# Measurement of the LCG2 and Glite File Catalogue's Performance

Craig Munro, Birger Koblitz, Nuno Santos, and Akram Khan

***Abstract*—When the Large Hadron Collider (LHC) begins operation at CERN in 2007 it will produce data in volumes never before seen. Physicists around the world will manage, distribute and analyse petabytes of this data using the middleware provided by the LHC Computing Grid. One of the critical factors in the smooth running of this system is the performance of the file catalogues which allow users to access their files with a logical filename without knowing their physical location. This paper presents a detailed study comparing the performance and respective merits and shortcomings of two of the main catalogues: the LCG File Catalogue and the gLite FiReMan catalogue.**

***Index Terms*—Catalogue, grid computing, performance.**

## I. Introduction

When the Large Hadron Collider (LHC) begins operation at CERN in 2007 it will produce petabytes of data which must be securely stored and efficiently analysed. To cope with this scale of data computing resources must also be increased. Tens of thousands of CPUs and large scale mass storage are required, more than it is feasible to accommodate at a single center. Instead the data will be distributed around the world to centers which form part of the LHC Computing Grid (LCG) [1]. Physicists will be able to access and analyse this data regardless of their geographical location using the LCG Middleware currently in development. This software provides the capability to control and execute the analysis programs while managing input and output data. Its performance and scalability is essential to guarantee the success of the experiments.

To prove this technology each experiment has performed intensive Data and Service Challenges [2] which stress these resources under realistic operating conditions. In 2004 the CMS collaboration, running at 25% of expected required capacity in 2007, discovered several issues in the Middleware. In particular the European Data Grid Replica Location Service (EDG RLS) file catalogues suffered from slow insertion and query rates which limited the performance of the entire system. These file catalogues allow users to use a human readable Logical File Name (LFN) which the catalogue will translate into the physical location of a replica of the file on the Grid.

The LCG File Catalogue (LFC) was written to replace the RLS catalogue and uses a stateful, connection-orientated approach. It has already been shown [3] to offer increased performance over the RLS catalogue. At the same time the Enabling Grids for E-sciencE (EGEE) [4] project has produced the File and Replica Management (FiReMan) catalogue as part of the gLite Middleware. Although it offers similar functionality to LFC, FiReMan is architecturally very different. It is implemented as a stateless web-service which clients contact using the Simple Object Access Protocol (SOAP) protocol.

This paper presents a comparison of the performance of the LFC and FiReMan catalogues using a variety of deployment strategies including Local and Wide Area Networks and Oracle and MySQL backends. Previous work has already been presented in [5] where insecure versions of the LFC and FiReMan catalogue were initially compared. It was shown that the LFC was faster for single operations but the ability of FiReMan to perform multiple operations in bulk, in a single SOAP message meant that it could perform more operations a second. Further analysis revealed that the LFC required a larger number of round trips to perform the same operations as FiReMan and thus performance suffered. Using the same methodology as before we now test new versions of the two catalogues which incorporate security. The range of the tests was also expanded to Wide Area Networks (WAN) as well as Local Area Networks (LAN) and the MySQL implementations as well as Oracle.

Previous studies discuss the difference in performance between TCP and SOAP implementations and between secure and insecure protocols respectively. Santos [6] demonstrates that SOAP is 2–5 times slower than an equivalent TCP implementation while Coarfa [7] estimates the overhead of SSL security degrades performance by a factor of 3.4–9.

The next section briefly discusses the main features of each catalogue. Section III presents the performance test methodology and the results of these tests with further discussion provided in Section IV. Conclusions are presented in the final section.

## II. File Catalogue Architecture

Grid Catalogues are used to store a mapping between one or more Logical File Names (LFNs), a Globally Unique Identifier (GUID) and a Physical File Name (PFN) of a replica of the file. This allows users to use the human readable LFN while the catalogue resolves the physical location. The LFC and FiReMan catalogues share many similarities. Both present a hierarchical filesystem view to users and provide an interface with commands such as `ls`, `mkdir` and `rm`. Authentication is performed using X.509 Grid Certificates and both Unix file permissions

Manuscript received January 20, 2006; revised April 7, 2006. This work was supported by the Particle Physics and Astronomy Research Council, Swindon, U.K.

C. Munro is with Brunel University, Uxbridge, Middlesex UBB 3PH, U.K. and also with CERN, 1211 Geneva 23, Switzerland (e-mail: craig.munro@cern.ch).

B. Koblitz is with CERN, 1211 Geneva 23, Switzerland.

N. Santos is with Coimbra University, Coimbra, P lo II - Pinhal de Marrocos 030–290, Portugal and also with CERN, 1211 Geneva 23, Switzerland.

A. Khan is with Brunel University, Uxbridge, Middlesex UBB 3PH, U.K.

Digital Object Identifier 10.1109/TNS.2006.877857

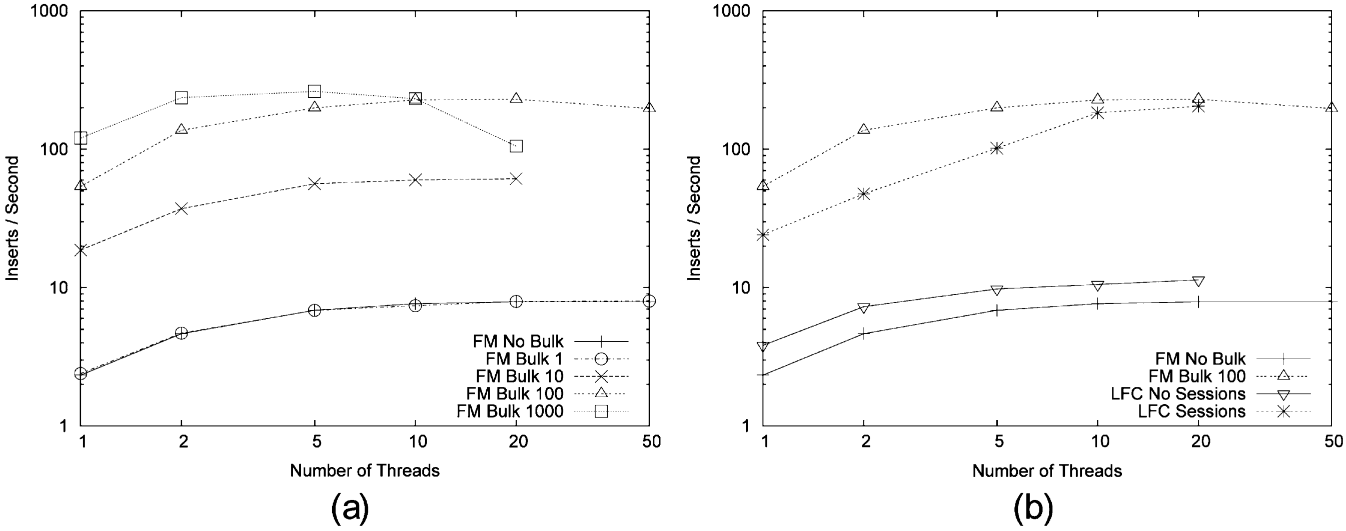

Fig. 1. (a) FiReMan (FM) insertion rate for single entry and increasing bulk sizes on a LAN using Oracle backend. (b) Comparison of FiReMan and LFC insert rate on a LAN using Oracle backend.

and POSIX Access Control Lists (ACLs) can be applied to entries. Each catalogue has an implementation using an Oracle or MySQL backend.

The differences are between the catalogues are discussed in the following sections.

*LCG File Catalogue:* The LFC is a connection-orientated, stateful server written entirely in C. A transactions API is available to start, commit or abort transactions and cursors are used within the database for handling large numbers of results. A sessions API is also available which removes the overhead of establishing an SSL connection before every operation. Aside from a function call their use is transparent to the user and allows for operations of different types to be performed within the session. LFC version 1.3.8-1sec_sl3 was used in these tests.

*Fireman:* FiReMan uses a service-orientated approach. Clients communicate via SOAP over HTTP(S) with an Axis application running within a Tomcat Application Server. The Oracle version uses stored procedures for the application logic with the Tomcat frontend only parsing the user's credentials. With MySQL all of the logic is contained within Tomcat. Multiple operations can be performed in bulk within a single SOAP message but these must be of the same type. Limited transaction support is available at the message level so that if one operation in a bulk message fails they all fail. The 2.1.6-2 and 1.4.4-1 releases were used for the Oracle and MySQL versions of FiReMan respectively with Tomcat 5-5.0.28-9.

## III. Performance Tests

Insertion and query rates of each catalogue were tested over Local and Wide Area Networks using Oracle 10 g and MySQL 4.1 backends. A multi-threaded C client was used to simulate multiple concurrent requests. Each test consisted of many (O(1000)) operations and was repeated three times to ensure accurate measurements. Prior to performing the tests one million entries were inserted into the catalogues.

For LFC all of the tests were performed after a `chdir` to the directory and without transactions, see [3] for a complete discussion of these points. The benefits of using sessions where an SSL connection is created once per test instead of operation was also examined. For FiReMan the effects of performing operations individually and with increasing bulk sizes was investigated.

In order to ensure a fair comparison between the two catalogues the same hardware was used for both LFC and FiReMan. The catalogue server and database backend shared a 2.4 GHz Dual Xeon with 1 GB of RAM. A dual 1 GHz PIII with 512 MB of RAM was used as a client for the LAN tests and a dual 3.2 GHz PIV with 2 GB of RAM was used for the WAN tests. Collocation of the catalogue server and the database should not be a problem as they are CPU and IO bound respectively. During all tests the CPU and memory consumption were monitored to ensure that the client was not limiting the overall performance of the tests. Available and used network bandwidth was also measured to ensure the network did not restrict tests. Ping round trip times were 0.3 and 315 ms for the LAN and WAN tests respectively.

Efforts were made to estimate the time required to establish an SSL connection on the server. By repeatedly performing a simple operation over ssh this was estimated to be 9 ms which places a limit of 111 new client connections every second.

### A. Oracle

The following section describes the results of the catalogue tests using the Oracle backend over a LAN and WAN.

*Local Area Network:* Fig. 1(a) shows the insert rate for the FiReMan catalogue for single entry and increasing bulk sizes. With one entry per SOAP message 2.3 inserts/s can be performed by one client, rising to 7.9 for 50 clients. It is clear that increasing the bulk entry size increases the insertion performance that can be expected from the catalogue. A maximum insert rate of 120 inserts/s with 1 client and 261 inserts/s when using a bulk size of 1000 was observed.

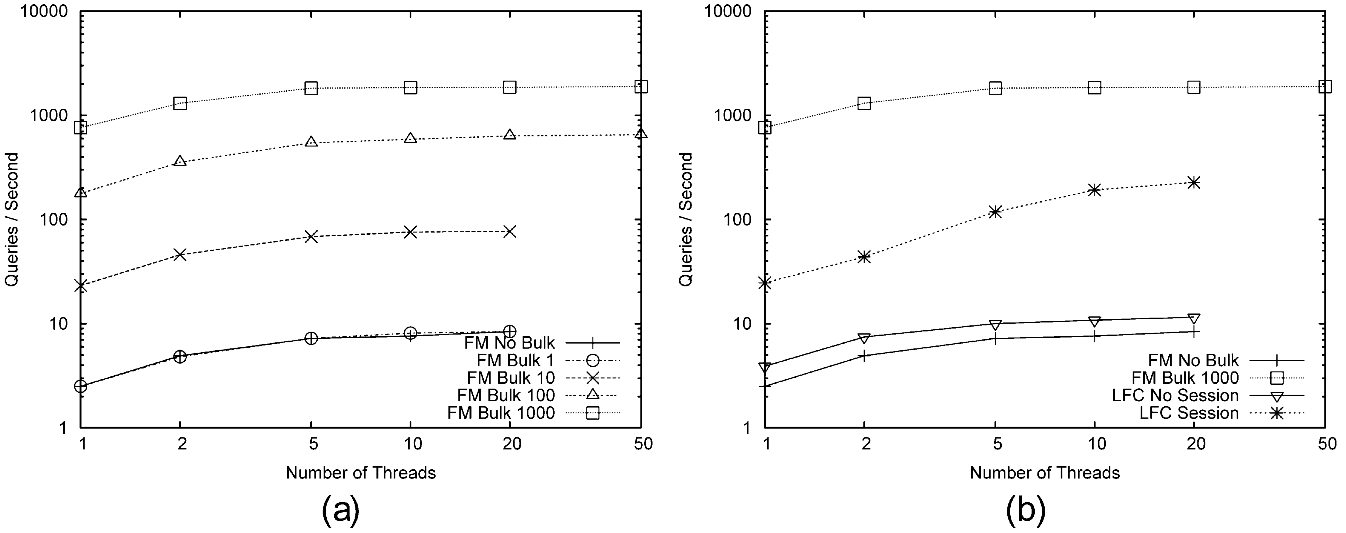

Fig. 2. (a) FiReMan query rate for single entry and increasing bulk sizes on a LAN using Oracle backend. (b) Comparison of FiReMan and LFC query rate on a LAN using Oracle backend.

A comparison between the LFC and FiReMan insert rate is shown in Fig. 1(b). For single entry the rates for both catalogues are largely similar although only FiReMan can scale to 50 clients without errors. The overhead of re-authentication before every operation becomes apparent when observing the performance advantages of using sessions with LFC and bulk operations with FiReMan. LFC goes from 3.8 inserts/s for a single client to 24.1 and from 11.4 to 204.6 for 20 clients when using sessions, a 20 fold increase in performance. A non-configurable limit of 20 threads is imposed on the LFC server which could explain why problems begin to be seen above 20 clients.

The query rate for increasing bulk sizes with FiReMan is shown in Fig. 2(a). Without bulk entries FiReMan is capable of 2.5 queries/s for a single client up to 8.4 for 20 clients. With a bulk size of 1000, nearly 1900 queries/s can be performed which is constant from 5 clients up to 50. As the test repeatedly queries for the same LFN the database should cache this result so that we can effectively observe the overhead the server introduces. As the bulk size increases we can see that the server is also able to support larger numbers of clients in parallel. This is due to the overlap of computation and communication that occurs when larger messages are sent less frequently.

Fig. 2(b) presents the comparison between FiReMan and LFC. Without sessions LFC can perform 3.9 queries/s for 1 client increasing to 11.5 with 20 clients. With sessions this rises to 24.6 and 227.0 for 1 and 20 clients respectively.

*Wide Area Network:* As Grids are by their very nature distributed it is important to evaluate the components in realistic deployment scenarios where the increased network latencies and reduced bandwidth can have a large effect. Tests were therefore conducted between a client in Taiwan and a server in Geneva with a round trip time of 315 ms.

The insert rate that FiReMan achieved with single and increasingly large bulk messages is shown in Fig. 3(a). For 1 client using single entry the insert rate is 0.52 per second increasing to 7.5 with 50 clients. The performance for single entry and with a bulk size of 1 is very similar. With a bulk size of 100 this increases to 21.1 and 109.9 for 1 and 50 clients respectively. Comparing single entry performance over a LAN and a WAN we can see that the performance of a single client over a WAN is 25% of that over a LAN. With 50 clients this figure approaches 100% due to the fact that the server is continuously busy which hides the latency of the network.

Fig. 3(b) presents the comparison between FiReMan and LFC. LFC performance increases with the number of clients up to a maximum of 20. Without sessions LFC can achieve a maximum of 7.7 inserts/s with 20 clients; with sessions this figure increases to 28.0. As discussed in [5] an operation in LFC requires several roundtrips which is especially limiting in the WAN context where every trip costs 100's of ms. Again, with an appropriately sized bulk message FiReMan is able to scale to 100 clients while LFC can support 20.

The query rate for the two catalogues is shown in Fig. 4. With single entry FiReMan can perform 0.5 queries/s with a single client increasing to a maximum of 1870 queries/s with a bulk size of 1000 and 50 clients. Without sessions LFC can query between 0.4 and 7.8 entries a second for 1 and 20 clients respectively. With sessions LFC demonstrates a 3-fold increase in performance with 1.0 and 24.7 queries/s using 1 and 20 clients.

### B. MySQL

All of the tests that have been performed so far have used the Oracle backend. As many of the sites on the Grid will choose the MySQL version it is important and relevant to also test this. This is especially interesting for the FiReMan catalogue as the Oracle and MySQL versions are completely different implementations using a common interface.

Fig. 5(a) shows a comparison of the FiReMan and LFC insert rate using the Oracle and MySQL backends on a LAN. The same implementation of LFC is used for MySQL and Oracle and, as expected, the performance of these is similar. The Oracle implementation appears to scale slightly better for larger numbers of clients. The FiReMan catalogue has an entirely different implementation for Oracle and MySQL and as could be expected the

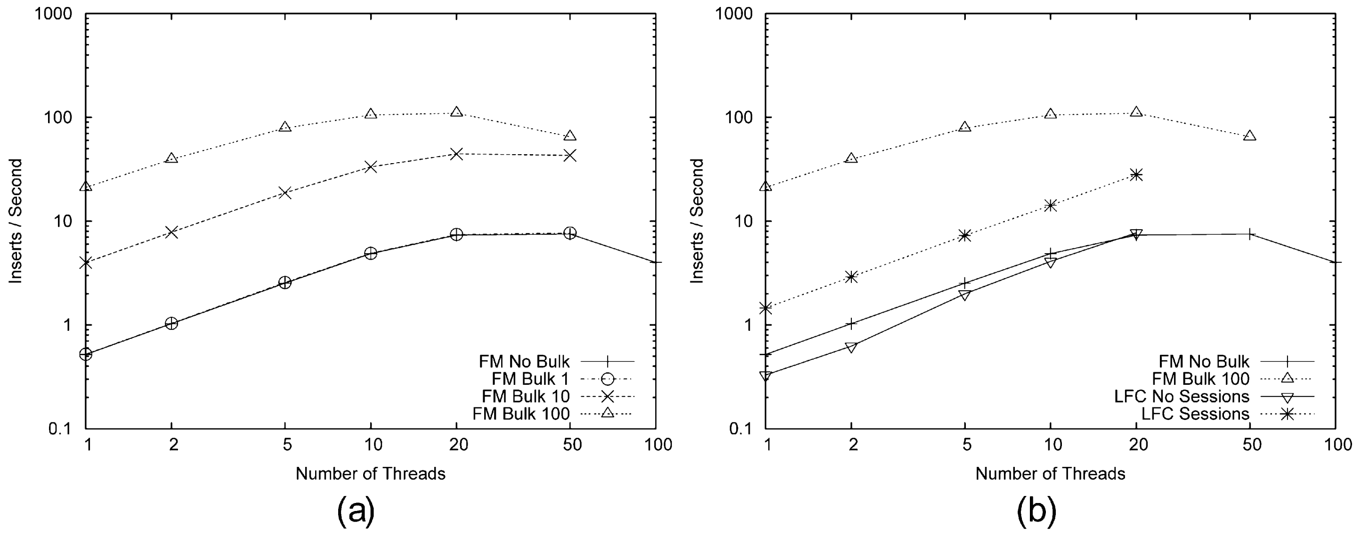

Fig. 3. (a) FiReMan insertion rate for single entry and increasing bulk sizes on a WAN using Oracle backend. (b) Comparison of FiReMan and LFC insert rate on a WAN using Oracle backend.

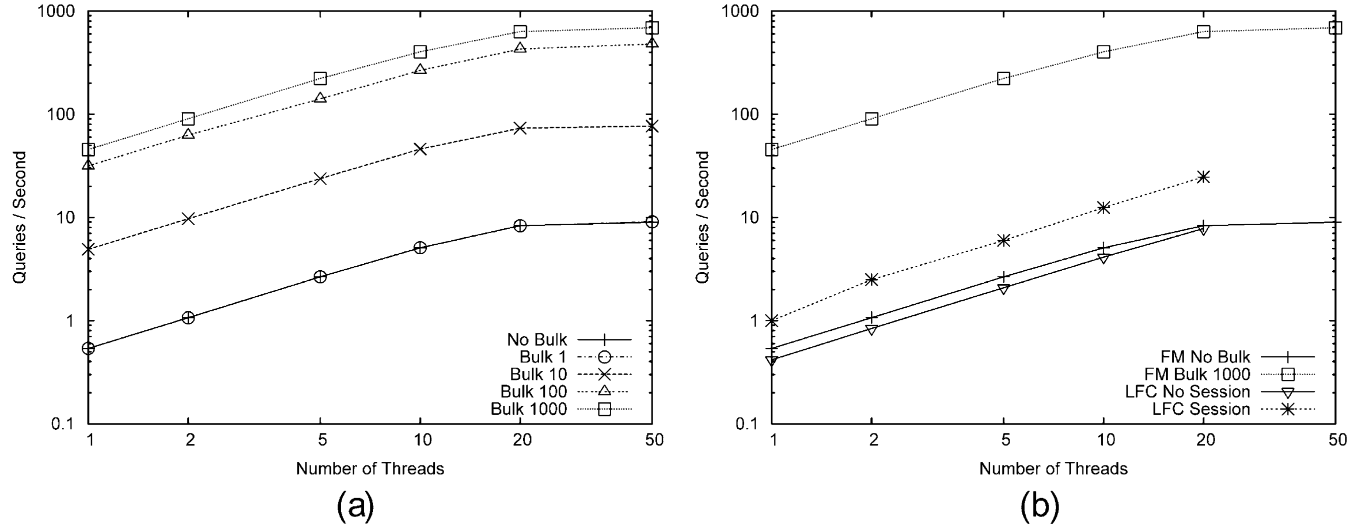

Fig. 4. (a) FiReMan query rate for single entry and increasing bulk sizes on a WAN using Oracle backend. (b) Comparison of FiReMan and LFC query rate on a WAN using Oracle backend.

two perform entirely differently. The benefits of implementing the application as stored procedures within Oracle are apparent when compared to the MySQL version. A maximum insert rate of 57 inserts/s is possible with up to 10 clients.

In Fig. 5(b) the difference between the FiReMan and LFC query rate using the Oracle and MySQL backends is illustrated. The Oracle version of FiReMan is clearly the fastest with around 1900 queries/s with the MySQL version next with a maximum query rate of 400 queries/s. The MySQL version of LFC can perform 24 queries/s with a single client up to 186 queries a second with 20 clients. Again the numbers for the MySQL and Oracle LFC are very similar.

## IV. Discussion

The catalogues that were tested are all bounded by the CPU. Establishing an SSL connection is an expensive operation both in terms of CPU time and network round trips. Any efforts to reduce this overhead are rewarded by increased performance. When FiReMan has many clients and large bulk messages, the bottleneck becomes memory consumption. With one entry being several kilobytes, memory can quickly become exhausted. The SOAP protocol of FiReMan transfers more data than the text protocol of LFC. Inserting a single entry with FiReMan transmits 16 KB of XML data but for a bulk size of 100 this is reduced by an order of magnitude to 1.9 KB an entry. A typical file insert into LFC requires 0.4 KB of data. With the measured network bandwidth of 94 Mbits/s and 1.43 Mbits/s for the LAN and the WAN respectively the network did not impose any constraints on the tests. What is important is the amount of data that has to be encrypted and, for FiReMan, parsed as XML. Reducing this overhead using bulk messages provides an obvious advantage. Collocation of the catalogue, which is CPU bound, and the database, which is IO bound did not limit the tests. Had they been on separate hosts a further latency would have been visible for small numbers of clients which would disappear as the number of clients rose.

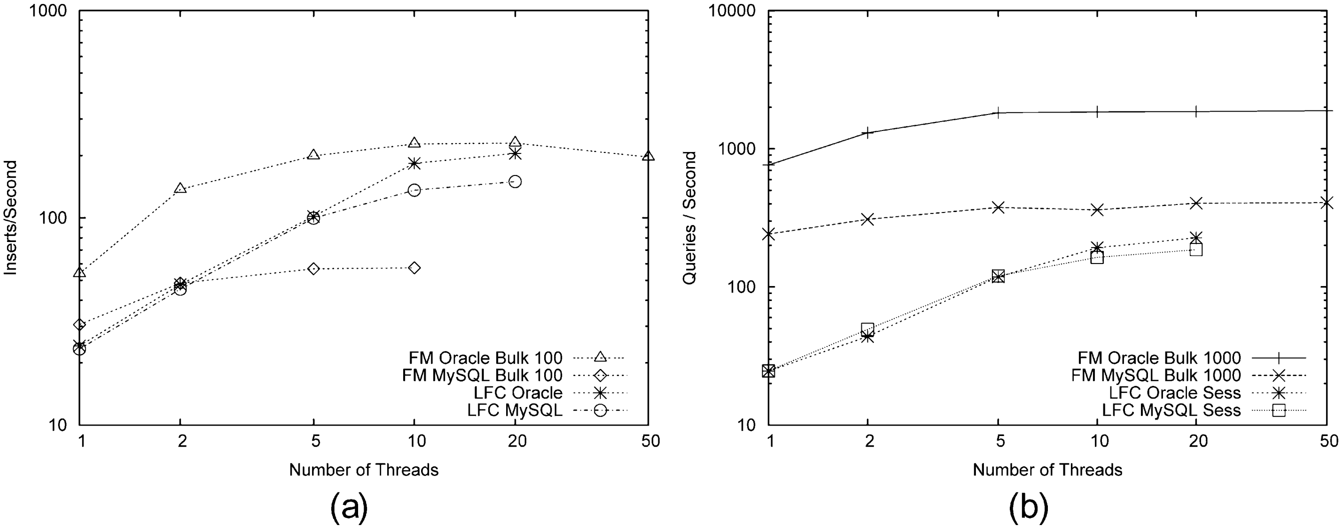

Fig. 5. (a) Comparison of the insertion rates of the FiReMan and LFC Catalogues using both the Oracle and MySQL backends on a LAN. (b) Comparison of the query rates of the FiReMan and LFC Catalogues using both the Oracle and MySQL backends on a LAN.

## V. Conclusion

This paper introduced the need for file catalogues and the importance of their performance in the context of a worldwide grid. The LFC and FiReMan catalogues provide similar functionality and both represent an improvement over the older RLS catalogues that were previously used. Architecturally the two catalogues are very different and the performance tests provide an interesting opportunity to compare a connection-orientated with a service-orientated application.

The overhead imposed by security is such that the catalogue performance for single entry is largely irrelevant. Of more importance is the methods used by the catalogues to eliminate re-authentication so that bulk operations are performant. The addition of sessions in LFC has made it possible to repeat multiple commands without re-authenticating and has the advantage that these commands do not need to be of the same type. It still suffers, particularly in Wide Area Networks, from the fact that it requires many round trips for each operation. Bulk operations in FiReMan still allow for the fastest operations with the optimum bulk size depending on the round trip time and the time taken to process each SOAP message. With large numbers of clients it is possible to balance this so that the CPU is kept busy regardless of the network speed. The constraint on this is the amount of memory available to construct these messages.

The MySQL results show that consistent performance can be expected from the LFC catalogue regardless of the backend. For FiReMan it is clear that the Oracle implementation outperforms MySQL due in part to the use of stored procedures.

## Acknowledgment

The authors wish to thank the ARDA section and the IT/GD group at CERN for all of their help and support.

## References

[1] LHC Computing Grid [Online]. Available: http://cern.ch/lcg

[2] A. Fanfani, J. Rodriguez, N. Kuropatine, and A. Anzar, "Distributed computing grid experiences in CMS DC04," in *Proc Computing in High Energy and Nuclear Physics (CHEP 04)*, Interlaken, Switzerland, 2004.

[3] J.-P. Baud, J. Casey, S. Lemaitre, and C. Nicholson, "Performance analysis of a file catalog for the LHC computing grid," in *Proc. 14th IEEE Int. Symp. High Performance Distributed Computing (HPDC-14)*, Research Triangle Park, NC, 2005, pp. 91–99.

[4] EGEE–Enabling Grids for E-sciencE [Online]. Available: http://cern.ch/egee

[5] C. Munro and B. Koblitz, "Performance comparison of the LCG-2 and gLite file catalogues," *Nucl. Instrum. Methods Phys. Res. A*, vol. A559, pp. 48–52, 2006.

[6] N. Santos and B. Koblitz, "Metadata services on the grid," *Nucl. Instrum. Methods Phys. Res. A*, vol. A559, pp. 53–56, 2006.

[7] C. Coarfa, P. Druschel, and D. Wallach, "Performance analysis of TLS web servers," in *Proc. 9th Network and Distributed System Security Symp.*, San Diego, CA, Feb. 2002, pp. 553–558.