Measuring the Usability of Triple Stores for Knowledge Management on Trauma Care Organizations

Joseph Utecht and Mathias Brochhausen PhD

Department of Biomedical Informatics, University of Arkansas for Medical Sciences, Little Rock, AR

The CAFE project[1] aims to provide a semantic web technology-based approach to compare the organizational structures of trauma centers and trauma systems. To achieve this we plan to use an RDF triple store that employs automatic inferences based on OWL representations. In order to engage users with the CAFE application real-time feedback is a requirement. Research has shown that even small delays in a website while attempting to perform some task will greatly decrease the rate at which people complete said task [1]. Although many RDF triple store performance measures have been published, there appears to be a gap when it comes to their use as the primary storage for real-time applications. The performance needs for this use case differ from the triple stores more studied use of offline reasoning and inference over large data sets. The objective of this research is to determine the feasibility of modern RDF triple stores as the primary storage for a real-time application.

We decided to focus on Apache Jena, Blazegraph, and Sesame as the RDF stores for our testing due to their support for RDFS reasoning, open source licenses, ability to handle large datasets, and REST endpoints for interaction. The tests were also run over trial versions of AllegroGraph and Stardog for comparisons to commercial products. We used Lehigh University Benchmark (LUBM) [2] generated data to test performance and capability of the triple stores. We did not use the default testing queries with this dataset, as they were more designed to measure the performance of OWL inference models; instead a new series of queries with increasing complexity were used for the test. To measure query performance the queries were run 1,000 times with different parameters from a benchmarking program written in Python that measured the time until the HTTP request was returned with the query results.

To have a baseline to measure the performance of the RDF triple stores we decided to use a relational database performing close to the same task. To accomplish this we converted a subset of the LUBM data into a relational format and loaded it into MariaDB. A miniature HTTP REST interface was created which would take a key as an HTTP parameter and then run a SQL query to the database returning the results as JSON encoded data. As we were not interested in attempting a comparison between the relative run times of SQL vs SPARQL queries, only two simple SQL queries were used to determine the minimum time this relational database with a REST interface would return data.

[1] `http://cafe-trauma.com/`

Query performance was highly volatile depending on the query and store being tested. At their fastest the RDF stores could return data within the same range as the optimized relational database, however the wrong query could see response times anywhere from 20 to 500x slower. Most interesting was that poor performance on queries was not uniform across stores and thus appears to be heavily influenced by their storage model and query optimization. A separate average is given excluding a query with the *optional* SPARQL keyword as this produced very poor results on some stores in comparison to other queries.

Table 1. Query Runtimes

Store	Average	Average w/o Optional	Minimum	Maximum
Relational	6	n/a	5	6
Jena	1251	461	7	4413
Blazegraph	37	37	36	38
Sesame	58	8	7	253
AllegroGraph	7	7	7	8
Stardog	n/a	22	10	56

Measured in milliseconds

Jena's performance was highly volatile based on the query. The best speeds were within the same range of the relational system as expected from the low number of indexes. The *optional* keyword which is known to cause potential slowdowns takes a heavy toll on Jena resulting in the slowest query of the entire test. Blazegraph's large number of indexes and well established query optimizer result in extremely stable query time. There however appears to be a 30ms processing time on anything Blazegraph is doing, of which we were unable to locate the cause. Stardog performed reasonably well, except that it was unable to return the *optional* query before it timed out after 5 minutes. Sesame and AllegroGraph showed great performance overall being within the same range as the relational system on most queries. However Sesame, while not as slow as Jena, also had problems with the *optional* keyword.

After examining the performance of RDF triple stores for query throughput in a real-time application and comparing it to the performance of a relational database, we found that performance in Sesame and AllegroGraph was consistently within the same range of the optimized relational database. Based on this performance we will move forward with our plans to use an RDF triple store as the primary storage for the real-time application in the CAFE project.

Acknowledgement Research reported in this publication was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number 1R01GM111324.

References

1. D. F. Galletta, R. M. Henry, S. Mccoy, and P. Polak, *Web site delays: How tolerant are users?* PhD thesis, 2002.
2. Y. Guo, Z. Pan, and J. Heflin, "LUBM: A benchmark for OWL knowledge base systems," *Web Semantics*, vol. 3, no. 2-3, pp. 158–182, 2005.