

MOCHA 2017

Mighty Storage Challenge

Presenter: Michael Röder



MOCHA @ ESWC 2017
Portoroz, Slovenia



HOBBIT
Holistic Benchmarking
of Big Linked Data



Horizon 2020, GA No 688227

May 30th, 2017

Axel-Cyrille Ngonga Ngomo, Institute for Applied Informatics, Germany
Iriní Fundulaki, Foundation for Research and Technology – Hellas (FORTH),
Greece
Mirko Spasic, OpenLink, UK
Henning Petzkam, Fraunhofer IAIS, Germany
Vassiliki Rentoumi, National Center for Scientific Research/Demokritos

<https://project-hobbit.eu/challenges/mighty-storage-challenge/>



USU

- 100 Euro for winners of tasks (have to beat the baseline)
- 100 Euro for the winner of most tasks

- Challenge for triple stores
- 4 Tasks
- 1st iteration
- Carried out using HOBBIT benchmarking platform

- Goal of Task 1: **Storage and Retrieval of Streamed Data from triple stores**
- Choke points:
 - **Scalability** (Data volume)
 - **Time complexity** (Data velocity)
- Input: RDF triples describing events in a production system

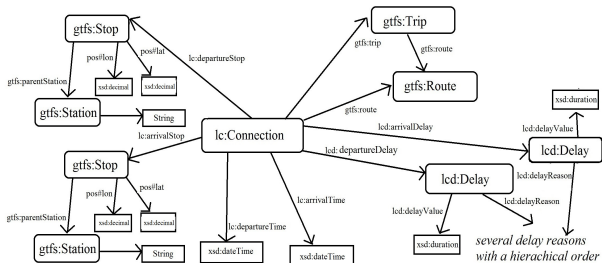
- Create reference set for each SELECT query (Jena TDB)
- Perform INSERT queries against triple store
- Perform SELECT queries against triple store
- KPIs:
 - 1 Maximum Triples Per Second (TPS)
 - 2 Recall, Precision and F-measure of each task and overall benchmark
 - 3 Delay between sending task and retrieving results

- Goal of Task 2: **To measure how data stores perform with different types of queries.**
- Starting point of DSB: Social Network Benchmark (SNB) from LDBC
- Synthetic Dataset for bulk-loading
 - RDF-ized and modified LDBC dataset
 - With lower structuredness corresponding to real-world datasets
 - 50M of triples
- Query selection performed based on the choke-points relevant for query executions (subquery unnesting, complex aggregate performance, detecting dependent group-by keys, etc)
 - Complex SPARQL SELECT queries (14 different types)
 - Simple SPARQL SELECT queries - lookups (7 different types)
 - SPARQL INSERT queries (8 different types)

- Workload consists of:
 - Bulk loading of the dataset
 - 1500 of operations (SELECT and INSERT) queries, executed sequentially
- Examination of the best performance of the tested system for a given query, by allowing the system to have all resources available
- The results of such benchmark should navigate the further optimizations of the system
- KPIs:
 - **Loading time**
 - **Throughput** (the number of queries per second)
 - Number of wrong query answers
 - Average query execution time per query type

- Task 3 is using the **Versioning Benchmark, SPBv** in order to test the ability of versioning systems:
 - to efficiently manage **evolving datasets**
 - to efficiently manage **queries** evaluated across the **multiple versions** of said datasets.
- Dataset produced by the **SPBv Data Generator**:
 - metadata represented in RDF about real world events (e.g., sport events, elections) produced.
 - data stored to different versions according to their creation date.
 - configurable in terms of dataset size, numbers of versions, generation period, data format etc.
- **Eight** different **query types** are supported
 - E.g. queries on single versions, multiple versions, deltas (difference of two versions), materialization queries on versions/deltas etc.

- Task 4 checks software for its capability of enabling **Faceted Browsing** through structured datasets.
- Faceted Browsing stands for a **session-based** and **state-dependent, interactive** method for **query formulation**, which provides a user with an effective way for exploration of a search space.
- Dataset produced by HOBBITs generator **PoDiGG** for **realistic public transport dataset**, simulating
 - train connections between stations on an artificially created map
 - including delay values and delay reasons.



- Workload consists of **11 browsing scenarios** comprising **173 SPARQL queries**
 - **Instance retrievals** - returns instances of state within browsing scenario
 - **Facet counts** - returns count for suggested facet for transition in browsing scenario
- **Choke points** consist of types of **transitions** from one state to the other
 - Ideally, a system **uses the information of the state** of the browsing scenario to return its answer to the SPARQL query that makes up the desired transition,
 - **instead of** answering the query on basis of **the entire dataset** in its original form.
- **KPIs** are **time** and **correctness**

Instance Retrieval

- Query-per-second score
- Precision
- Recall
- F1-score
- for each choke point/type of transition individually computed

Facet Counts

- Query-per-second score
- Error (Absolute distance to gold standard)
- Error ratio (Absolute distance to gold standard divided by gold standard)

- QUAD (Potocki et al.)
Tasks 1, 2 and 4
- Virtuoso (Spasić et al.)
Task 1, 2 and 4
- Baseline Virtuoso Open Source

- Second half of 2017: **Open** challenge
 - Easier participation
 - Online leader board
 - (Planned) winner announcement in February 2018
- Nearer future: **Results at the closing session**
- Now
 - Quad participant presentation
 - Virtuoso participant presentation

Thank You!
Questions?