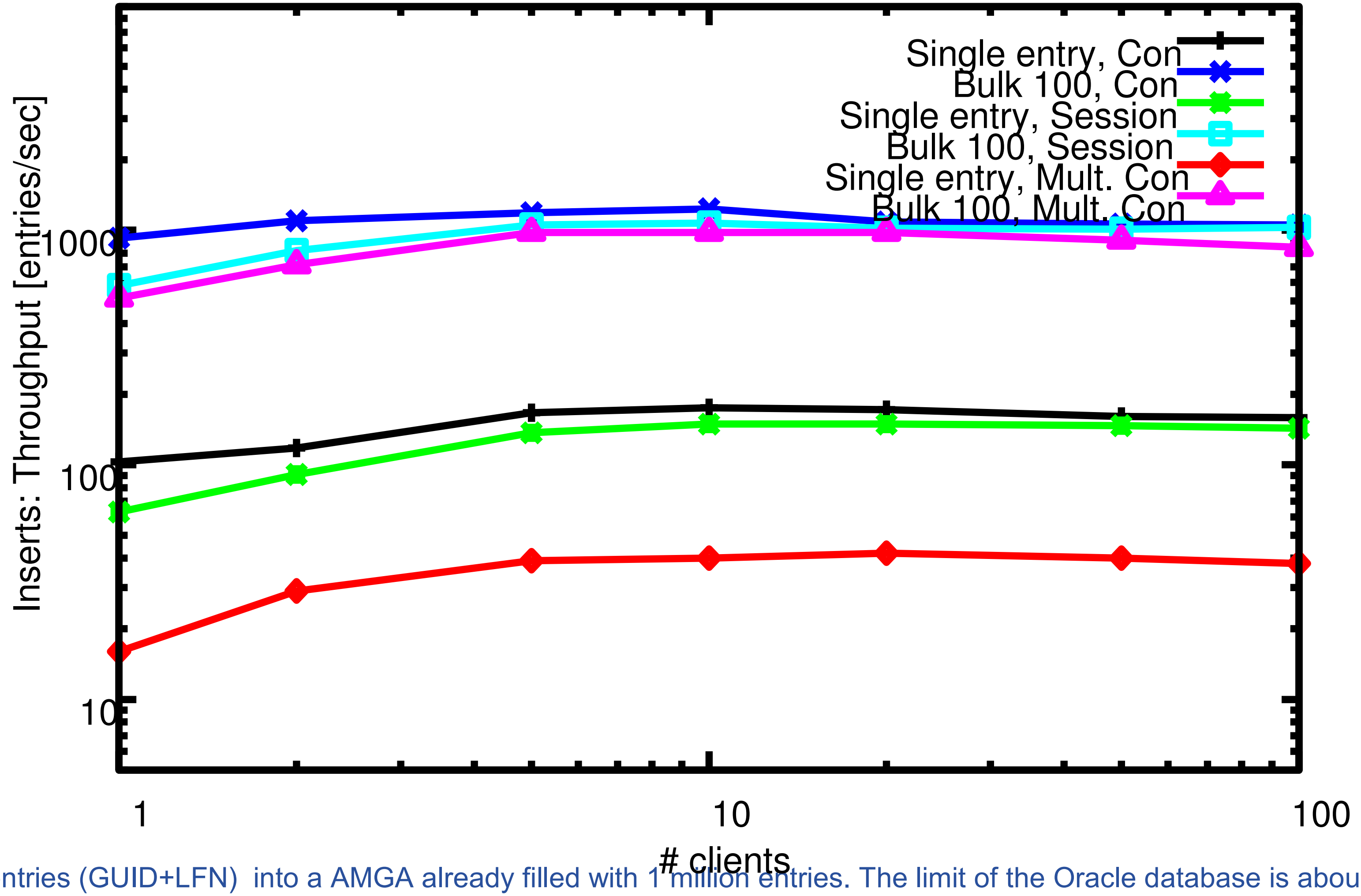


AMGA Performance: Queries



Retrieving GUIDS from AMGA for a given LFN where the catalogue holds 1 million entries. Using an SSL session spanning several TCP connections is about 10 times faster than having several TCP connections where the client authenticates in each. A single connection is only sightly faster than a session. A bulk operation with 100 entries is able to increase the speed over the single entry operation in all cases by about another factor of 10.

AMGA Performance: Inserts

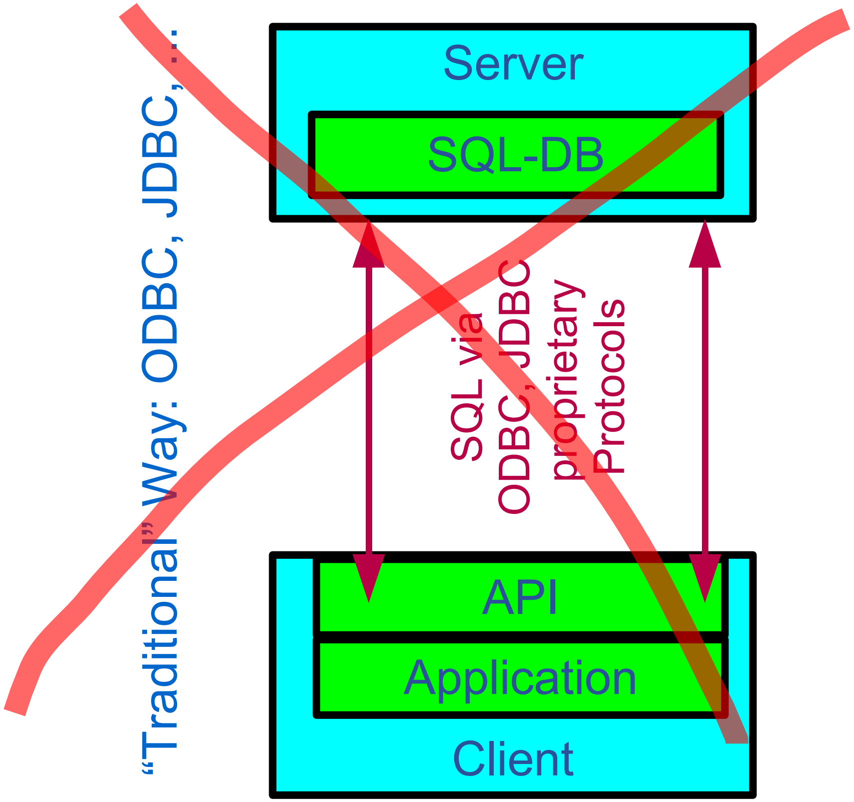


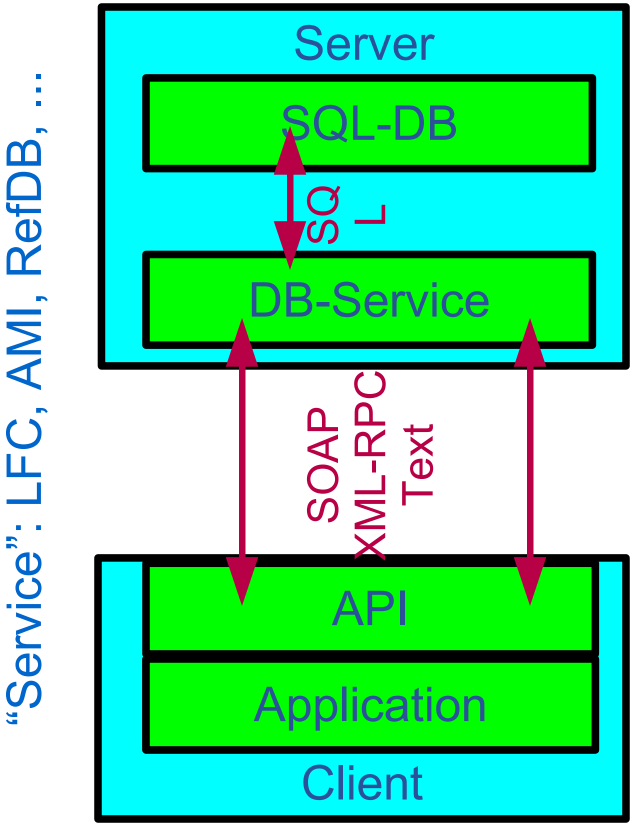
Inserting entries (GUID+LFN) into a AMGA already filled with 1 million entries. The limit of the Oracle database is about 1000 entries/sec which can be reached with bulk operations of 100 entries in all of the 3 connection modes of AMGA: Multiple individually authenticated TCP connections, a single SSL session spanning multiple TCP connections and a single TCP connection. The SSL session is nearly as fast as the single connection.

All measurements with an Oracle backend.

DB Access on the Grid

- Traditional DB access doesn't work on Grid:





+Performance
+Simple Implementation
– Security, Monitoring
– Authentication, resource management??

+Lightweight Client
+Security: GSI, x509
– Performance
– Implementation: State