Chapter 4
The Berlin SPARQL Benchmark

Christian Bizer
Freie Universität Berlin, Germany

Andreas Schultz
Freie Universität Berlin, Germany

ABSTRACT
The SPARQL Query Language for RDF and the SPARQL Protocol for RDF are implemented by a growing number of storage systems and are used within enterprise and open Web settings. As SPARQL is taken up by the community, there is a growing need for benchmarks to compare the performance of storage systems that expose SPARQL endpoints via the SPARQL protocol. Such systems include native RDF stores as well as systems that rewrite SPARQL queries to SQL queries against non-RDF relational databases. This article introduces the Berlin SPARQL Benchmark (BSBM) for comparing the performance of native RDF stores with the performance of SPARQL-to-SQL rewriters across architectures. The benchmark is built around an e-commerce use case in which a set of products is offered by different vendors and consumers have posted reviews about products. The benchmark query mix emulates the search and navigation pattern of a consumer looking for a product. The article discusses the design of the BSBM benchmark and presents the results of a benchmark experiment comparing the performance of four popular RDF stores (Sesame, Virtuoso, Jena TDB, and Jena SDB) with the performance of two SPARQL-to-SQL rewriters (D2R Server and Virtuoso RDF Views) as well as the performance of two relational database management systems (MySQL and Virtuoso RDBMS).

1. INTRODUCTION
The SPARQL Query Language for RDF (Prud’hommeaux & Seaborne 2008) and the SPARQL Protocol for RDF (Kendall et al, 2008) are increasingly used as a standardized query API for providing access to datasets on the public Web¹ and within enterprise settings². Today, most enterprise data is stored in relational databases. In order to prevent synchronization problems, it is preferable in many situations to have direct SPARQL access to this data without having to replicate it.

DOI: 10.4018/978-1-60960-593-3.ch004
The Berlin SPARQL Benchmark

into RDF. Such direct access can be provided by SPARQL-to-SQL rewriters that translate incoming SPARQL queries on the fly into SQL queries against an application-specific relational schema based on a mapping. The resulting SQL queries are then executed against the legacy database and the query results are transformed into a SPARQL result set. An overview of existing work in this space has been gathered by the W3C RDB2RDF Incubator Group\(^3\) and is presented in (Sahoo et al., 2009).

This article introduces the Berlin SPARQL Benchmark (BSBM) for comparing the SPARQL query performance of native RDF stores with the performance of SPARQL-to-SQL rewriters. The benchmark aims to assist application developers in choosing the right architecture and the right storage system for their requirements. The benchmark might also be useful for the developers of RDF stores and SPARQL-to-SQL rewriters as it reveals the strengths and weaknesses of current systems and might help to improve them in the future.

The Berlin SPARQL Benchmark was designed in accordance with three goals:

1. The benchmark should allow the comparison of storage systems that expose SPARQL endpoints across architectures.
2. The benchmark should simulate an enterprise setting where multiple clients concurrently execute realistic workloads of use case motivated queries against the systems under test.
3. As the SPARQL query language and the SPARQL protocol are often used within scenarios that do not rely on heavyweight reasoning but focus on the integration and visualization of large amounts of data from multiple data sources, the BSBM benchmark should not be designed to require complex reasoning but to measure SPARQL query performance against large amounts of RDF data.

The BSBM benchmark is built around an e-commerce use case, where a set of products is offered by different vendors and consumers have posted reviews about products. The benchmark query mix emulates the search and navigation pattern of a consumer looking for a product.

The implementation of the benchmark consists of a data generator and a test driver. The data generator supports the creation of arbitrarily large datasets using the number of products as scale factor. In order to be able to compare the performance of RDF stores with the performance of SPARQL-to-SQL rewriters, the data generator can output two representations of the benchmark data: An RDF representation and a purely relational representation.

The test driver executes sequences of SPARQL queries over the SPARQL protocol against the system under test (SUT). In order to emulate a realistic workload, the test driver can simulate multiple clients that concurrently execute query mixes against the SUT. The queries are parameterized with random values from the benchmark dataset, in order to make it more difficult for the SUT to apply caching techniques. The test driver executes a series of warm-up query mixes before the actual performance is measured in order to benchmark systems under normal working conditions.

The BSBM benchmark also defines a SQL representation of the query mix, which the test driver can execute via JDBC against relational databases. This allows the comparison of SPARQL results with the performance of traditional RDBMS.

This article makes the following contributions to the field of benchmarking Semantic Web technologies:

1. It complements the field with a use case driven benchmark for comparing the SPARQL query performance of native RDF stores with the performance of SPARQL-to-SQL rewriters.
2. It provides guidance to application developers by applying the benchmark to measure
Related Content

Ubiquitous Semantic Applications: A Systematic Literature Review
[www.igi-global.com/article/ubiquitous-semantic-applications/113714?camid=4v1a](www.igi-global.com/article/ubiquitous-semantic-applications/113714?camid=4v1a)

Searching Linked Objects with Falcons: Approach, Implementation and Evaluation
[www.igi-global.com/chapter/searching-linked-objects-falcons/55048?camid=4v1a](www.igi-global.com/chapter/searching-linked-objects-falcons/55048?camid=4v1a)

CommuniMents: A Framework for Detecting Community Based Sentiments for Events
[www.igi-global.com/article/communiments/176735?camid=4v1a](www.igi-global.com/article/communiments/176735?camid=4v1a)

On the Graph Structure of the Web of Data
[www.igi-global.com/article/on-the-graph-structure-of-the-web-of-data/203693?camid=4v1a](www.igi-global.com/article/on-the-graph-structure-of-the-web-of-data/203693?camid=4v1a)