

The AMGA Metadata Service

*B. Koblitz, IT-PSS
with N. Santos and V.Pose*

*Post C5 Presentation
June 30th, 2006*

- **What is AMGA? What is Metadata on the Grid?**
- **Interface definition**
- **Implementation of AMGA**
- **Comparison with other Catalogue services**
 - Benchmarks of LFC, FiReMan and AMGA
- **Replication of Metadata with AMGA**

- **AMGA (ARDA Metadata Grid Application) is the Metadata catalogue of gLite 1.5**
- **AMGA started out as ARDA's tool to investigate metadata access on the GRID**
- **Metadata is relationally structured data for grid jobs (lives normally in databases)**
- **AMGA works in 2 modes:**
 - Side-by-Side a File Catalogue (LFC): File Metadata
 - Standalone: General relational data on Grid
- **AMGA has 2 front ends:**
 - SOAP with PTF standardised interface
 - Text-based TCP streaming protocol (proprietary, documented)
- **<http://project-arda-dev.web.cern.ch/project-arda-dev/metadata/>**

- **AMGA implements a common interface designed in close collaboration of gLite-DM and ARDA teams**
(P. Kunszt, R. Rocha, N. Santos, B.K.)
contains man ideas from UK GridPP Metadata group,
LHCb (Bookkeeping, GANGA), HEP...
- **EGEE standard for metadata access (AMI, FiReMan)**
- **Design Ideas:**
 - Versatile: Performance (HEP), Security (BioMed)
 - Provides simple relational data manipulation interface
 - Modular: Interface for Entry manipulation, schemes, security
 - Few requirements on back end, can be SQL-DB, XML...
 - Allows stateless & statefull implementations
- **Description of WSDL at**
<https://edms.cern.ch/document/573725>

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

Examples

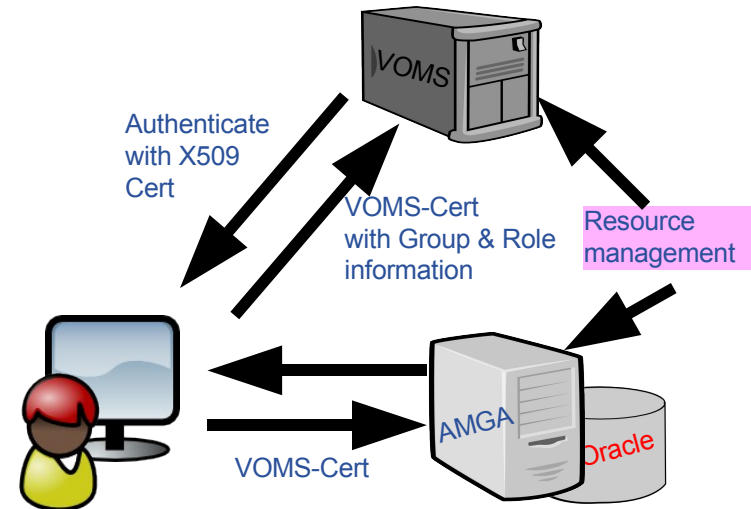
```
mkdir /jobs
addattr /jobs jobStatus int
addentry /jobs/job1 jobStatus 0
updateattr /jobs jobStatus 1 jobID>100
selectattr /DLibrary:FileName /DLAudio:Author /DLAudio:Album
'/DLibrary:FILE=/DLAudio:FILE and like(/DLibrary:FileName, "%.mp3")'
```

- Security very important for BioMed, not for HEP

Security ↔ Speed

- AMGA supports:

- SSL connections (Optional!)
- Authentication based on Password, X509 Cert, Grid Proxy
- Posix-ACLs and Unix permissions for entries and collections
- Built-in group-management like AFS or via VOMS



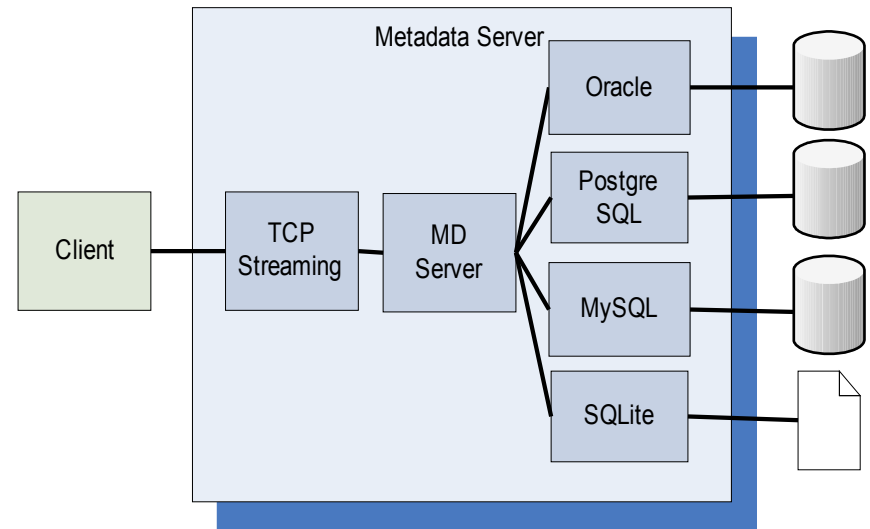
- Currently no security on attribute basis

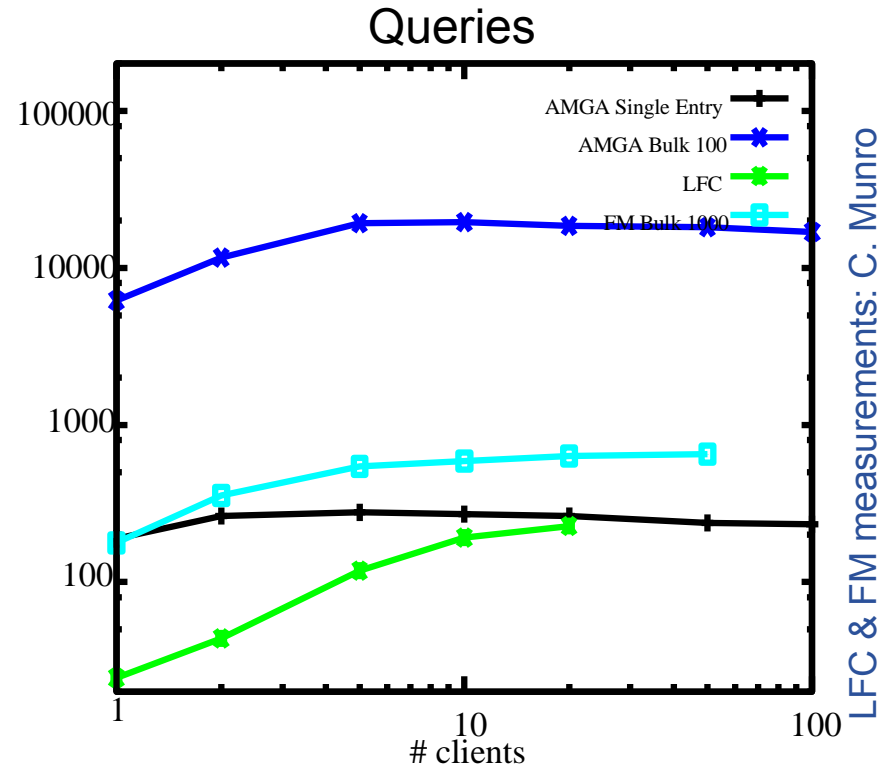
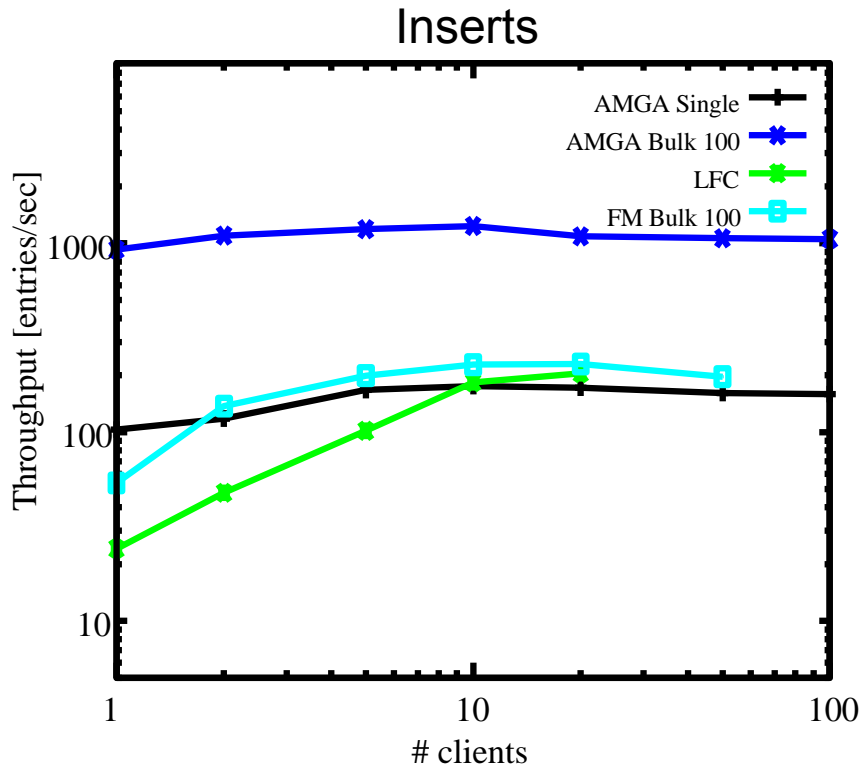
- AMGA allows to create views: Safer, faster, similar to RDBMS

Security tested by GILDA and BioMed teams: Very positive feedback for built-in group management & ACLs

- **AMGA Implementation:**

- SOAP and Text frontends
- Streamed Bulk Operations
- Supports single calls, sessions & connections
- SSL security with grid certs (negotiated by client)
- Own User & Group management + VOMS
- PostgreSQL, Oracle, MySQL, SQLite backends
- Works alongside LFC
- C++, Java, Perl, Python clients



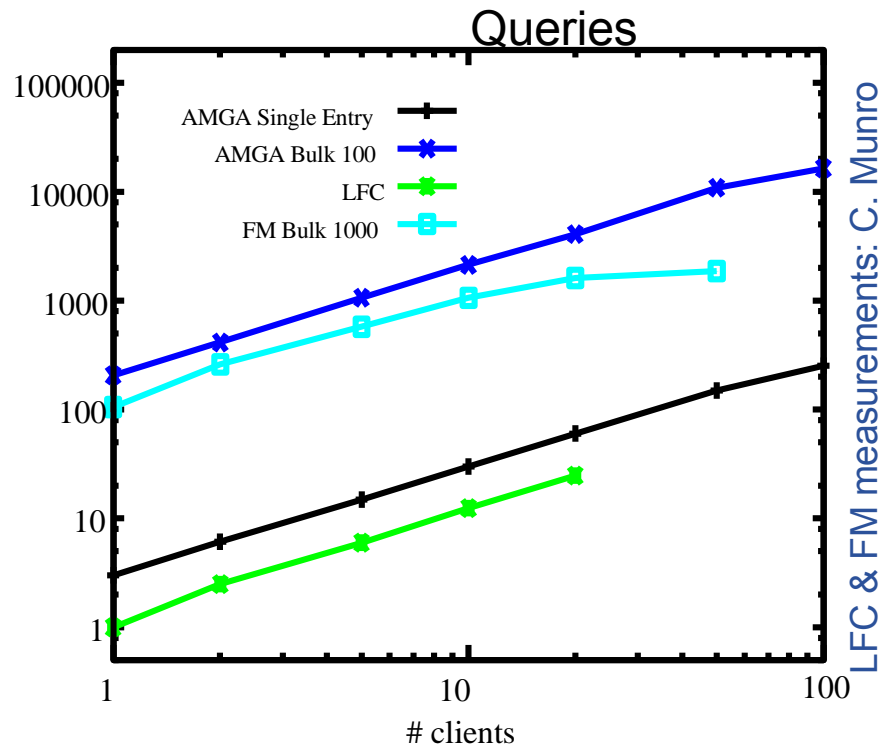
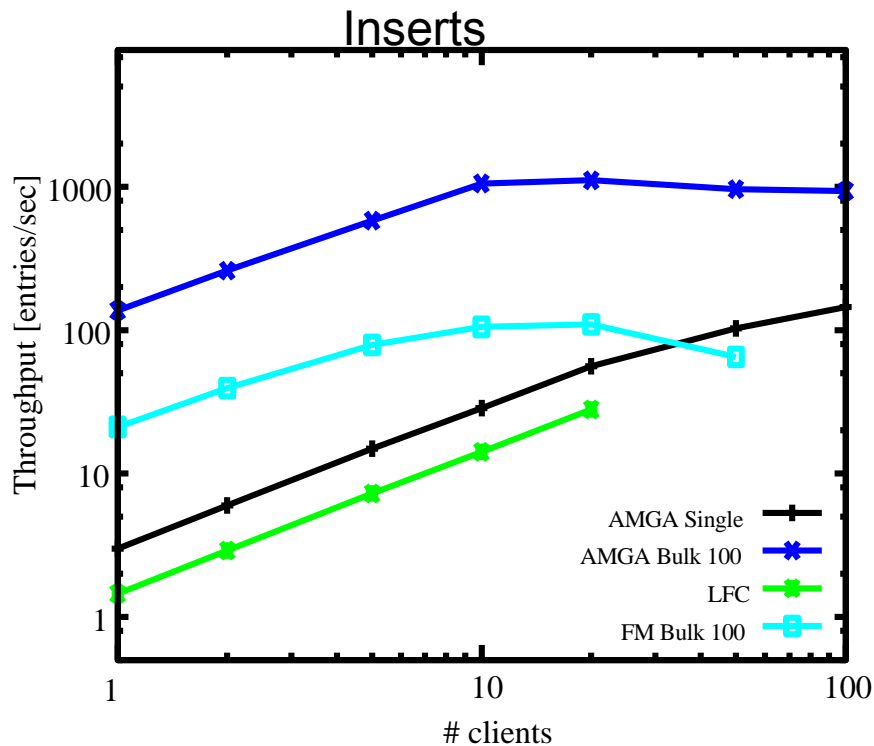


LFC & FM measurements: C. Munro

- 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



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

- **Close collaboration with LFC team to improve LFC implementation:**
 - Performed many benchmarks and bottleneck analyses
 - Would like to implement thread pools in LFC:
 - LFC currently limited to compiled in number of threads
 - DB Connection pools key to good performance
 - Would like to implement session caching in LFC
 - Without session caching upper limit of SSL connections is 30/s

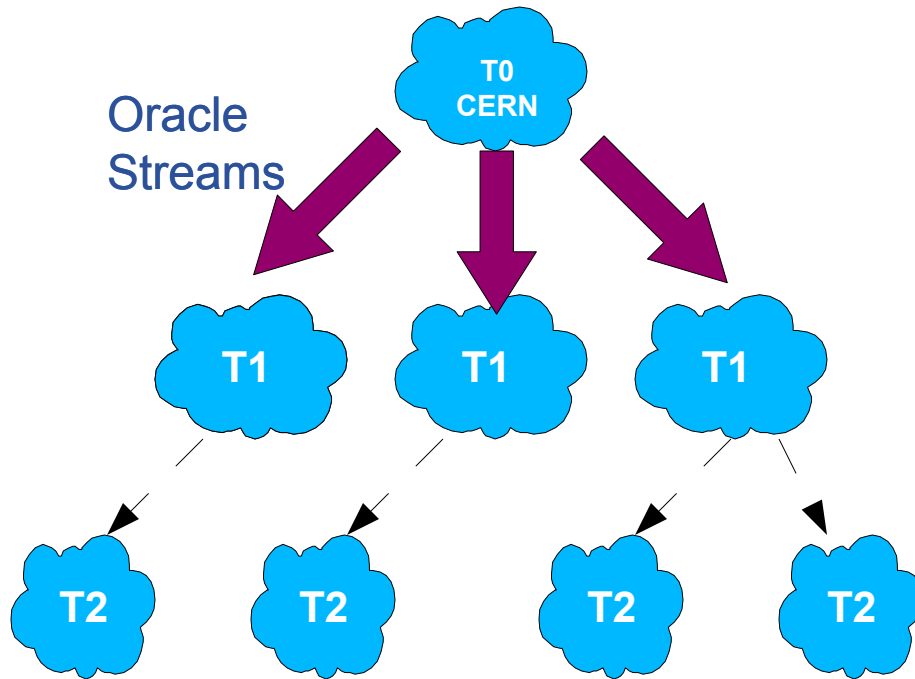
- **AMGA in (pre-)production within several projects:**
- **LHCb and ATLAS: GANGA**
 - Highly dynamic relational data for tracking jobs
 - AMGA used for communication between user clients
- **LHCb Logging and Bookkeeping**
 - 15 Million log entries of jobs
- **EGEE BioMed Medical DataManger**
 - Highly secure access to medical images metadata
 - Many different security roles, uses VOMS
- **Generic applications:**
 - Metadata for EGEE-GILDA DigitalLibrary
 - UNOSAT project: Satellite images metadata, GIS

AMGA-Replication implemented as PhD project (N. Santos):

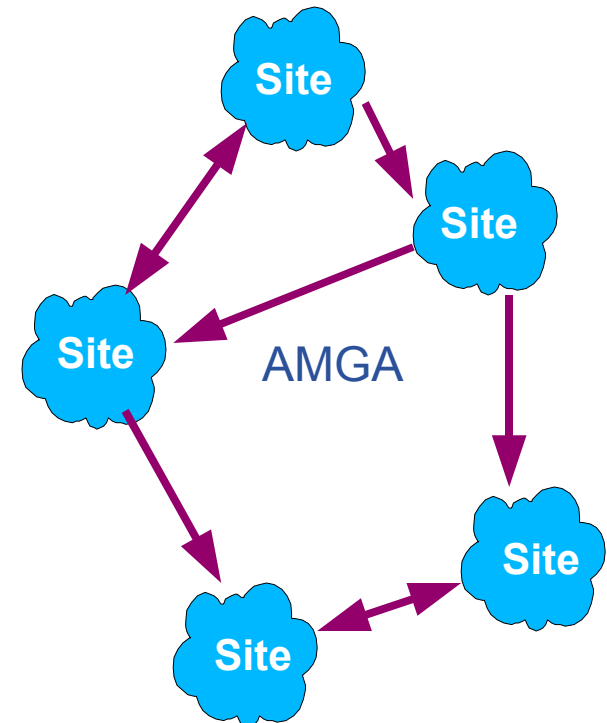
- **Scalability:** More connections, smaller latencies
- **Reliability:** Avoid single point of failures
- **Administration:** Less DBA time central management possible
- **AMGA implements replication & federation itself:**
 - Asynchronous log-shipping of Metadata commands: **cross-vendor**
 - **Users can changes replication and federation setup**
- **Initial implementation currently tested**
- **Federation capabilities essential for EGEE Medical DataManager**
 - Keep sensitive patient information distributed

Database Replication for LHC:

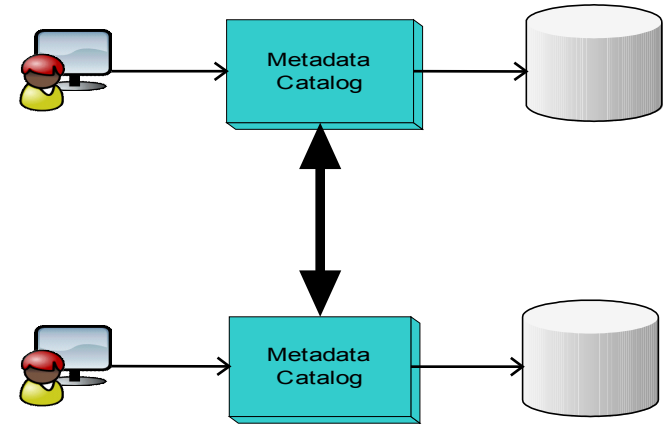
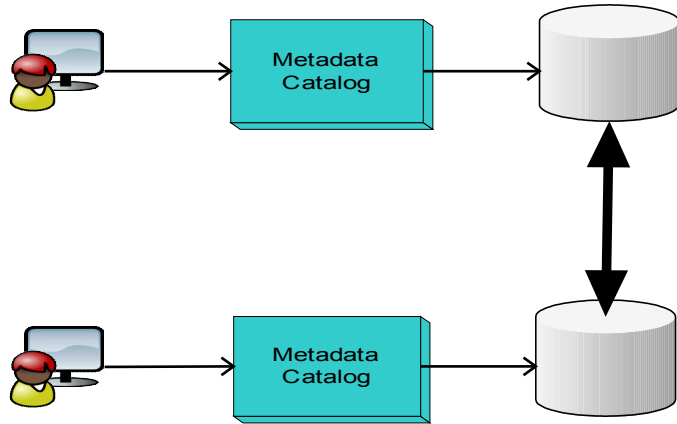
EGEE Grids (e.g. BioMed):



- Tiered Architecture
- Replication with Oracle Streams to Tier 1
- Require Oracle on T1



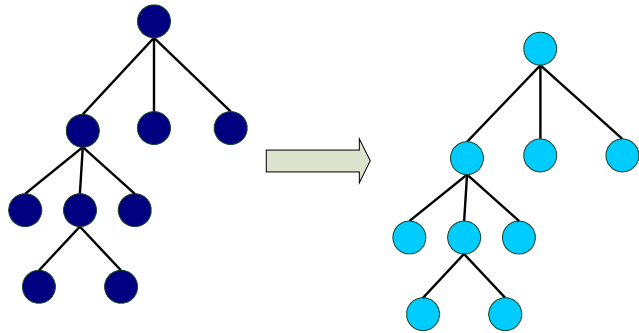
- Loose cluster of sites
- Diverse infrastructure (MySQL, Oracle, ...)



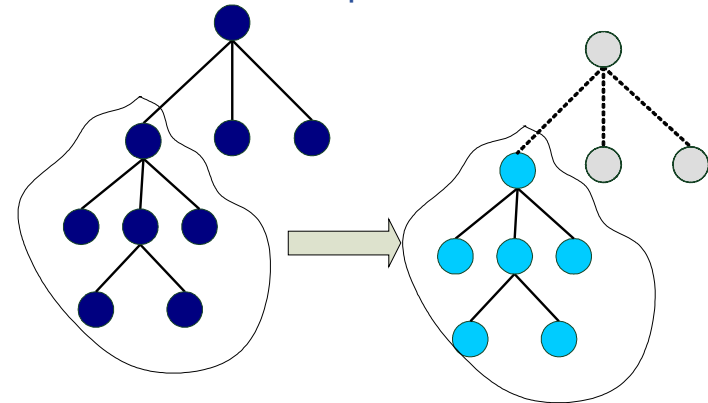
- **Database replication:**
 - Vendor dependent
 - Replicates nearly all application data
- **3D Project (IT-PSS):**
 - CERN -> T1 replication with Oracle Streams
- **Collaboration with 3D team on Monitoring & Benchmarks**
- **Metadata replication:**
 - DB independent, cross DB
 - Complex implementation
 - Customized to Metadata

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

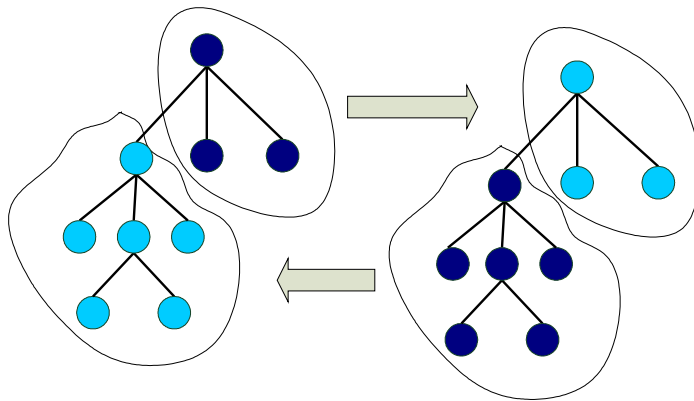
Full replication



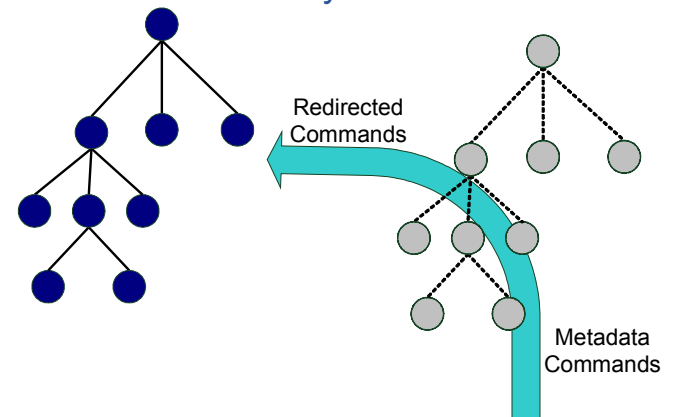
Partial replication

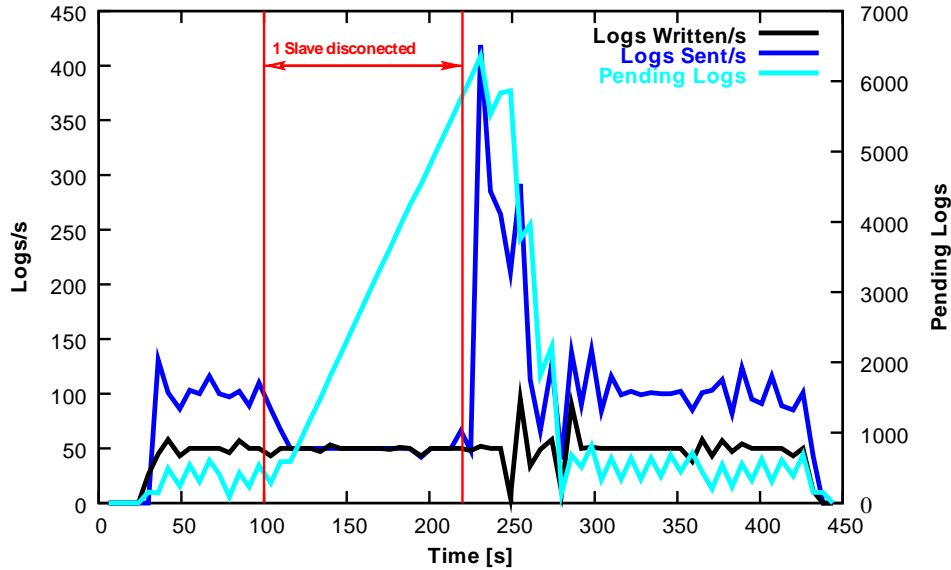


Federation



Proxy





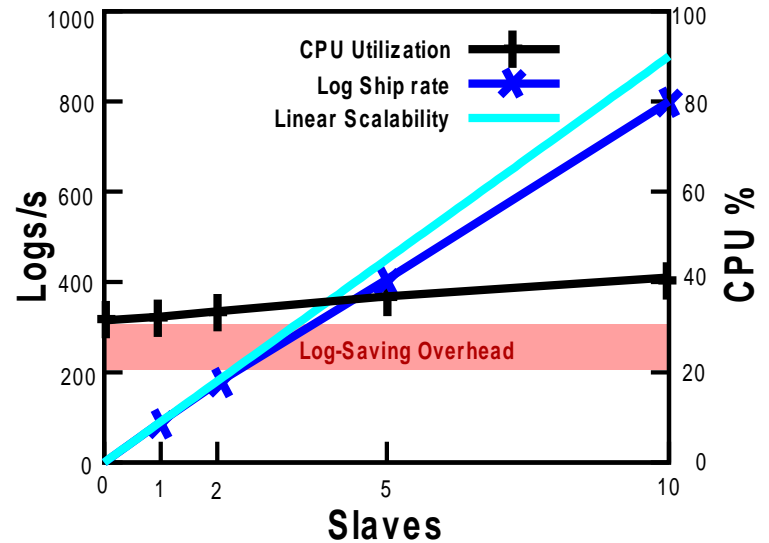
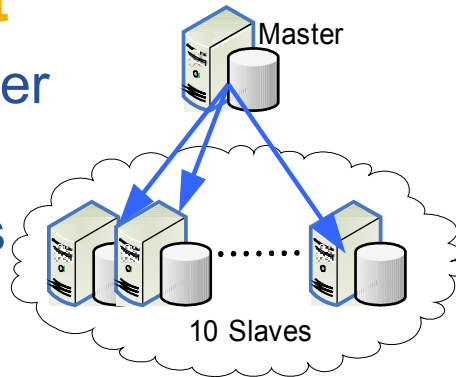
AMGA keeps logs for disconnected slaves

Reconnected slaves are brought up-to-date automatically

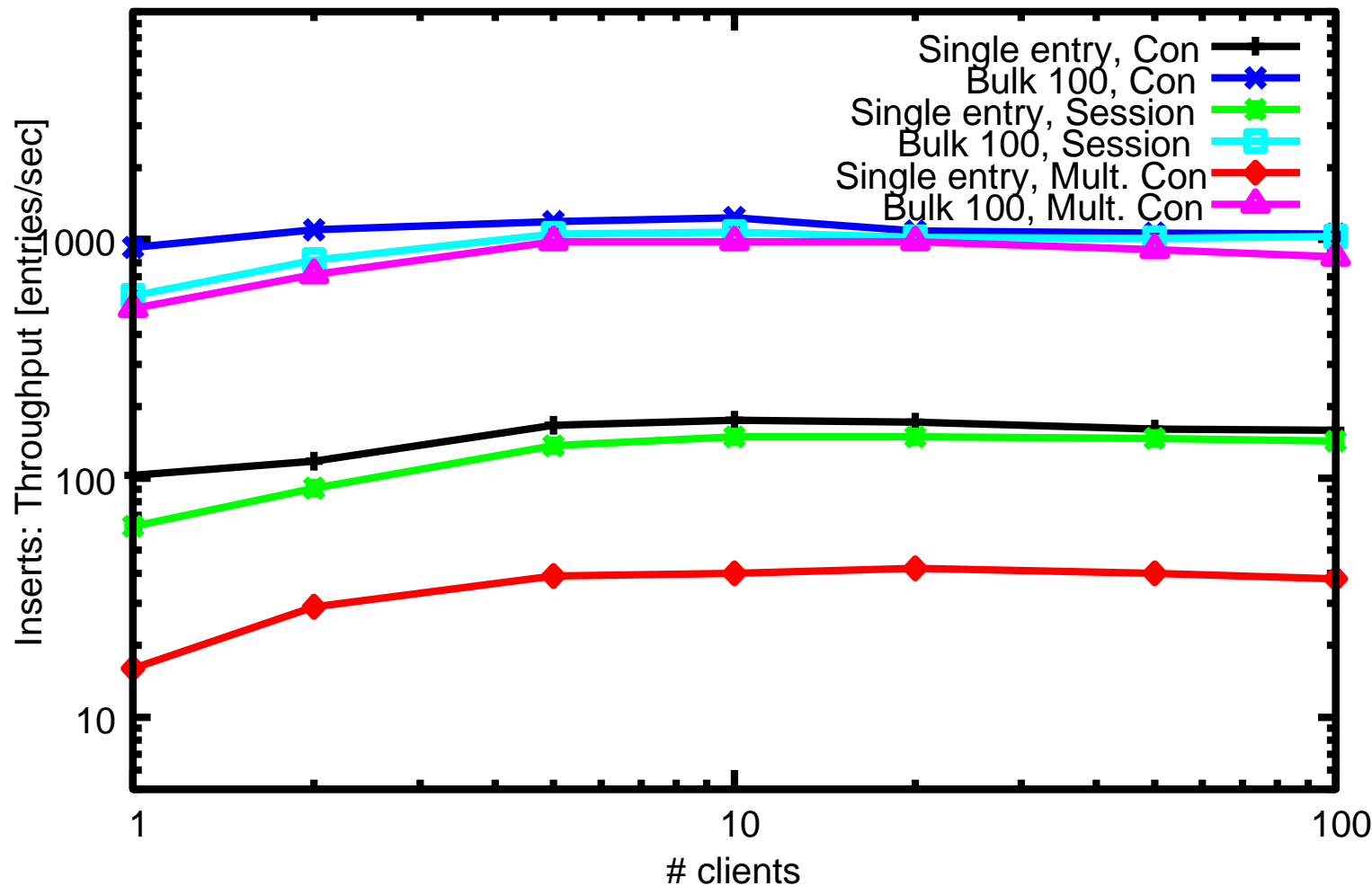
Fast recovery

Scalability test

- Setup: 1 master
10 slaves
- Inserts at 90/s
- 10% CPU for 10 slaves

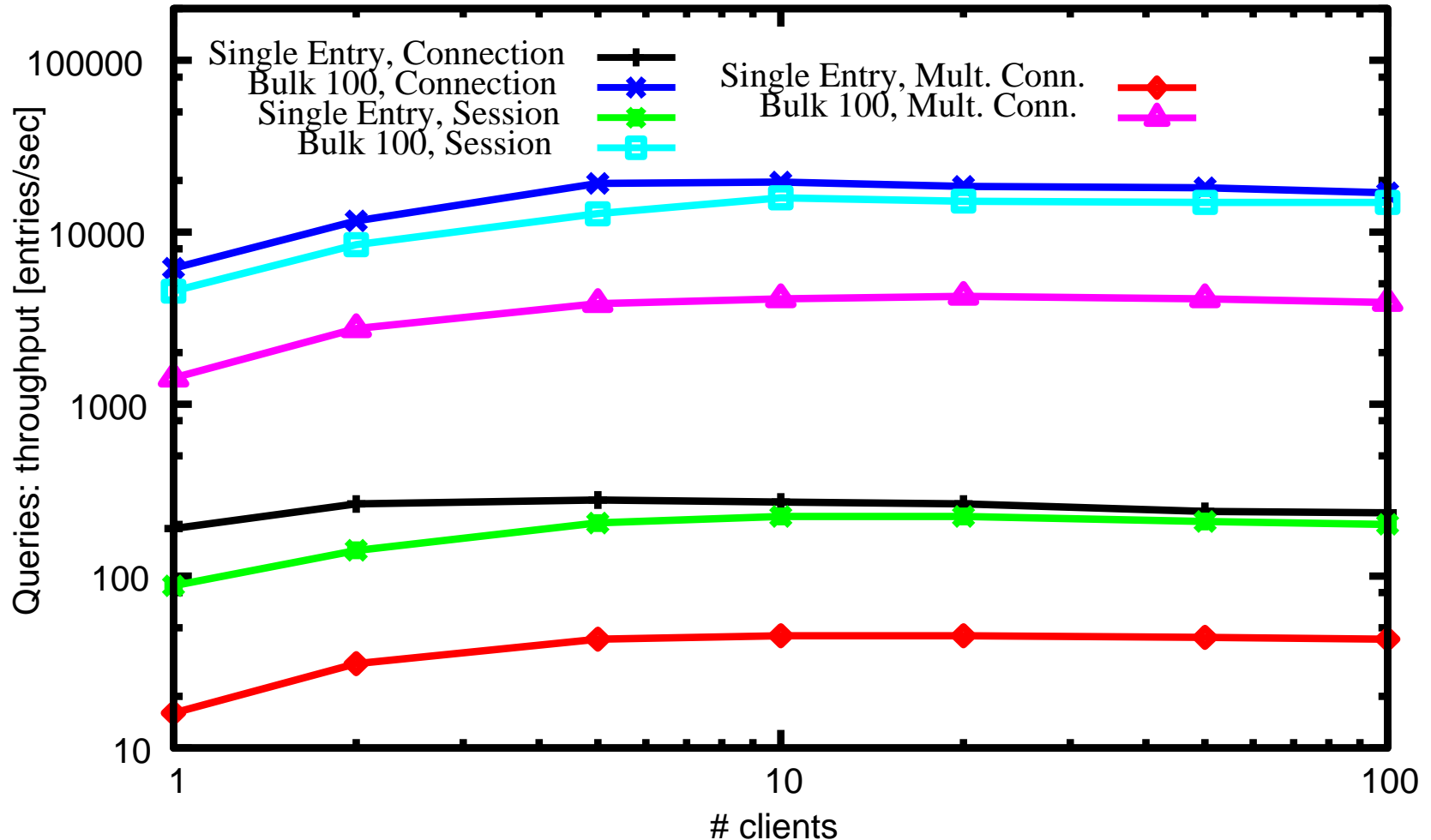


- ARDA metadata server **AMGA** in gLite 1.5, back in 3.1?
- Implements **EGEE Metadata Interface**
- **AMGA** has seen **heavy performance/stability testing**
 - AMGA's protocol throughput 10 times higher than LFC's or FiReMan's due to combination of sessions and bulk streaming
- **AMGA** includes very **flexible** (and optional) **security**
- **AMGA** currently in preproduction for
LHCb(GANGA, bookkeeping), MDM, DigitalLibrary, UnoSat...
- **Replication features available in AMGA**
- **AMGA Web Site**
<http://project-arda-dev.web.cern.ch/project-arda-dev/metadata/>



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.



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 slightly 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.