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

Joseph Utecht and Mathias Brochhausen Ph.D.
Department of Biomedical Informatics, University of Arkansas for Medical Sciences, Little Rock, AR

UAMS

Abstract

The Comparative Assessment Framework for Environments of Trauma Care (CAFE) project aims to provide a semantic web technology-based approach to compare the organizational structures of trauma centers and trauma systems. To achieve that CAFE will provide a web service that displays graphical representations of organizational structures including comparative annotations to the representatives of trauma care environments. To ensure comparability, the data about the organizational structure of different trauma centers and trauma systems will be stored in an RDF triple store, which employs automatic inferences based on OWL representations to achieve semantic integration. In order to engage users with the CAFE application, real-time feedback is a requirement [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 store's more traditional 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. To make this determination we measured the load time of just over one million triples of synthetic data and then ran various queries to emulate the types of demands this use would create. The resulting time measurements showed a large difference in the range of results in query performance. The lower end results are fast enough to compete with traditional relational databases while the results on the high end would not be suitable for this use case. The conclusion of this for the CAFE project is that we will be able to use a triple store, but we need to use the performance implications gained from this study to carefully tailor the queries we will run.

Method

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 [2]. The tests were also run over trial versions of AllegroGraph and Stardog for comparisons to commercial products. We used Lehigh University Benchmark (LUBM) [3] 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, through the HTTP REST interface, 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. Testing was performed on a VirtualBox VM with 8GB of ram, running CentOS 7 and the application server Tomcat when needed. The virtual machine was used to attempt to prevent external factors from influencing results.

To have a benchmark 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 on 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 only two simple SQL queries were used to determine the minimum time this relational database with a REST interface would return data.

The testing queries were designed to emulate the small queries often seen in this type of application. At the same time they were also attempting to test reasoning and any query optimization that the stores might be performing. The first query is a baseline of how fast the store can return data, as this is just returning data out of an index. The second query is a test of the optional keyword, which is known to cause problems in some stores. The third query returns multiple matches to see what effect that might have. The fourth query performs a join between two matches. The final query is slightly more complex than the fourth but also requires that both ub:Faculty and ub:Student be reasoned as the actual triples are only ever subclasses of those two. In the queries STUDENT, PROFESSOR and PAPER were replaced with different known URIs each query.

Queries

Baseline

```
PREFIX ub: <http://swat.cse.lehigh.edu/onto/univ-bench.owl#>
SELECT ?name
WHERE {
        <STUDENT> ub:name ?name .
      }
```

Optional Keyword

```
PREFIX ub: <http://swat.cse.lehigh.edu/onto/univ-bench.owl#>
SELECT ?name ?advisor ?email ?telephone
WHERE {
        optional {?x ub:name ?name .}
        optional {?x ub:advisor ?advisor .}
        optional {?x ub:email ?email .}
        optional {?x ub:telephone ?telephone .}
        values ?x { <STUDENT> }
      }
```

Multiple Value

```
PREFIX ub: <http://swat.cse.lehigh.edu/onto/univ-bench.owl#>
SELECT ?author
WHERE {
        ?author ub:publicationAuthor <PAPER> .
      }
```

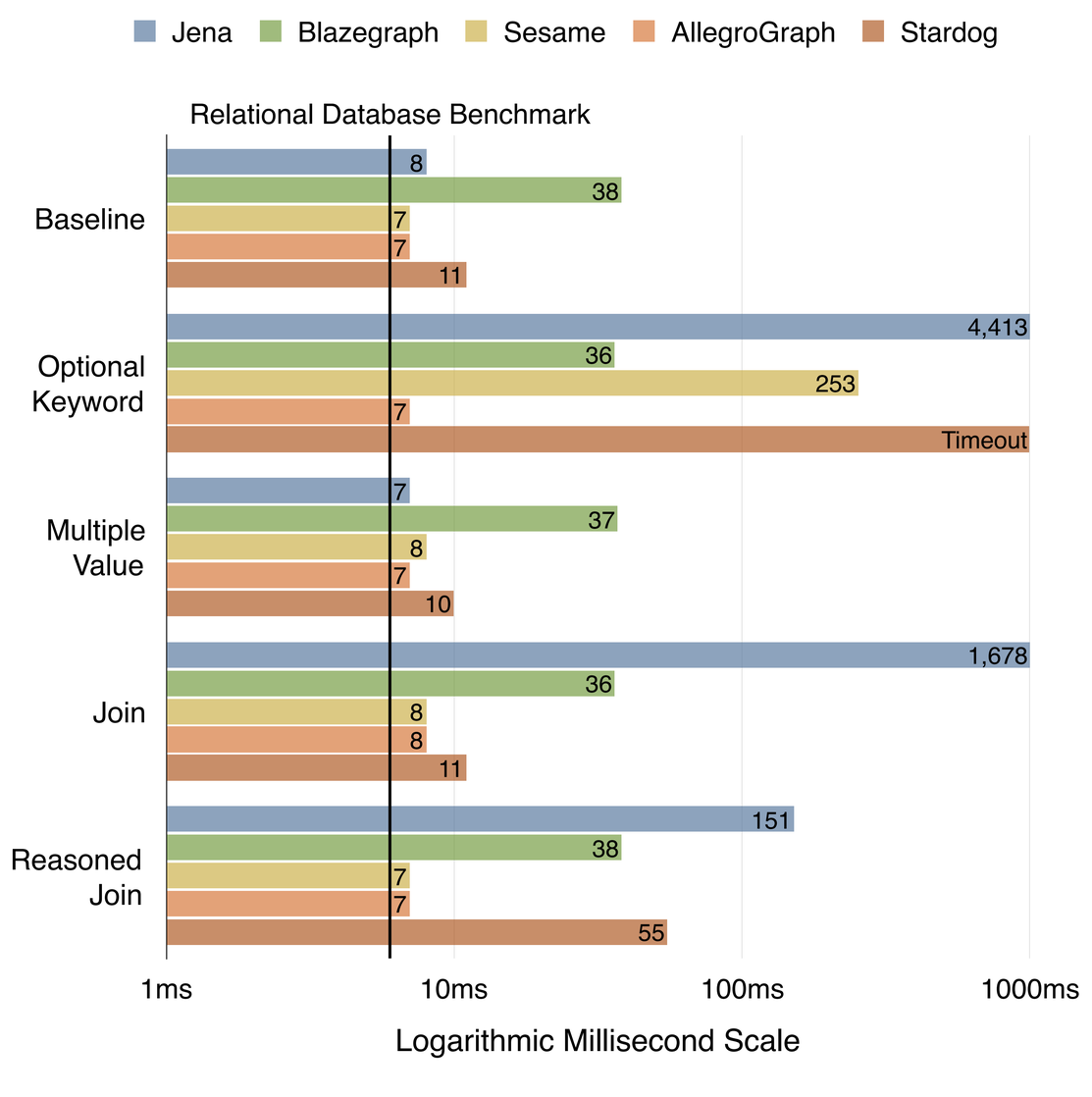
Join

```
PREFIX ub: <http://swat.cse.lehigh.edu/onto/univ-bench.owl#>
SELECT ?student
WHERE {
        ?student ub:takesCourse ?course .
        <PROFESSOR> ub:teacherOf ?course .
      }
```

Reasoned Join

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://swat.cse.lehigh.edu/onto/univ-bench.owl#>
SELECT ?student ?faculty
WHERE {
        ?student rdf:type ub:Student .
        ?faculty rdf:type ub:Faculty .
        ?student ub:advisor ?faculty
        values ?faculty { <PROFESSOR> }
       }
```

Results



Conclusion

We examined the performance of RDF triple stores for query throughput in a real-time application and compared it to the performance of a relational database. We found that performance in two of the stores, AllegroGraph and Sesame, was within the same range of the optimized relational database for most queries. Based on these results we will move forward with our plans to use an RDF triple store as the primary storage for the real-time web application in the CAFE project.

Acknowledgements

Research reported on this poster 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. M. Voigt, A. Mitschick, and J. Schulz, "Yet another triple store benchmark? Practical experiences with real-world data," CEUR Workshop Proceedings, vol. 912, no. Sda, pp. 85–94, 2012.
3. 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.