ACAT Conference, DESY, 25th May 2005

Performance Comparison of the LCG2 and gLite File Catalogues

Craig Munro / IT-GD, CERN / Brunel University

EGEE is a project funded by the European Union under contract IST-2003-508833

Overview

- Background
- Catalogue Architectures
- Test Methodology & Setup
- Performance Comparison
- Further Work
- Conclusions

Background

- File Catalogue provides mapping between Logical and Physical File
- Data Challenges of 2004 exposed limitations in LCG Data Management tools
- LCG File Catalogue (LFC) developed to address problems with EDG RLS
- File and Replica Management (FiReMan) catalogue developed as part of the EGEE project's gLite middleware
- Suite of tests developed to check the functionality and performance of LFC catalogue adapted for FiReMan

LCG File Catalogue (LFC)

- Connection orientated, stateful, written in C
  - Cursors used to return results
- Hierarchical Filesystem view of entries
- User exposed transaction API
  - But no bulk calls
- Supports Oracle and MySQL backends
- GSI for Authentication and Authorisation
- Unix Permissions and POSIX ACLs

File Replica Management (FiReMan) Catalogue

- Stateless Web Service Interface
- Hierarchical Filesystem view of entries
- Supports Bulk Operations
  - Limited transaction support (limited to single message)
- Supports Oracle and MySQL backends
- VOMS integration
- Unix Permissions and POSIX ACLs

Test Methodology

- Multi-threaded C client program written to test each type of operation (insert, query, delete etc.)

```
./create_files -d /grid/dteam/arda/insert/
-f $num_files -t $num_threads
```

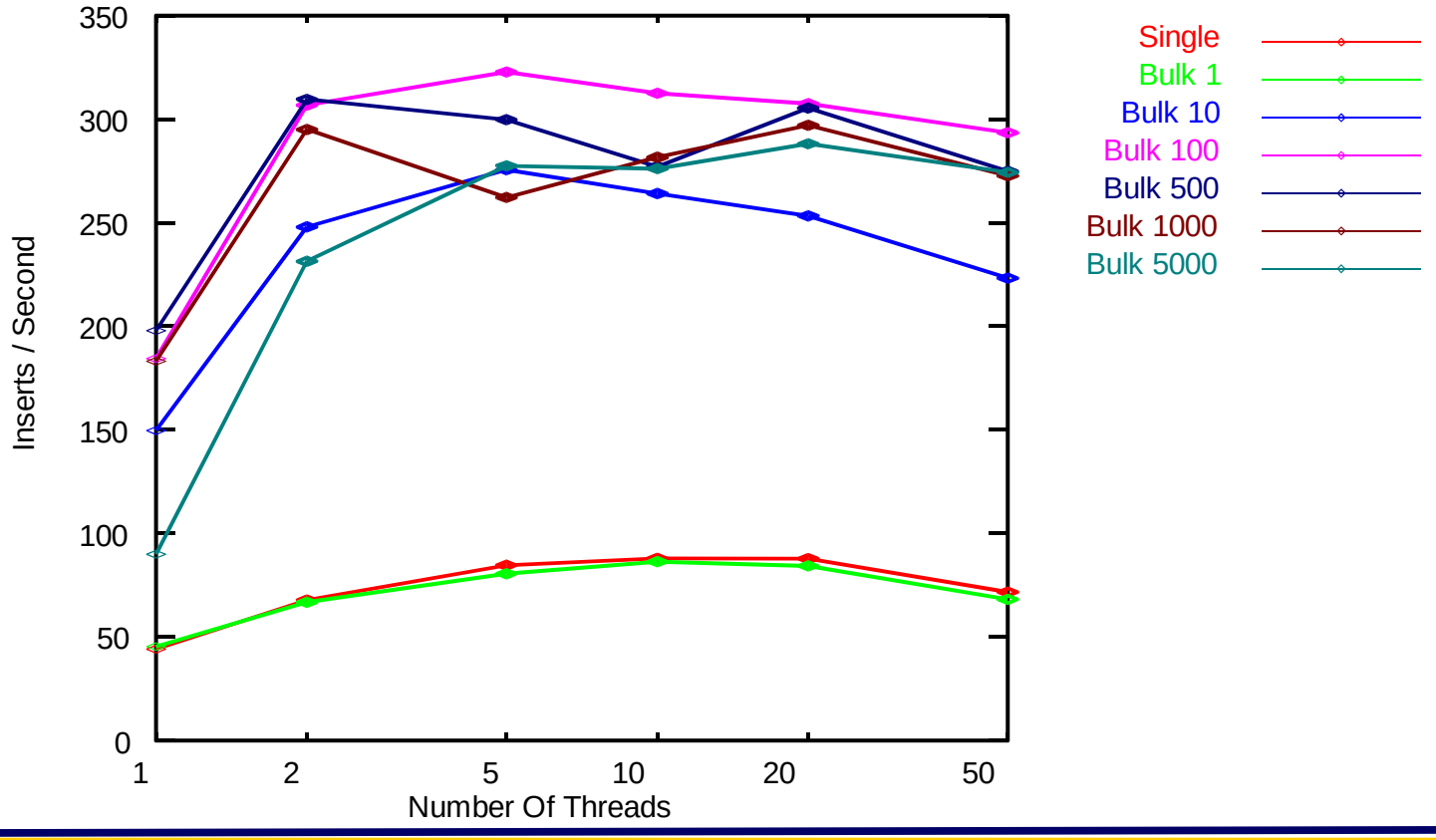
- Typically, each operation performed O(1000) times in client program, mean result returned
- Test performed several times, average taken
- Any entries removed before next test run

Test Setup

- LFC and FiReMan tests performed on identical hardware
- Server:
  - Dual Xeon 2.4Ghz with 2048 MB RAM
  - Catalogue and Oracle DB running on same machine
- Client:
  - PIII 800 Mhz, 512 MB RAM
  - configurable number of threads
- 100 Mb/s LAN
- Insecure FiReMan and LFC

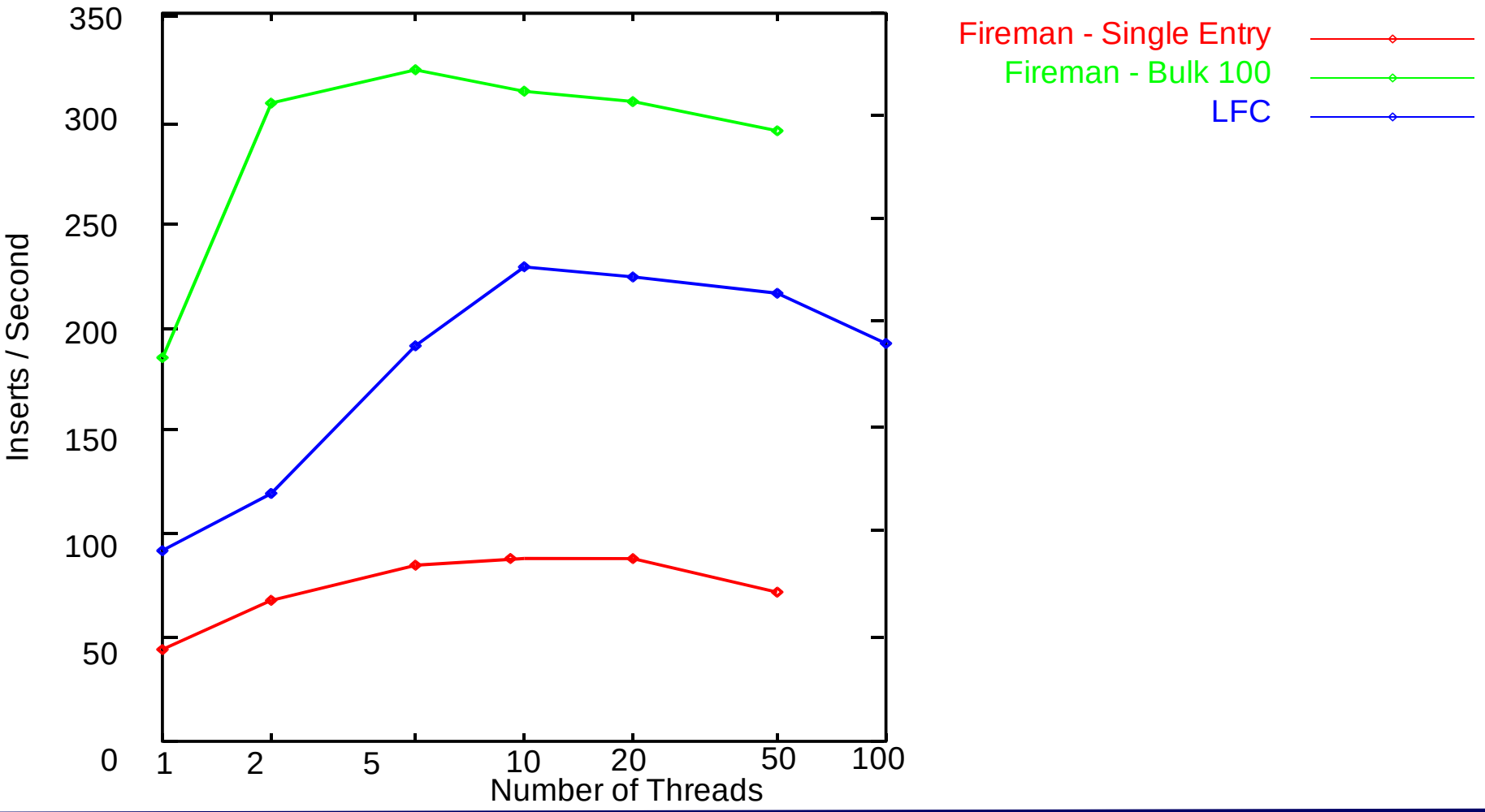
FiReMan Performance - Inserts

- Insert rate for increasing bulk size and number of clients
  - Time-outs above 50 clients



FiReMan Performance - Insert

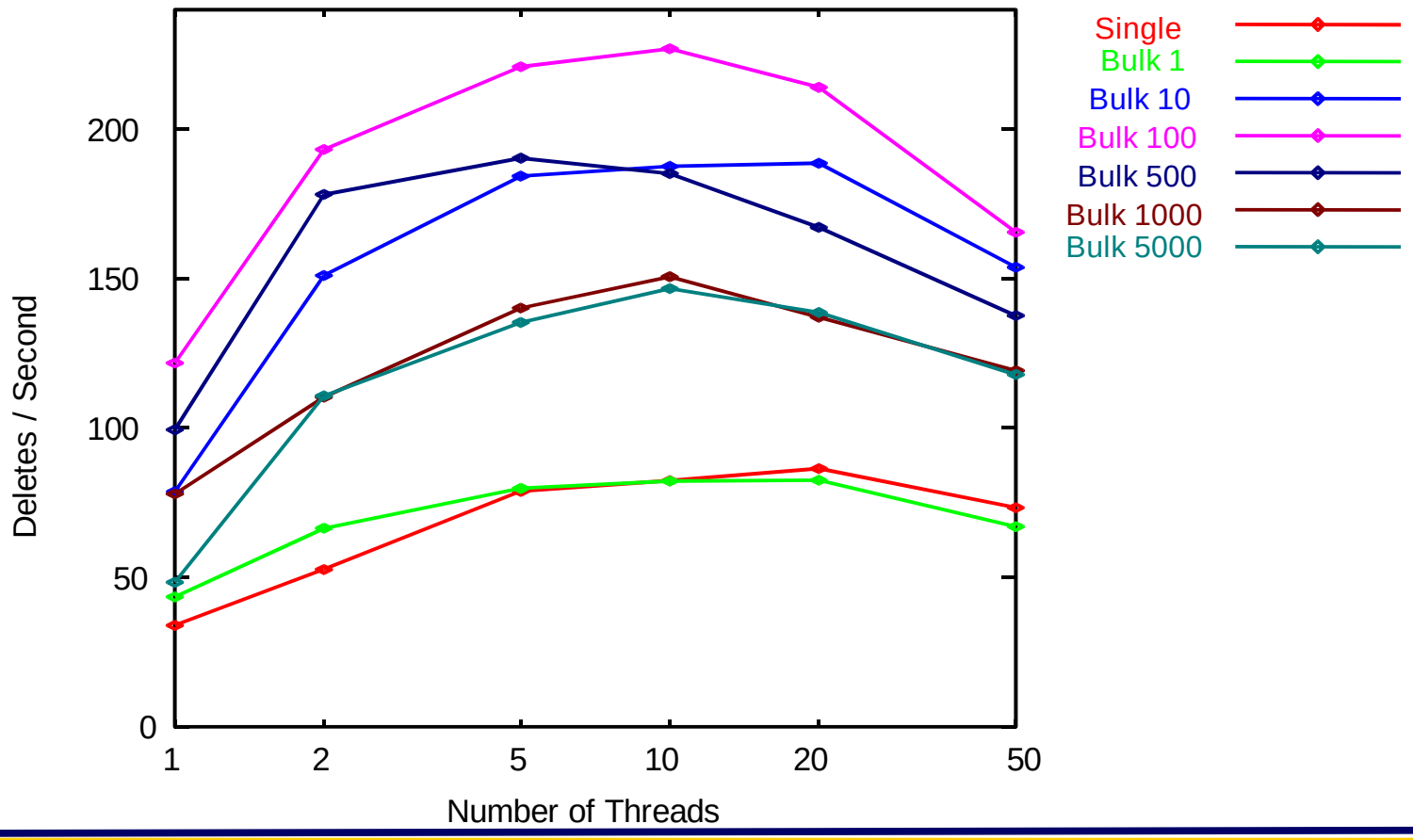
- FiReMan – LFC Insert Comparison



FiReMan Performance - Delete

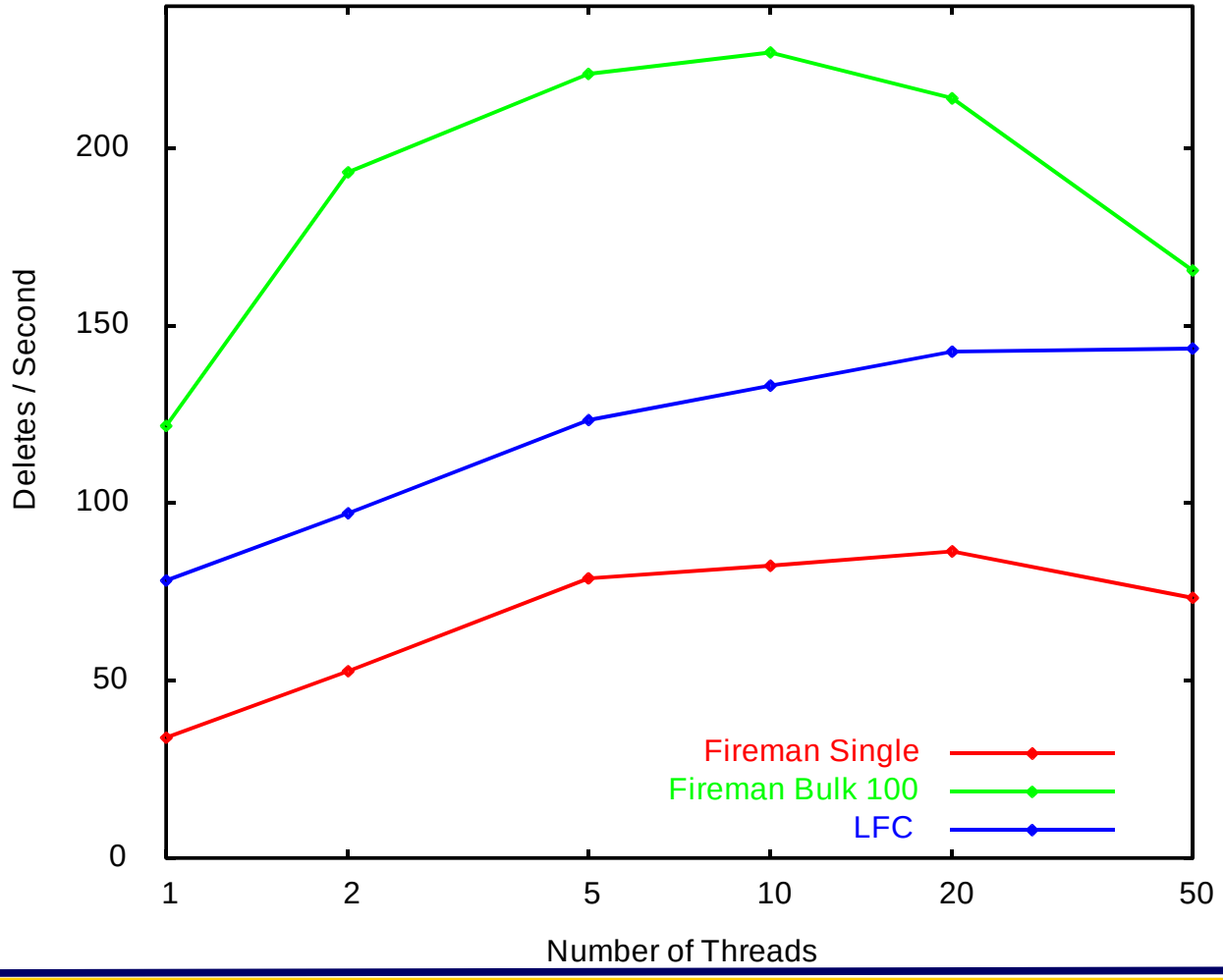
- Deletion Rate:

  Time-outs above 50 clients



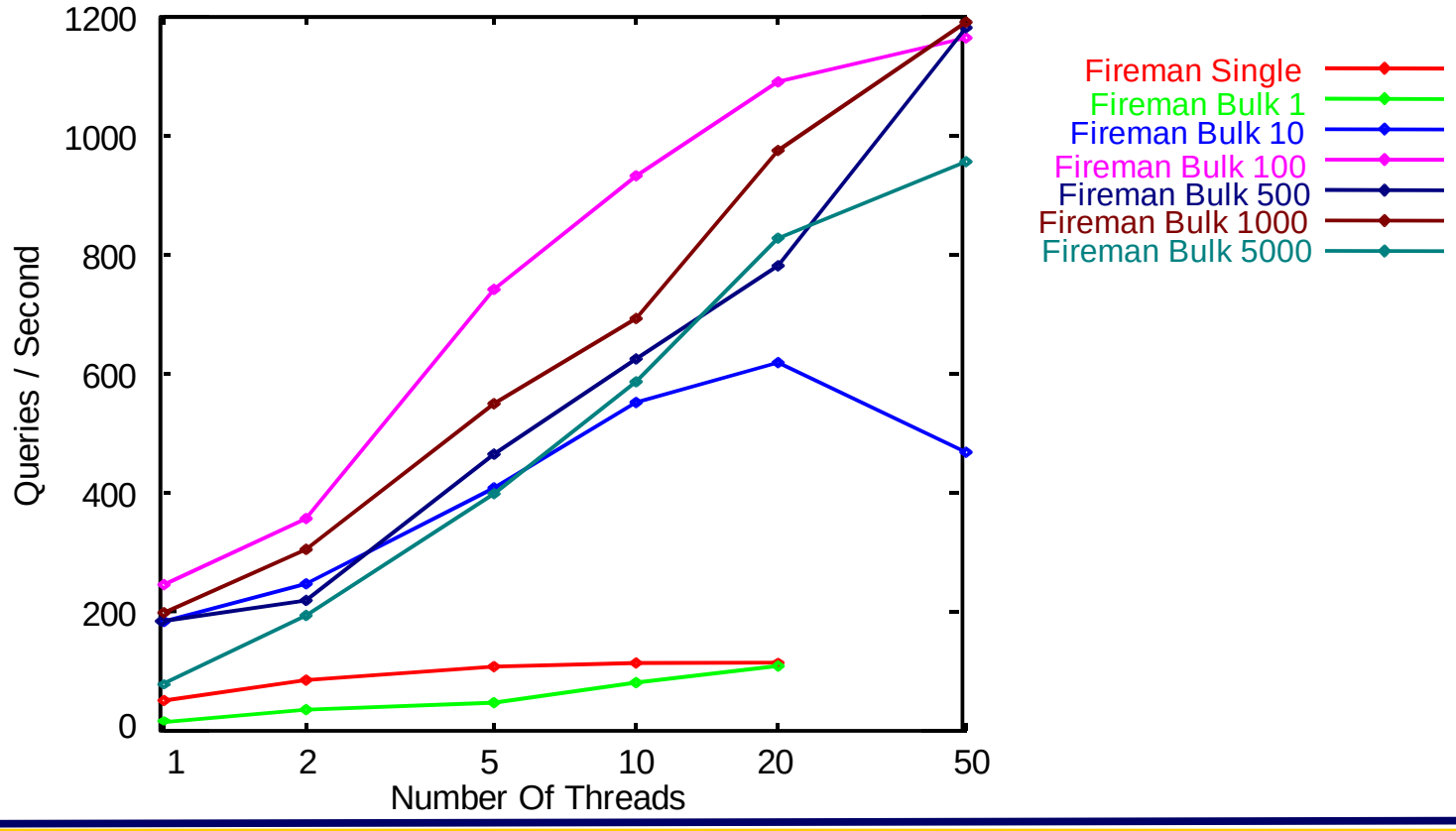
FiReMan Performance - Delete

- Comparison with LFC:



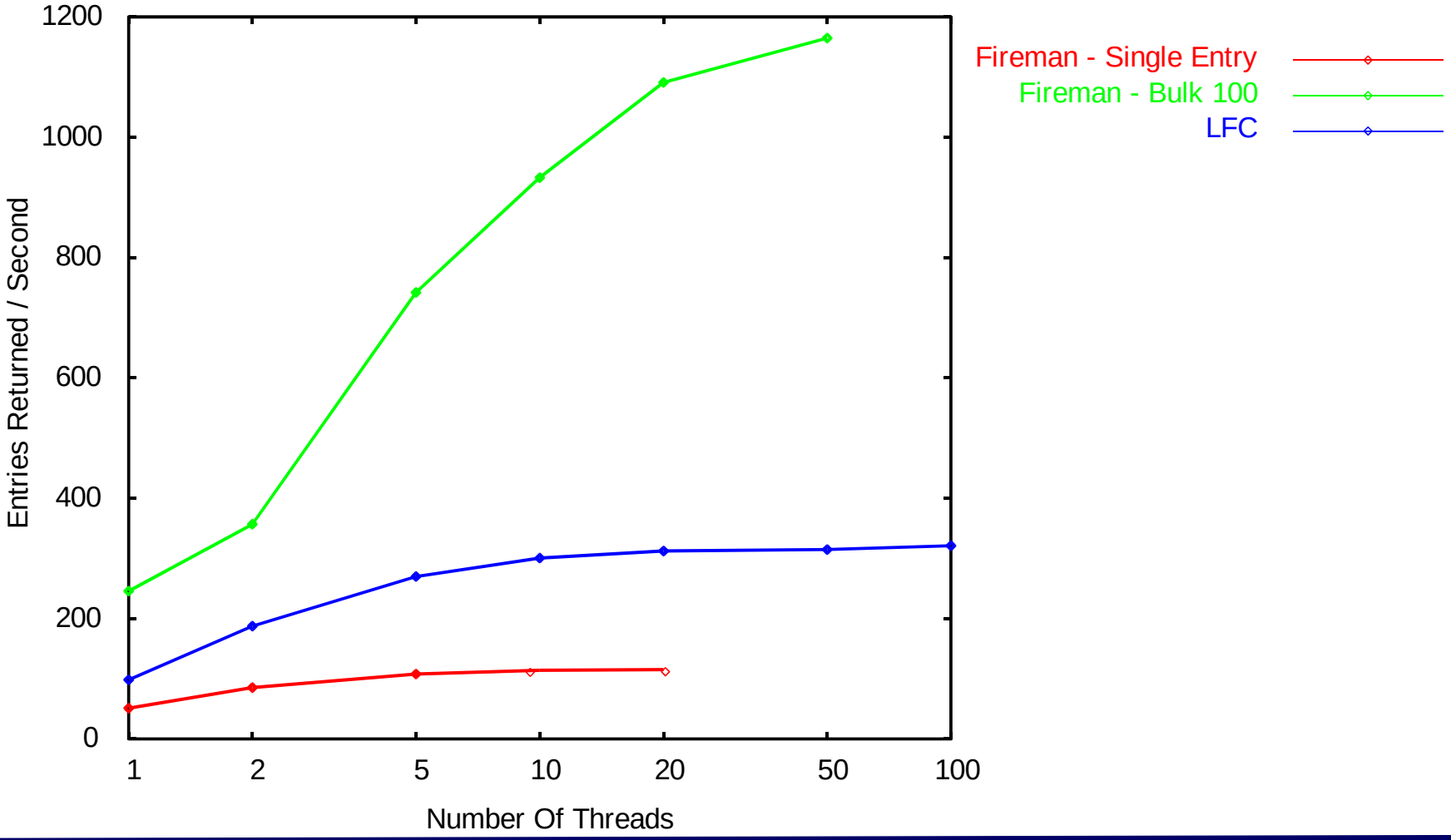
FiReMan Performance - Queries

- Query Rate for an LFN
  - Time-outs above 20 clients for single / bulk 1
  - Time-outs above 50 clients for larger bulk sizes



FiReMan Performance - Queries

- Comparison with LFC:

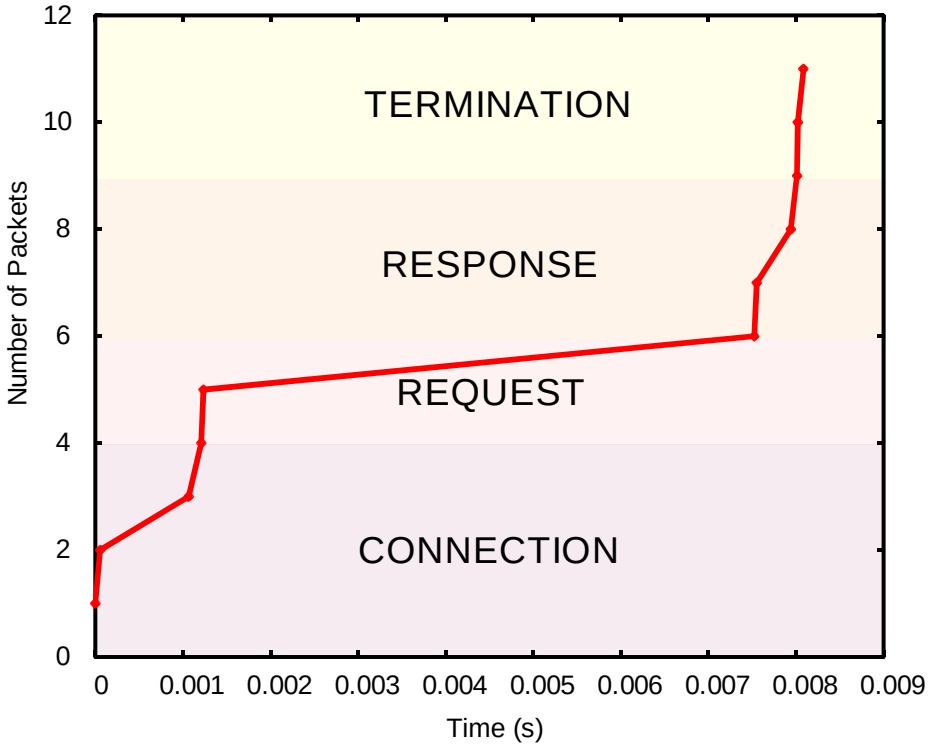


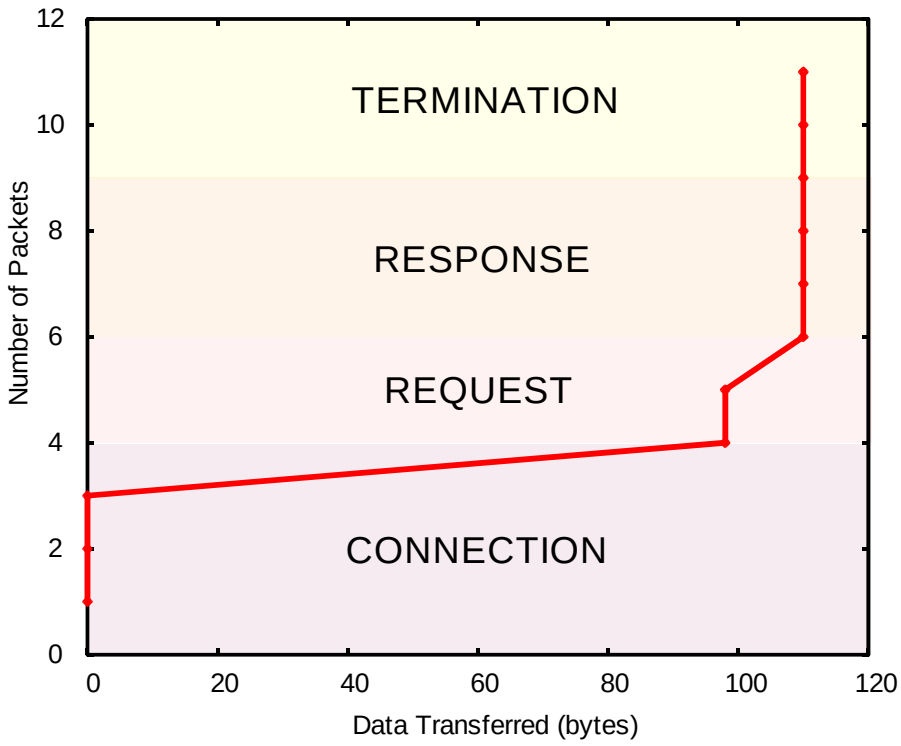
Performance Summary

- Both LFC and FiReMan offer improvements over EDG RLS performance
- LFC better for single operations
  - Smaller protocol overhead
  - SOAP overhead for small payloads
- FiReMan better for bulk operations
  - SOAP not all that bad
  - LFC does not support bulk operations
- Many factors affecting performance
  - Start by analysing protocol

LFC Protocol Analysis

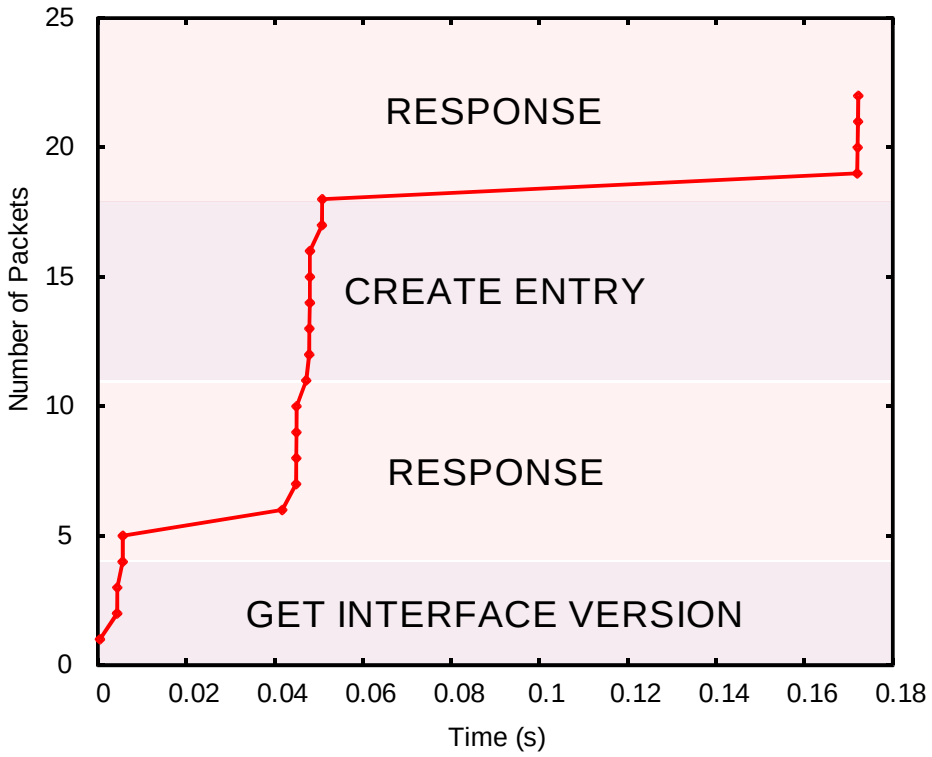
- Create 1 entry in insecure lfc catalogue using API

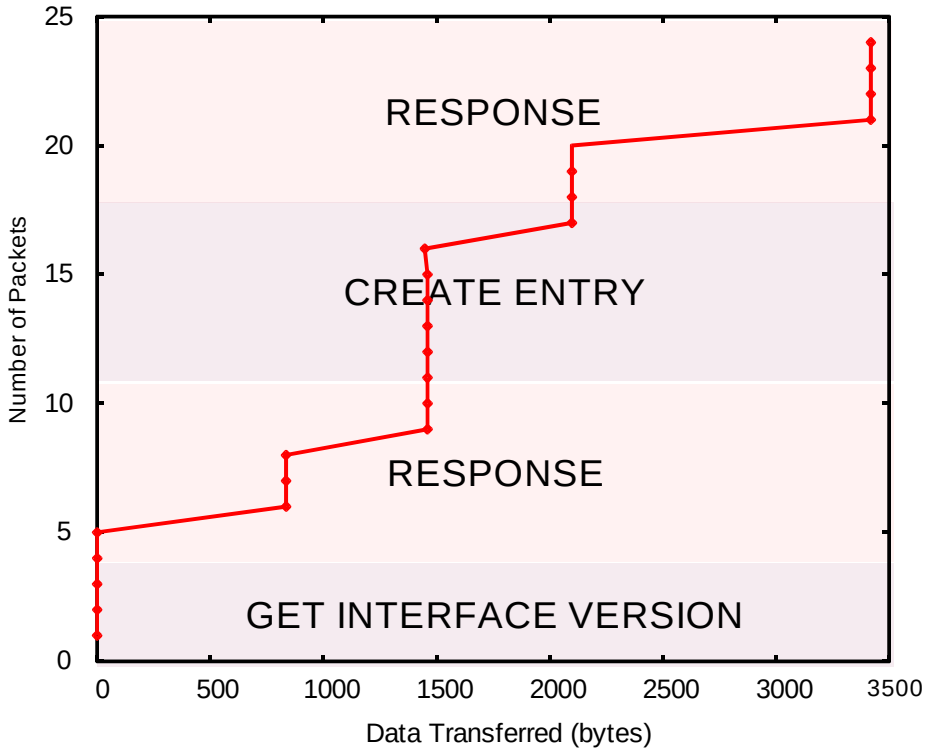




FiReMan Protocol Analysis

- Create one entry in insecure FiReMan catalogue using API

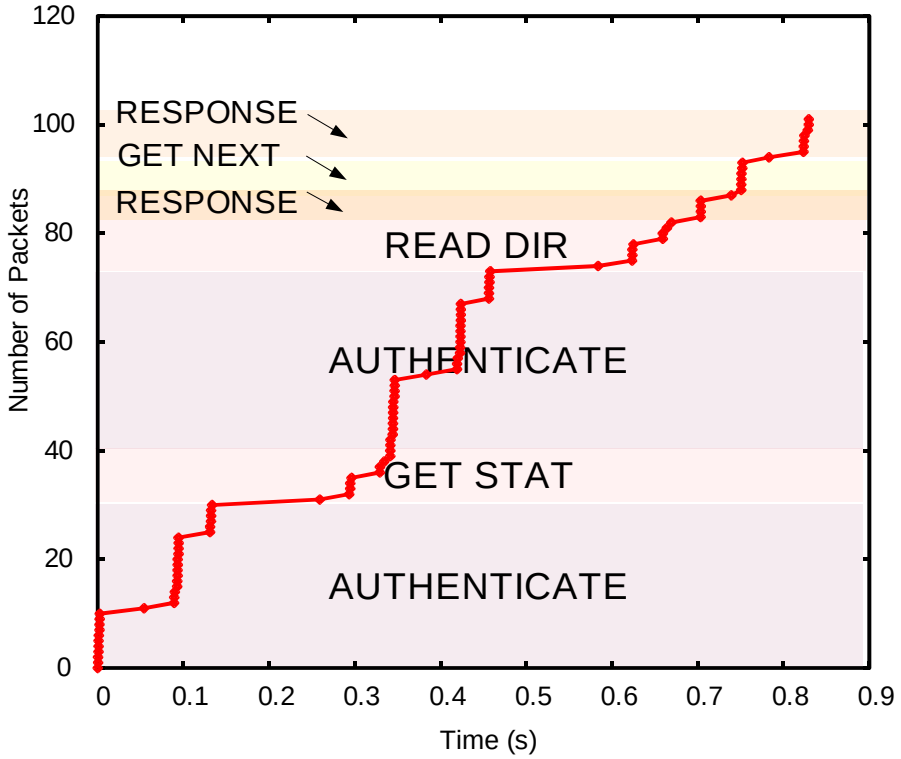


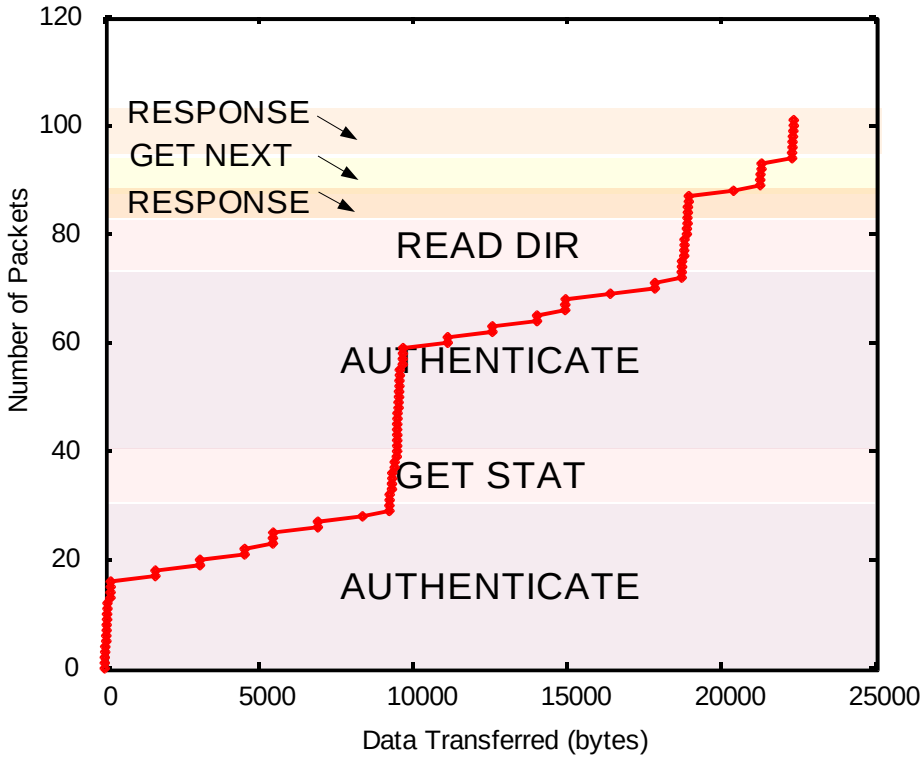


LFC Protocol Analysis

- lfc-ls command using secure catalogue
- Secure == Authenticated

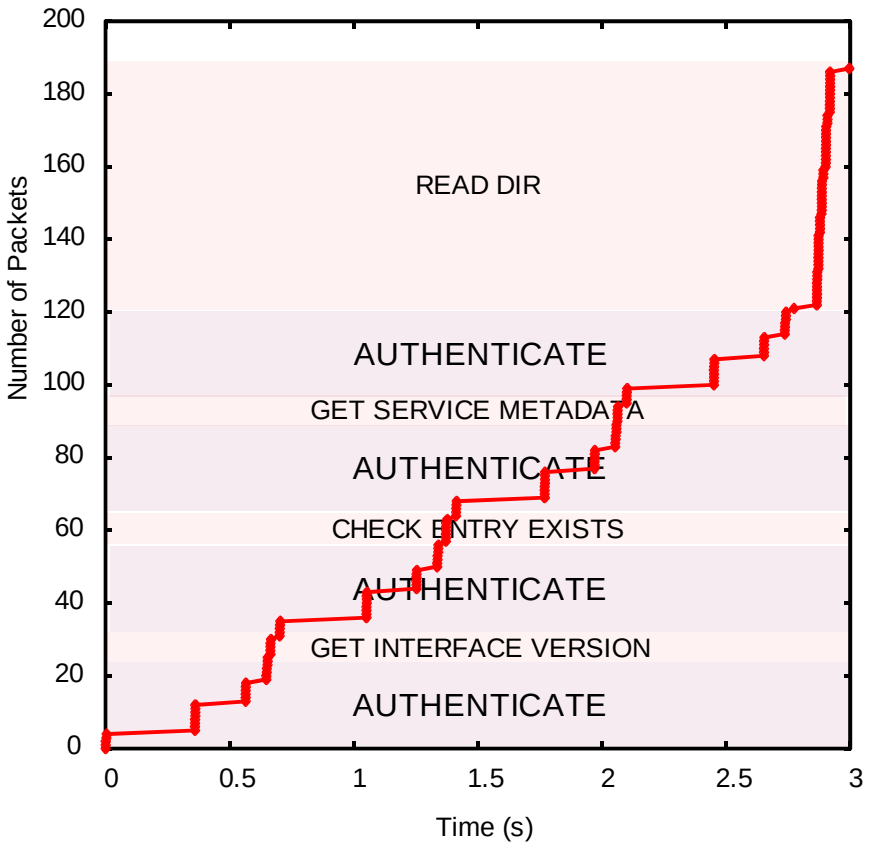
  Requests and responses still sent in clear text

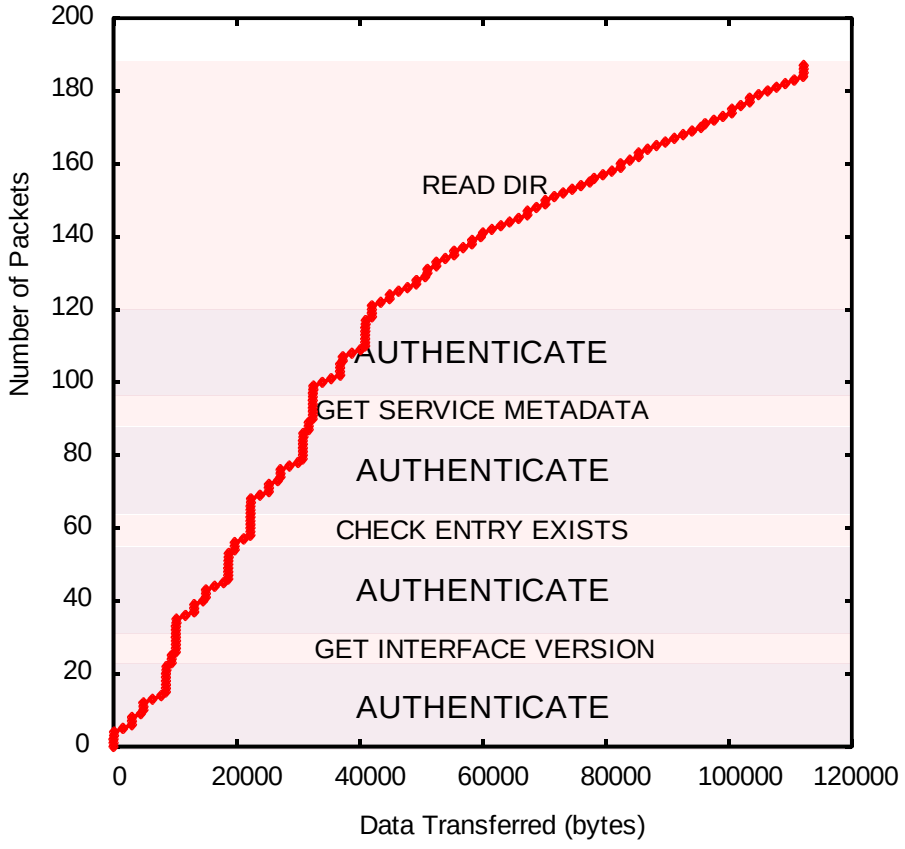




FiReMan Protocol Analysis

- glite-catalog-ls command using secure catalogue





LFC / FiReMan Protocol Comparison

- Comparison of single / multiple entry with insecure catalog

	Packets	Data (bytes)	Time (ms)
LFC 1 entry	27	110	8
FiReMan 1 entry	24	3419	170
LFC 100 entries	1309	99472	185000
FiReMan 100 entries	168	172886	760

- Main Issues:
  - LFC not encrypted
  - Both have to re-authenticate before every operation
  - LFC has to request more data from server
  - Both may have issues with multiple concurrency
    - Checking for directory which is subsequently removed
  - 1 big SOAP message can outperform many small TCP messages
    - But scaling is an issue

Further Work

- Benchmarks using Secure Catalogues
- More realistic Use Cases for tests
- For comparison compare with operations executed directly on database
  - > 80% efficiency should be possible for insecure catalogues

Conclusion

- Both LFC and FiReMan offer large improvements over EDG RLS
- Still some issues remaining:
  - Scalability of FiReMan
  - Bulk Entry for LFC
  - Inefficiencies identified in protocols
- More work needed to fully understand performance results
- Need to test with real Use Cases
  - EGEE pre-production service being set up

Questions?

Enabling Grids for E-science in Europe