Scalability and Performance Analysis of the EGEE Information System

F. Ehm
CERN

E-mail: Felix.Ehm@cern.ch

L. Field
CERN

E-mail: Laurence.Field@cern.ch

M. W. Schulz
CERN

E-mail: Markus.Schulz@cern.ch

Abstract. Grid information systems are mission-critical components in today's production grid infrastructures. They provide detailed information about grid services which is needed for job submission, data management and general monitoring of the grid. As the number of services within these infrastructures continues to grow, it must be understood if the current information system used in EGEE has the capacity to handle the extra load. This paper describes the current usage of the EGEE information system obtained by monitoring the existing system. A test framework is described which simulates the existing usage patterns and can be used to measure the performance of information systems. The framework is then used to conduct tests on the existing EGEE information system components to evaluate various performance enhancements. Finally, the framework is used to simulate the performance of the information system if the existing grid would double in size.

1. Introduction

Universities and research institutes are real organizations that have computing centres and are ultimately responsible for their resources. Scientists from different institutions collaborate on specific research topics and they would like to use the resources available to them. The difficulty faced by the scientist is that each organization where the resources are located have different, security infrastructures, policies and systems. Grid computing is seen as the solution to this problem of

heterogeneity by making the resources available via a common “Grid Services”. Various projects have built on this concept to create multi-institutional infrastructures for e-Science [1].

The information system is a critical piece of any grid architecture. It enables the user and other services to discover what resources are available in the grid and further details about them. Over recent years a number of grid projects have emerged which have built grid infrastructures that are now the computing backbones for various user communities. Although there is some diversity in the current grid information systems with respect to implementation and data models, there is a great deal of commonality between the core functionality and deployment scenarios [2].

At the resource level there is some entity which obtains information about the resource. This entity is usually known as an information provider. An interface is required to enable the information from the information provider to be queried. In many deployment scenarios, an entity exists at the organisational boundary which provides information about all resources for which the organisation is responsible. This is usually realised by querying all the resource level interfaces and caching the information. An interface is required to enable the information in the cache to be queried. At the grid level there is some entity which caches the information about resources from all the organisations. This is usually realised by querying all the site level interfaces and caching the result. An interface is required to enable the information in the cache to be queried.

As the functionality of each level is essentially the same, the building block can be used at the different levels of the system. The core part of this building block is the query interface. As the interface hides the implementation of the cache, this building block can be represented by an endpoint to which a query can be sent. The main difference between the levels is the amount of information which is returned for the query. In this respect, the most important aspect of the interface is the performance when returning large query results as the top level the query interface will have to return information about many services. As grid infrastructures grow more services are present in the infrastructure, which results in even larger query results. In addition, more services also results in more queries to the query interface. As scalability is the primary issue for information systems, the performance of the information system needs to be understood and evaluated against the estimated future requirements.

The grid information system performance needs to be benchmarked using the current scale of the existing infrastructure. From this it should be possible to determine how the system will cope by extrapolating the results. To be sure that the system will meet the future requirements, a simulation needs to be conducted with the projected future size of the grid.

2. Testing Methodology

If an information system component consists of a cache with a query interface, there are two main functions which need to be tested for scalability, updating the cache and querying the cache via the interface. The results and conclusion of scalability investigations related to updating the cache are available from previous work [5]. This work will focus on the performance of the querying interface.

An information system interface can be represented by an endpoint to which a query can be sent and the query result is returned. In this scenario the performance of the interface can be measured by evaluating the query response time. This metric can be affected by number of factors;

- The total data size over which the query must be executed.
- The complexity of the query.
- The data size of the query results.

- The loading due to other parallel queries.
- The implementation

A simple testing framework was designed to measure the response time of queries acting upon an information system endpoint. The queries used should be examples of actual queries seen on the infrastructure, see table 1. To enable realistic loads to be simulated, many queries must be executed in parallel and hence multiple query hosts are required to generate such load. The framework is configurable so that number of hosts and parallel queries per host can be controlled. The framework queries act upon an information system endpoint which is required to be setup and populated with data representing the real grid infrastructure. All query hosts and the machine running the information system endpoint, were instrumented to monitor standard metrics such as CPU load, memory usage etc. using the *Lemon* [3] monitoring tools. The framework is designed in such a way that it could be used against different information system endpoints. The query clients for the endpoint can be added as a plug-in to the test framework which will run the query client. In addition it is possible to define the query to be used. This is important as the query used will alter the data size of the query result and hence will affect the response time. A master script copies the necessary files to the query hosts and initiates the query client. This client executes *n* times the query as a background process. When the query command has completed, the response time is recorded for that query and another one is executed. This mechanism ensures a constant number of requests are running in parallel.

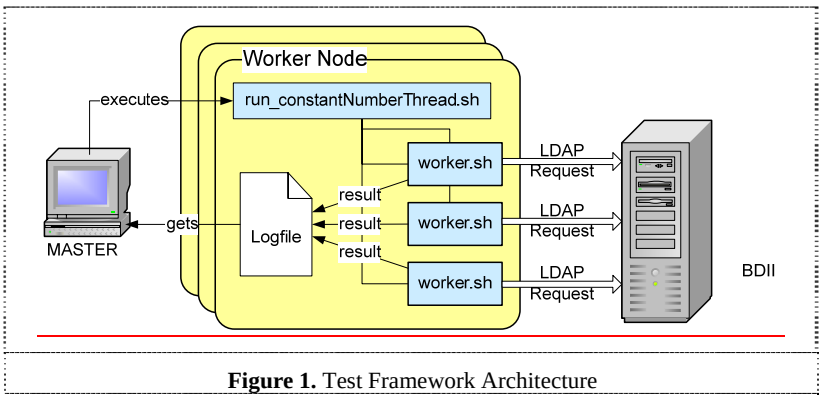

Figure 1. Test Framework Architecture

This process of starting new queries runs for a given amount of time after which queries will no longer be initiated and query client will exit. When all the clients have finished the master script copies the results from the query hosts, process the results and displays a summary of the test. Only the results in the middle of the test are used to ensure that the values represent a steady state at that load. It is important to record the number of failed queries and the number of queries that timed-out. These are indications that the scalability limit has been reached.

3. Benchmarking

In order to benchmark the existing system, the current usage of the existing system needed to be understood. As of July 2007, the EGEE [6] information system contained information for 251 sites which provided 1428 Services. This information represents a data size of 28 Mbytes. The EGEE information system is built using the Berkley Database Information Index (BDII) as the cache/query interface component [5]. The top level BDII is deployed as a load balanced service. Network monitoring of the service (see table 1) showed that it handles around 2 million connections per day. To understand the queries and usage patterns, the log file of one host in the load balanced service was

monitored over a four hour period. This time window should be sufficient to observe a stead state load and iron out any statistical spikes due to a single bulk operation. The log files were then analyzed to find the queries used and their frequency. This information can be used to simulate a realistic query load on and information system endpoint.

.

Table 1. Common Queries.

Number of Queries	Query Type
6075	Find the Close CE to an SE
5475	Find the VOs SA for an SE
5043	Find all SRMs
4791	Find an SE
2432	Find the Close SE to a CE
2117	Find all Services for a VO
664	Find all CEs for a VO
638	Find all SAs for a VO
479	Find all SubClusters
448	Find the GlueVOView for a CE

3.1. Response time for various queries

A top level BDII was installed and configured using *OpenLDAP* [4] version 2.0.27 on an Dual Intel(R) Xeon(TM) CPU 2.40GHz with 1GB of RAM. The BDII was populated with information from the production infrastructure. The cache update mechanism was disabled to avoid it affecting any measurements. Using the information from the usage patterns, queries were categorised by data size. The test framework was used to execute the queries for the various data sizes and with different query loads. The results are shown in figure 2. The results show that the response time increases linearly with the query load. The query result size does effect the response time but not in any significant way. This suggests that for query results smaller than 1 Mb, most of the time is due to the connection overhead and database operation.

4. Performances Enhancements

A number of performance enhancements were suggested to improve the response time and hence scalability of the top level BDII. Each improvement was tested using the test framework to evaluate the effect.

4.1. The effect of indexing on the response time.

The analysis of the log files highlighted the most frequent queries. This information was used to create indexes in the OpenLDAP database. Tests were run to measure the response time with and without indexing. Figure 3 shows the result for a typical query. The results show that indexing the database improves the response time by nearly two orders-of-magnitude.

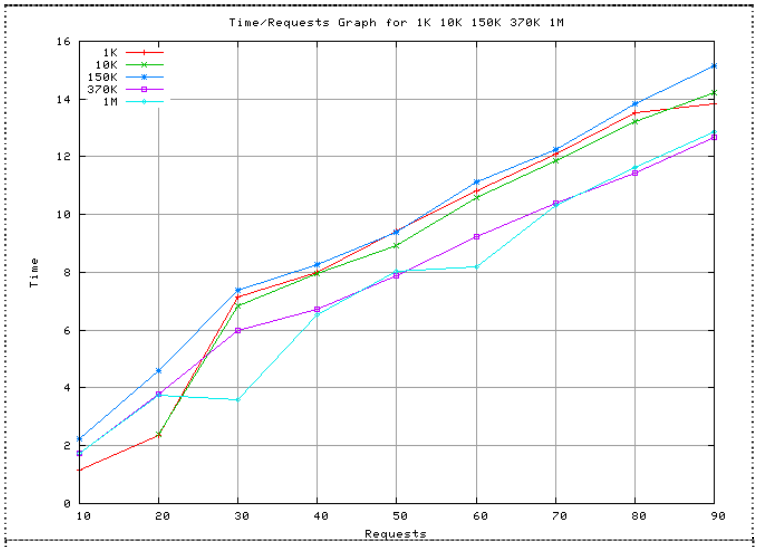

Figure 2. Query response time for different query loads and query result sizes.

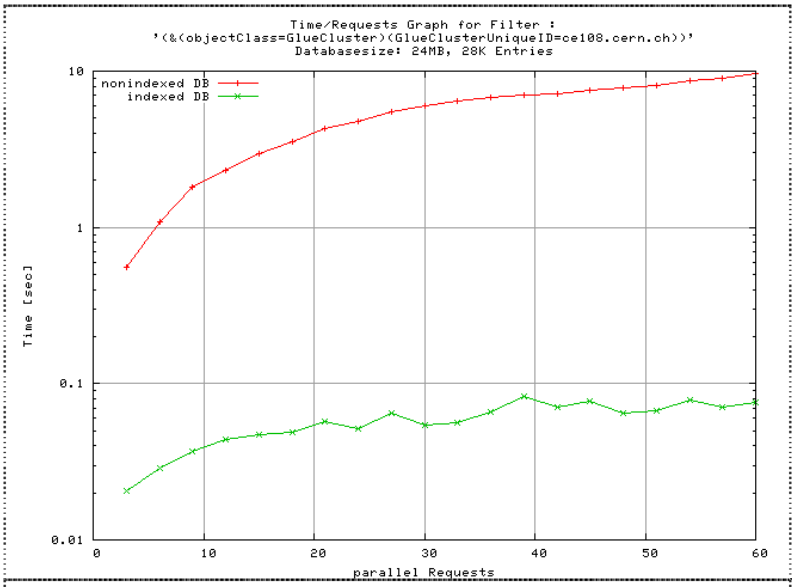

Figure 3. Query response time for different query loads showing he difference between indexed and no-indexed databases. (log scale!)

4.2. The effect of the Open LDAP version on the response time.

The reference platform for the EGEE infrastructure is Scientific Linux [7]. Scientific Linux version 3 by default provides OpenLDAP version 2.0.27, whereas Scientific Linux version 4 provides OpenLDAP version is 2.2.13. Tests were run to see the impact on performance due to the new version of OpenLDAP. The results are shown in Figure 4.

4.3. The impact of hardware on the response time.

The main scalability limitation of the BDII is due to the number of queries that can be processed in a given amount of time. As each query is orthogonal, the move to multi-core processors should give a significant advantage. Tests were run to see this effect. The following hardware was used in the test; Dual Intel dual core, 2.8 GHz with 1GB RAM and Dual Intel quad core 2.8 GHz with 2GB RAM. The results are shown in Figure 4.

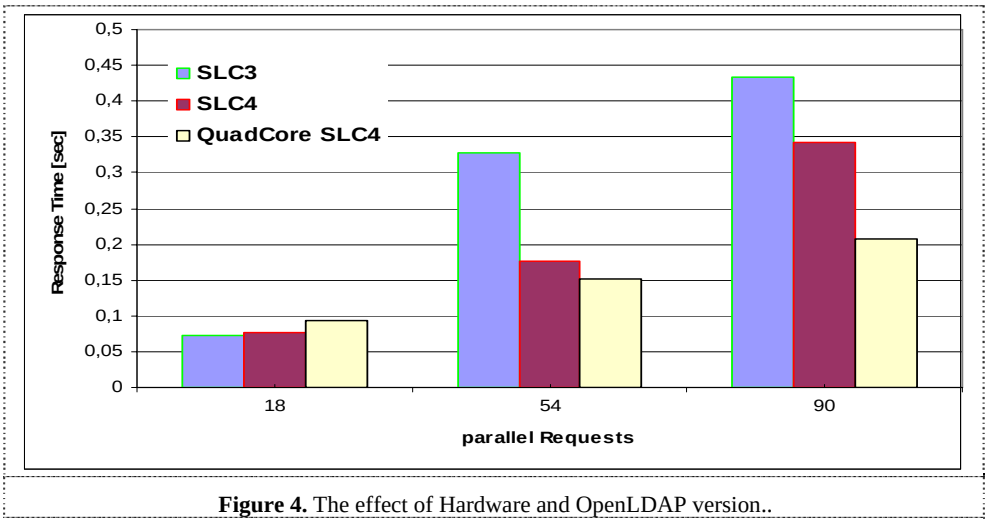

Figure 4. The effect of Hardware and OpenLDAP version..

Figure 4 shows that both version 2.2.13 and more cores both improve the response time. With 54 parallel requests, the performance improvements with the 2.2.13 version can clearly be seen. As there is a small difference between the 2.2.13 version with Dual and Quad cores suggests that the server is not loaded and hence more cores do not make much of a difference. With 90 parallel requrests, the server is loaded so multiple cores do help to reduce the response time.

5. Extrapolation

The use of indexing, multi-core processors and the new OpenLDAP version have significantly improved the performance of the BDII. A final test was conducted to see how the BDII would

perform if the existing EGEE infrastructure doubled in size. This would case the amount of data in the information to double in size as well as the number of queries.

5.1. The effect of doubling the data size.

In order to conduct this test the BDII needed to be populated with double the amount of data. To do this the information from the production system was added twice but the LDAP DNs were modified so that the entries would be unique and not duplicates. The test was conducted using an indexed BDII on a Dual Intel Quad Core 2.8 GHz with 2GB RAM and OpenLDAP version is 2.2.13. There results are shown in Figure 5.

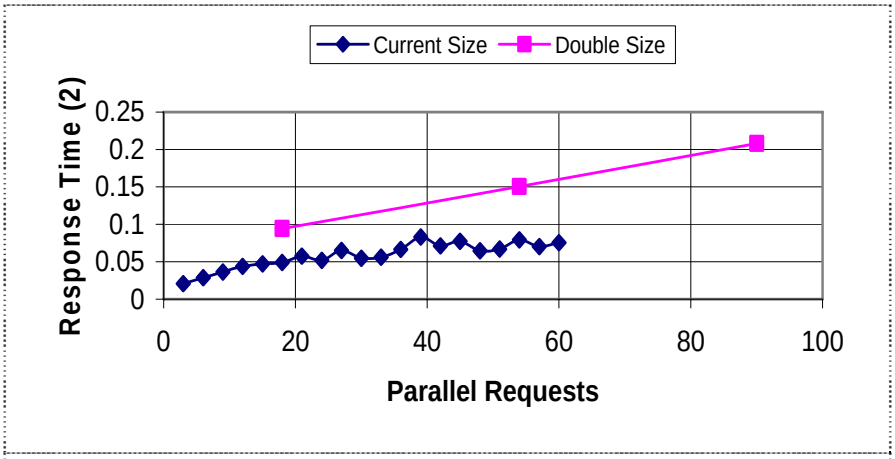

Figure 5. Query response time for different query loads showing the current infrastructure size and double size

The top level BDII service on the EGEE infrastructure is handling 2 Million queries per day which represents a query rate of 23Hz. Doubling the size of the infrastructure and hence doubling the amount of queries would increase the query rate to 46Hz. From Figure 5 it can be seen that for 80 parallel requests the response time is 0.2s which corresponds to a query rate of 400Hz. This query rate is an order-of-magnitude greater than what we would expect to see in the near future on the production infrastructure.

6. Conclusion

A testing framework was developed in order to evaluate the performance of the existing EGEE information system. This framework was created in extensible way so that it could be used to evaluate any information system endpoint. The usage patterns of the EGEE infrastructure were investigated to ensure that the framework would be able to perform a realistic simulation of the load seen in the production infrastructure.

The existing top level information system component, the top level BDII was tested using the framework as a point of reference. Following on from this a number of suggestions for improving the scalability performance of the BDII was evaluated. Each suggestion did indeed give a significant

performance improvement. All the suggestions combined result in a two orders-of-magnitude improvement to the scalability and performance of the BDII.

A final test was done to see if these performance enhancements will enable the BDII to meet the future requirements of the EGEE production infrastructure. The tests show that the current information system will indeed be able to handle a doubling in size of the infrastructure.

In terms of future work, it has to be understood how the OpenLDAP technology at the core of the BDII compares to the performance of other such technologies. The framework developed can be re-used to evaluate other technologies and information system components.

References

[1] L.Field, M. W. Schulz, *Grid Interoperability: The Interoperations Cookbook* Proc of CHEP 2007

[2] L.Field, M. W. Schulz, *Grid Deployment Experiences: The Evolution of the LCG Information System* Proc of CHEP 2006

[3] Lemon Monitoring Tool - http://www.cern.ch/lemon

~~[3]~~[4] OpenLDAP - http://www.openldap.org

~~[4]~~[5] L.Field, M. W. Schulz, *Grid Deployment Experiences: ThePath to a Production Quality LDAP Based Information System* Proc of CHEP 2004

~~[5]~~[6] EGEE Infrastructure - http://www.eu-egee.org/

~~[6]~~[7] Scientific Linux - http://www.scientificlinux.org/

Field Code Changed