# Performance Comparison of the LCG2 and gLite File Catalogues

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, performance, grid computing**

## 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 this end 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 EDG Replica Location Service[3] 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) in their applications which the catalogue can translate into the physical location of one of the possibly many replicas 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 [4] to offer increased performance over the RLS catalogue. At the same time the Enabling Grids for E-sciencE (EGEE) 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 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. This represents an extension of the work already presented in [5]. The main differences being the use of the SSL protocol for the client and server communication which dramatically changes the performance characteristics. Secure communication will be a required mode of operation on the Grid. In addition to the previously released study we now also include studies on the affects of Wide Area Networks and with the MySQL database backend which are both important considerations for centers on the Grid. The next section briefly discusses the main features of each catalogue. Section III presents the performance test methodology and presents the results of these tests. The final section presents the conclusions that can be drawn from this work.

## 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. The catalogue removes the need for Grid Users to be aware of the physical location of their file. They can instead use the human readable LFN.

The LFC and FiReMan catalogues share many similarities. Both present a hierarchical filesystem view to users and provide a familiar interface with commands such as `ls`, `mkdir` and `rm`. They require Grid Certificates for authentication and allow for Unix file permissions and POSIX ACLs on entries. Each catalogue has an implementation using an Oracle or MySQL backend.

The following sections highlight the differences between the two catalogues.

### *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

Craig Munro is with Brunel University and CERN. Nuno Santos is with Coimbra University and CERN. Birger Koblitz is with CERN and Akram Khan is with Brunel University.

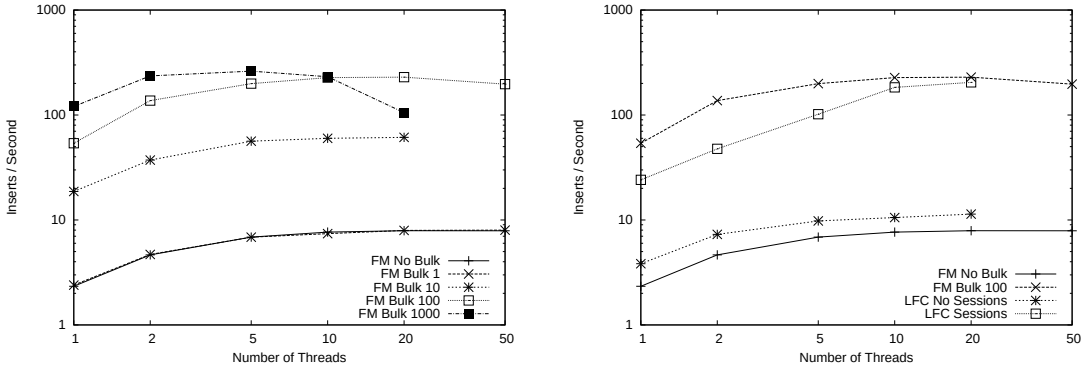



Fig. 1. (a) FiReMan insertion rate for single entry and increasing bulk sizes on a LAN using Oracle backend. (b) Comparison of FiReMan and LFC insert rate on a LAN using Oracle backend.

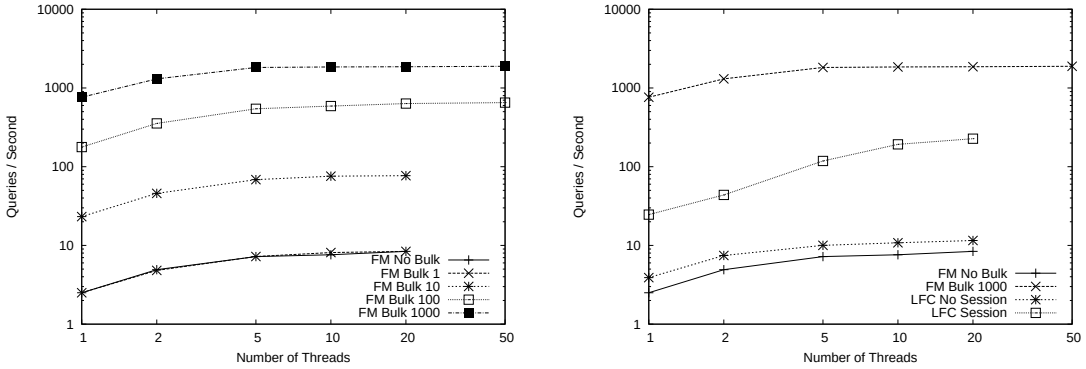



Fig. 2. (a) FiReMan query rate for single entry and increasing bulk sizes on a LAN using Oracle backend. (b) Comparison of FiReMan and LFC query rate on a LAN using Oracle backend.

for handling large numbers of results. A sessions API is also available which removes the costly overhead of establishing an SSL connection before every operation.

*FiReMan*

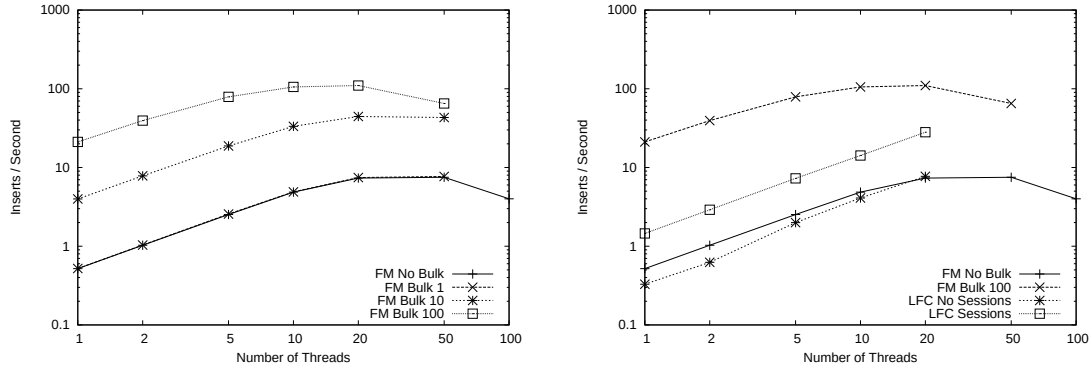
FiReMan uses a service-orientated approach. Clients communicate via SOAP over HTTP(S) with an Axis application running within a Tomcat Application Server. The Oracle version uses stored procedures for the application logic with the Tomcat frontend only parsing the user's credentials. With MySQL all of the logic is contained within Tomcat. Bulk operations are supported so that multiple operations of the same type can be performed within a single SOAP message. Limited transaction support is available at the message level so that if one operation in a bulk message fails they all fail.

## III. Performance Tests

Insertion and query rates of each catalogue were tested over Local and Wide Area Networks using Oracle and MySQL backends. A multi-threaded C client was used to simulate multiple concurrent requests. Each test consisted of many operations and was repeated multiple times to ensure accurate measurements. Before performing the tests 1 million entries were inserted into the catalogue.

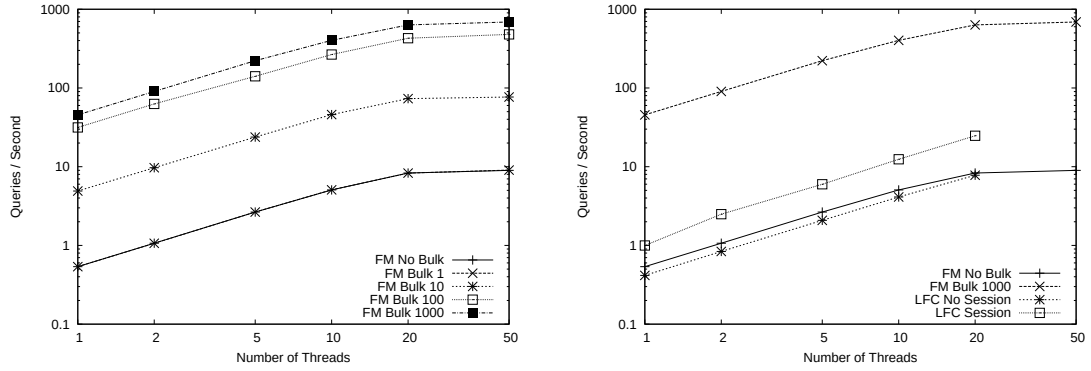
For LFC all of the tests were performed after a `chdir` to the directory and without transactions, see [4] for a complete discussion of these points. The effects of performing sessions where an SSL connection is created once per test instead of operation was also investigated. For FiReMan the effects 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 2.4GHz Dual Xeon with 1GB 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 2GB of RAM was used for the WAN tests. During all tests the CPU and memory consumption were monitored to ensure that the client was not limiting the overall performance of the tests. Round trip times were 0.3 and 315ms for the LAN and WAN tests respectively.




(a) (b)

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.




(a) (b)

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.

### *A. Oracle*

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

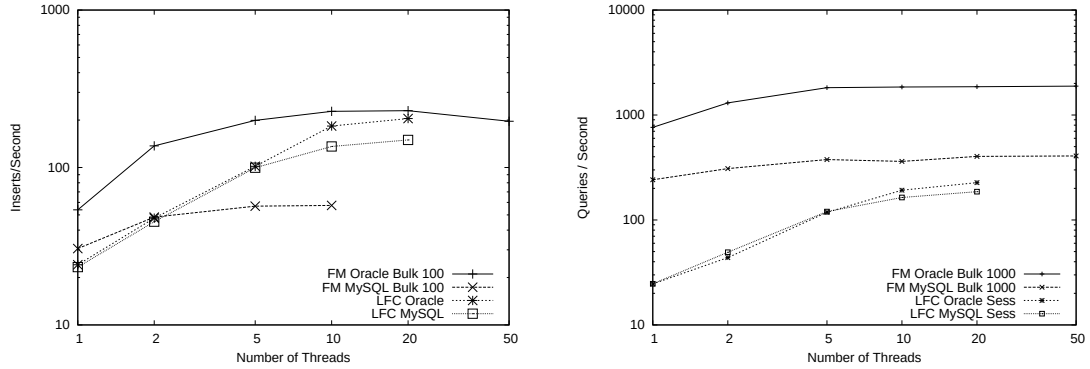
*Local Area Network:* Figure 1(a) shows the insert rate for the FiReMan catalogue for single entry and increasing bulk sizes. With one entry per SOAP message 2.3 inserts/s can be performed by one client, up to 7.9 for 50 clients. It is clear that increasing the bulk entry size increases the insertion 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. For many clients and large bulk messages, the bottleneck becomes memory consumption. With one entry being several kilobytes, memory can quickly become exhausted.

A comparison between the LFC and FiReMan insert rate is shown in Figure 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 doing bulk operations with FiReMan and sessions with LFC. 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. By default the LFC server runs with 20 threads which could explain why problems begin to be seen above 20 clients.

The query rate for increasing bulk sizes with FiReMan is shown in Figure 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 a better overlap of computation and communication that occurs when sending less larger messages.

Figure 2(b) presents the comparison between FiReMan and LFC. Without sessions LFC can perform 3.9 queries/s, up to 11.5 with 20 clients, increasing to 24.6 and 227.0 respectively with sessions.

*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 can have a large effect. Therefore tests conducted between a




(a) (b)

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.

client in Taiwan and a server in Geneva connected by a network with a round trip time of 315ms.

The insert rate that FiReMan achieved with single and increasingly large bulk messages is shown in Figure 3(a). For 1 client using single entry the insert rate is 0.52 per second up to 7.5 with 50 clients. The performance for single entry and with a bulk size of 1 is virtually identical. With a bulk size of 100 this increases to 21.1 and 109.9 for 1 and 50 clients respectively. These figures represent a reduction in performance of between 25% and almost 100%. These high rates are possible using large numbers of clients as the server is continuously busy the round trip time ceases to matter.

Figure 3(b) presents the comparison between FiReMan and LFC. The performance of LFC with and without session increases linearly from 1 to 20 clients. LFC can achieve a maximum of 7.7 inserts/s with 20 clients and without sessions and up to 28.0 with 20 clients and sessions. 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 Figure 4. With single entry FiReMan can perform 0.5 queries/s with a single client and up to a maximum of 1870 queries/s with a bulk size of 1000 and 50 clients. Without a session LFC can query for between 0.4 and 7.8 entries a second for 1 and 20 clients respectively. With sessions 1.0 and 24.7 queries/s with 1 and 20 clients are possible. By creating a session and eliminating the need to re-establish a secure connection LFC you can observe this 3-fold increase in performance. As LFC can maintain state it also has the advantage that any operation can be performed within this session.

### *B. MySQL*

While all of the tests that have been performed so far used the Oracle backed it is important and relevant to also test the MySQL backend as this is the version that many of the sites will or would deploy. This is especially interesting for the FiReMan catalogue as the Oracle and MySQL versions are completely different implementations using a common interface.

Figure 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 with the Oracle implementation being faster as the number of clients grows. The FiReMan catalogue has an entirely different implementation for Oracle and MySQL and as such the two perform entirely differently. The Oracle version benefits from using Oracle stored procedures while with Tomcat all of the logic is implemented within Tomcat. A maximum insert rate of 57 inserts/s is possible with up to 10 clients.

In Figure 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 are very similar to the Oracle version.

## IV. 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.

With the inclusion of security the performance of the two catalogues while doing single operations has become virtually identical. 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 which uses stored procedures outperforms the MySQL implementation where all of the logic is contained within Tomcat.

## ACKNOWLEDGEMENTS

The authors wish to thank the ARDA section and the IT/GD group at CERN for all of their help and support. Craig Munro would also like to thank the Particle Physics and Astronomy Research Council, Swindon, UK for his funding.

## REFERENCES

[1] LHC Computing Grid. `http://cern.ch/lcg`.
[2] A. Fanfani, J. Rodriguez, N. Kuropatine, and A. Anzar. Distributed computing grid experiences in CMS DC04. In *CHEP 04*, September 2004.
[3] European Data Grid. `http://cern.ch/edg-wp2/`.
[4] Jean-Philippe Baud, James Casey, Sophie Lemaitre, and Caitriana Nicholson. Performance analysis of a file catalog for the LHC computing grid. In *HPDC 14*, 2005.
[5] Craig Munro and Birger Koblitz. Performance comparison of the LCG-2 and gLite file catalogues. In *ACAT 05*, May 2005.