TECHNISCHE UNIVERSITÄT DRESDEN

MMT Interactive Media Lab Dresden

DRESDEN concept

Yet Another Triple Store Benchmark?

Practical Experiences with *Real-World Data*

Martin Voigt, Annett Mitschick and Jonas Schulz
Dresden University of Technology

SDA 2012, Paphos, 9/27/2012

Outline

- Motivation: Project Background & Requirements
- Existing Benchmarks
- Benchmark Setup
- Experimental Results
- Conclusion and Future Work

Motivation :: Topic/S Project

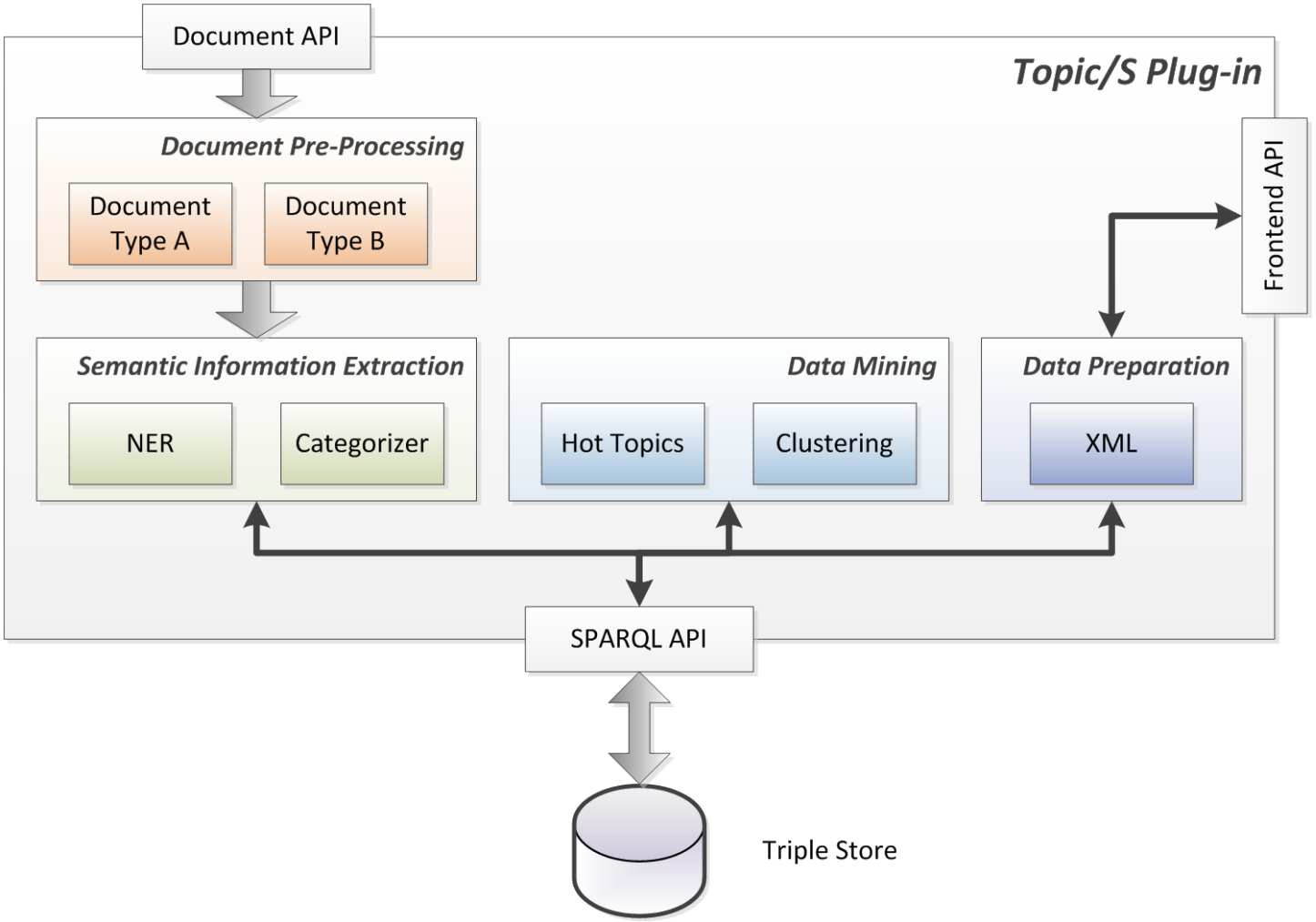
- Topic/S [1]: research project in the **publishing** sector
  - SME partner *fink&partner Media Services,* i.a. CM solutions for:

- Funded by the *European Social Fund (ESF)*

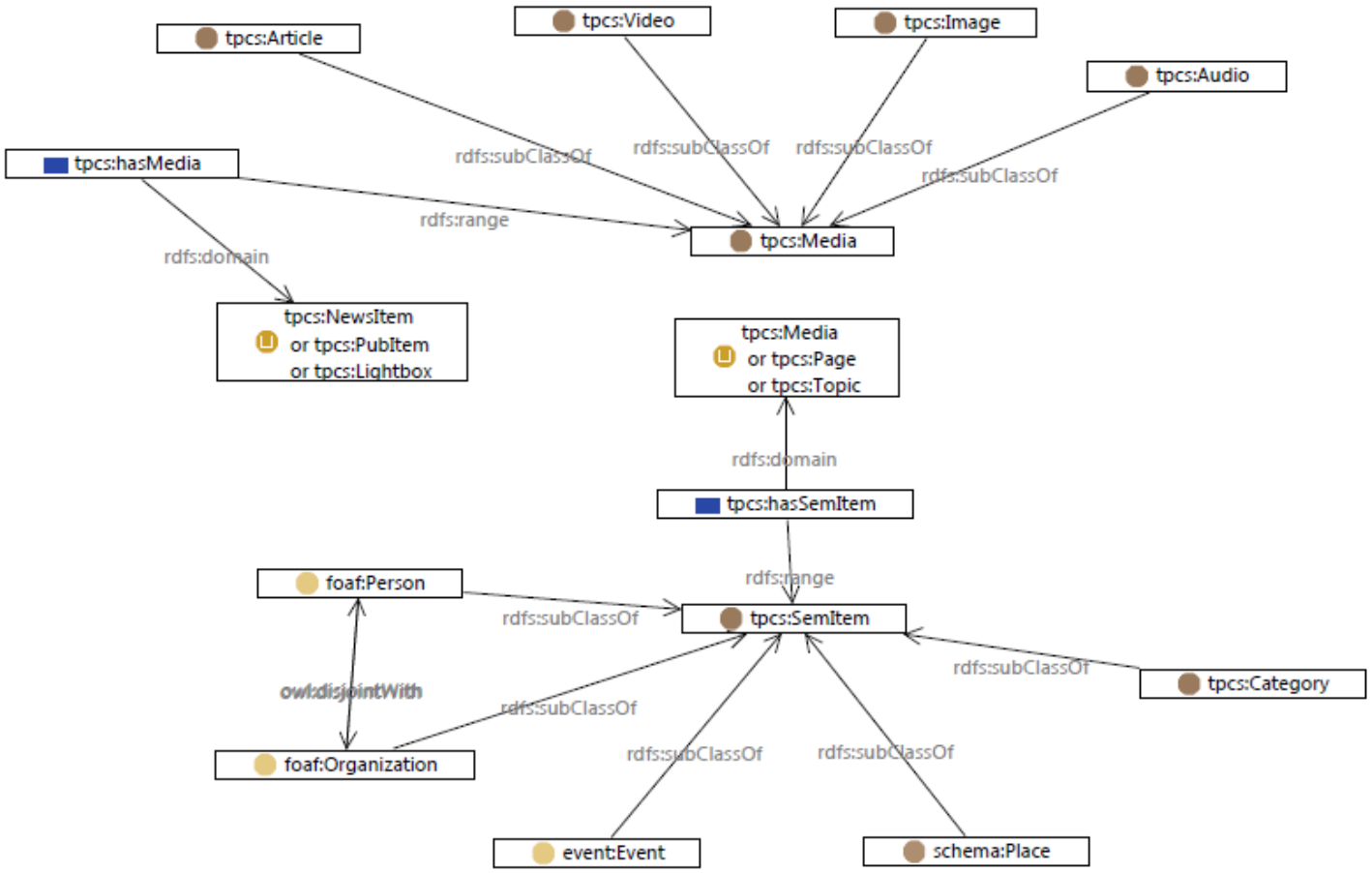
- Goals
  1. topic-based ranking and search of texts, images, and videos delivered by press media agencies
  2. Identification and recommendation of **“hot topics”**

- Approach
  - NLP: Named entity (NE) recognition & disambiguation
    → persons, places, organizations, events…
    from **multilingual** media (news articles, metadata of pictures and videos)
  - **Automatic** categorization of the news
  - Semantics-based search and recommendation for “hot topics” using **reasoning** over NEs and their temporal relations
  - Use of a powerful state-of-the-art **triple store**

- Architecture



- Overview of Topic/S ontology schema



Motivation :: Project Requirements

- A “good” triple store is required – but what are the needed capabilities?
- Storage of huge semantic data sets
  - hosting of “un-controlled” **real world** datasets to re-use Named Entities
    - especially
      - New York Times (300k triples)
      - German DBPedia (1.3m triples)
      - (parts of) YAGO2 Core [10] and GeoNames (447m triples)
  - “Controlled” information about thousands up to a million of media articles
    - their Named Entities and categories
    - different publishing information, e.g., authors, newspaper, and metadata, e.g., from EXIF or IPTC

Motivation :: Project Requirements

- Non-synthetic datasets (real-world)
- Reasoning at RDFS level within the data sets
- SPARQL Update functionality is required (SPARQL 1.1 [8])
  - Approx. 5000 new media assets per day
- Parallel use of application / triple store by up to 100 users

➔ No current benchmark identified which met **all** criteria

Existing Benchmarks

- lack of comprehensive tests on real-world datasets (non-synthetic)
  - automatically generated datasets used for benchmarking differ from real datasets, like DBpedia or Word-Net, regarding “structuredness“ [2]
    → lead to different benchmarking results
- Berlin Sparql Benchmark [3, 4] most advanced benchmark available with regard to data size, parameters, or number of RDF stores
  - using generated datasets, apart from our media archive domain
  - latest test does not address SPARQL 1.1

Existing Benchmarks

- SP²Bench [6]
  - use of a broad range of possible SPARQL constructs and their combination, without reasoning
  - carried out in 2008 on a generated dataset
- Fedbench [7]
  - query federation on real-world datasets
  - does not address RDFS reasoning nor SPARQL 1.1
  - current RDFS stores like Virtuoso are not tested
- Europeana RDF Store Report [5]
  - build upon previous studies extended by own benchmark using the Europeana dataset (approx. 380 million triples).
  - most up-to-date available results (March 2011)
  - tests did not consider RDFS reasoning, SPARQL 1.1, and heavy load (multiple queries in parallel)

Benchmark Setup

- Datasets
  - New York Times dataset [9]
  - Jamendo [10]
  - Linked Movie DB [11]
  - YAGO2 Core [12]

	NY Times	Jamendo	Movie DB	YAGO2 Core
Size (in MByte)	56,2	151,0	891,6	5427,2
Triples (in Mio)	0,35	1,05	6,15	35,43
Instances (in k)	13,2	290,4	665,4	2648,4
Classes	19	21	53	292861
Properties	69	47	222	93

Benchmark Setup

- RDF Triple Stores
  - **Fuseki** (Apache Jena TDB SPARQL Server) 0.2.3 + Jena TDB Loader 0.9.0
  - **NanoSPARQLServer** of BigData 1.2.0 (deployed on Apache Tomcat Server)
  - **OWLIM LITE** 5.0.5001 via Sesame 2.6.5 (deployed on Apache Tomcat Server)
  - OpenLink **Virtuoso** 6.01.3127 Build: Feb 15 2012

Criteria

- ✓ freely available
- ✓ allows to handle up to 100 million triples
- ✓ supports RDFS reasoning and SPARQL 1.1
- ✓ build for the Java runtime environment

Benchmark Setup :: Queries

- 6 generic queries
  - Q1: count all triples
    ```
    SELECT (COUNT(*) AS ?no) WHERE { ?s ?p ?o }
    ```
  - Q2: count all entities
    ```
    SELECT (COUNT(DISTINCT ?s) AS ?no) WHERE { ?s a [] }
    ```
  - Q3: count all distinct subjects
    ```
    SELECT (COUNT(DISTINCT ?s) AS ?no) WHERE { ?s ?p ?o }
    ```
  - Q4: count instances of a class
    ```
    SELECT  ?class (COUNT(?s) AS ?count) WHERE { ?s a ?class } GROUP BY
    ?class ORDER BY ?count
    ```
  - Q5: count subjects per predicate
    ```
    SELECT ?p (COUNT(?s) AS ?count ) { ?s ?p ?o } GROUP BY ?p ORDER BY
    ?count
    ```
  - Q6: count all triples with no literals as objects
    ```
    SELECT (COUNT(DISTINCT ?o ) AS ?no) { ?s ?p ?o
    filter(!isLiteral(?o)) }
    ```

Benchmark Setup :: Queries

- 7 dataset specific queries per dataset = 28 queries
  e.g. NYTimes: get names of all places ordered by their population which must be at least more than 10,000

```
SELECT DISTINCT ?name ?subject ?pop
WHERE {
     ?subject <http://www.geonames.org/ontology#name> ?name .
     ?subject <http://www.geonames.org/ontology#population> ?pop .
     FILTER(<http://www.w3.org/2001/XMLSchema#integer>(?pop) > 10000).
} ORDER BY <http://www.w3.org/2001/XMLSchema#integer>(?pop)
```

- 2 Update queries (insert + delete) per dataset = 8 queries

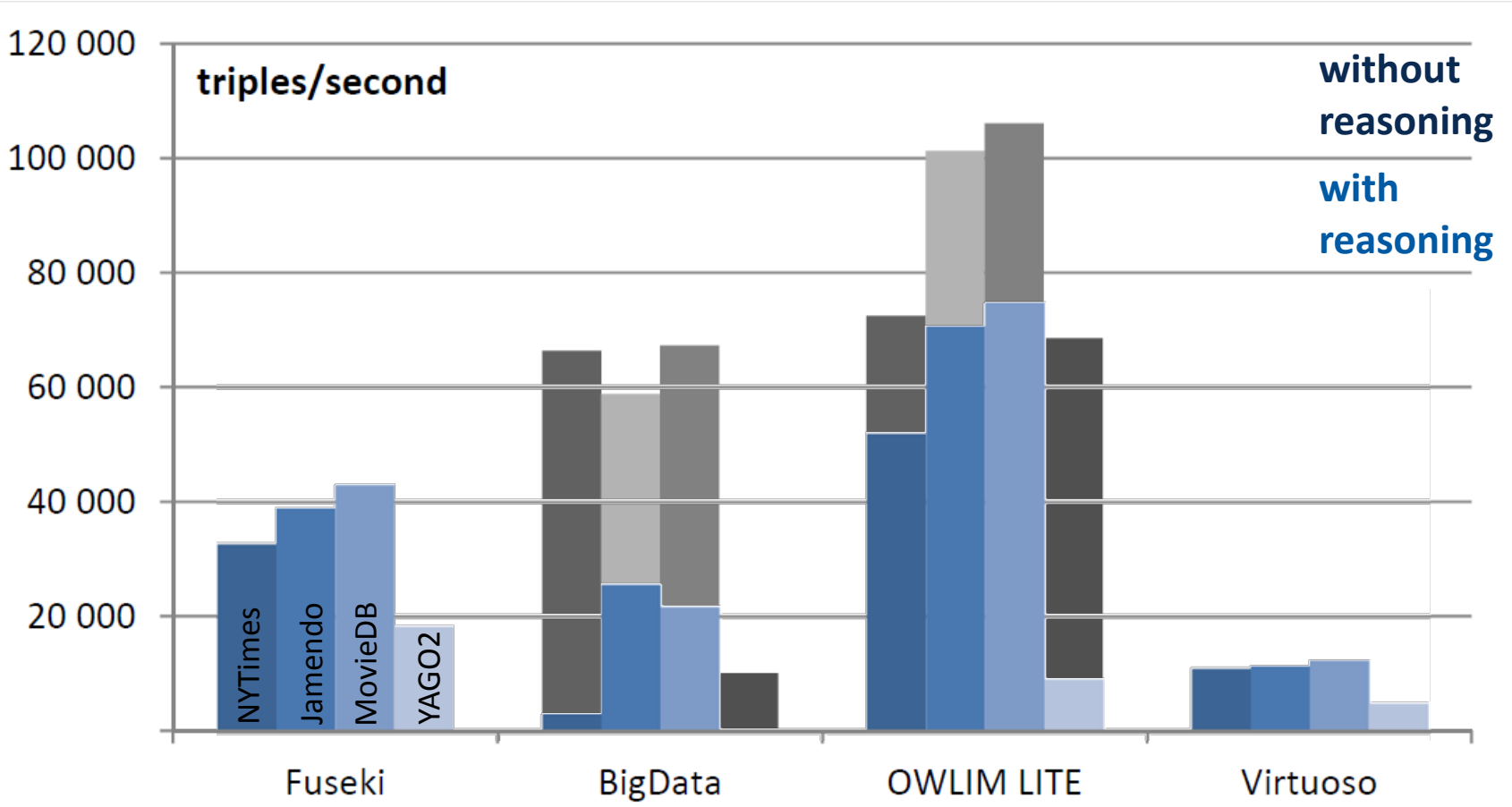
  → 42 distinct queries, 15 queries per store

Benchmark Strategy

- Steps
  - Installation and configuration of the stores
  - Merging splitted datasets, converting into N-Triple format
  - Loading each dataset three times, recording loading time and taking average of measured values
  - Run queries (20 times, round-robin) for each dataset per store with the help of a test driver for single client test without and with reasoning enabled
  - Run selected queries with a test driver for multi-client test for a specific number of parallel clients

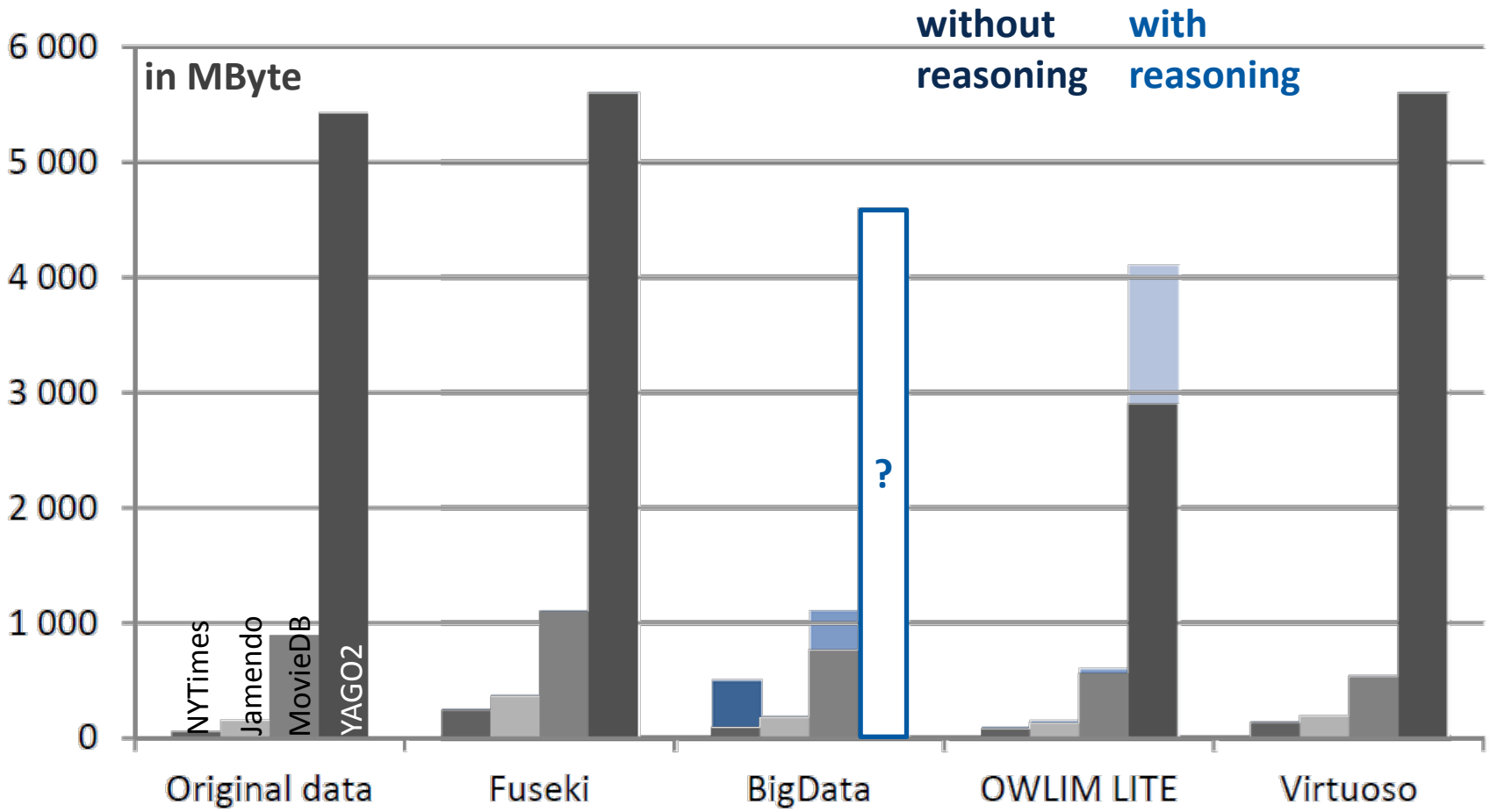
Experimental Results

- **Loading time** of each dataset (average of three runs)



Experimental Results

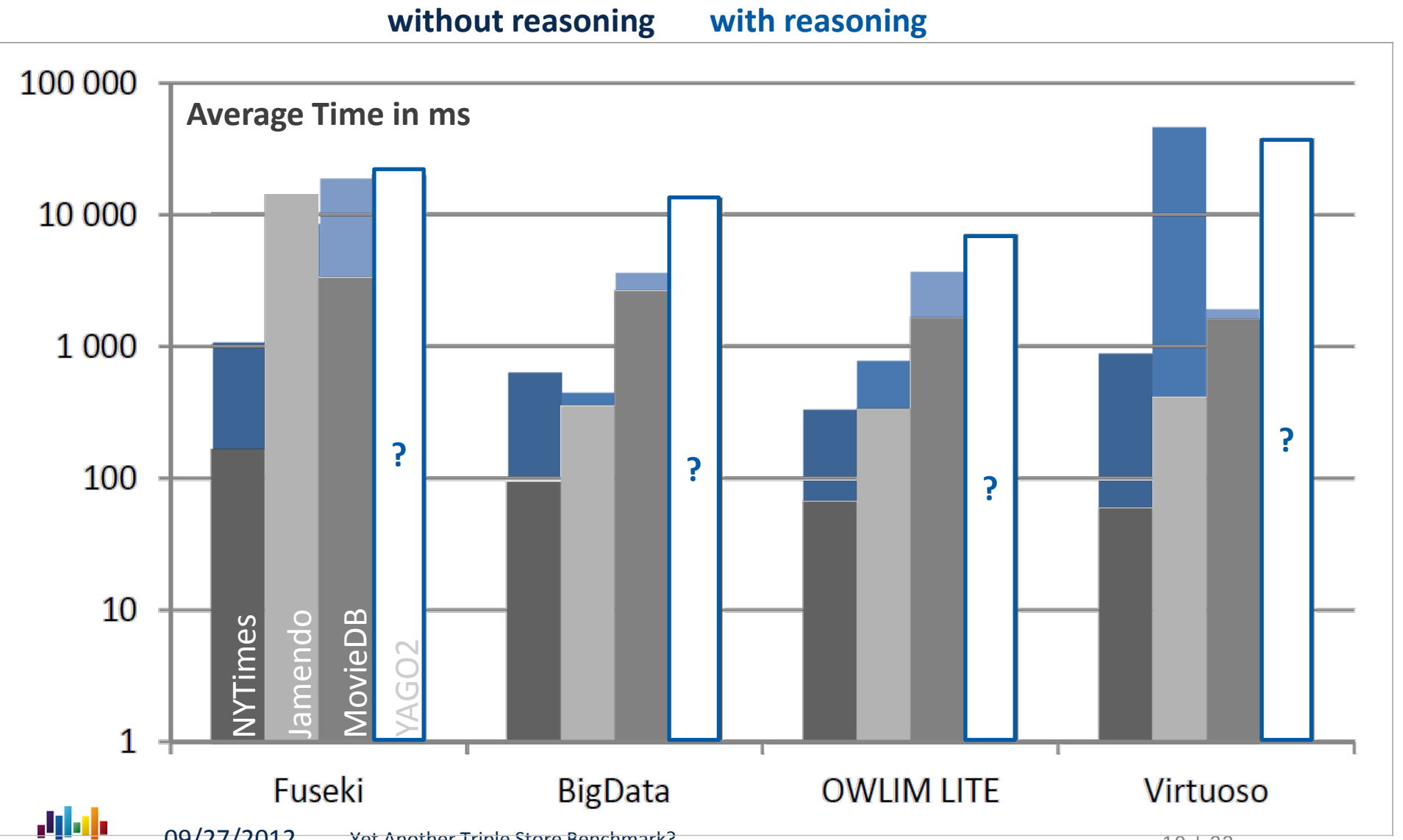
- **Memory requirement** of each store per loaded dataset



Experimental Results

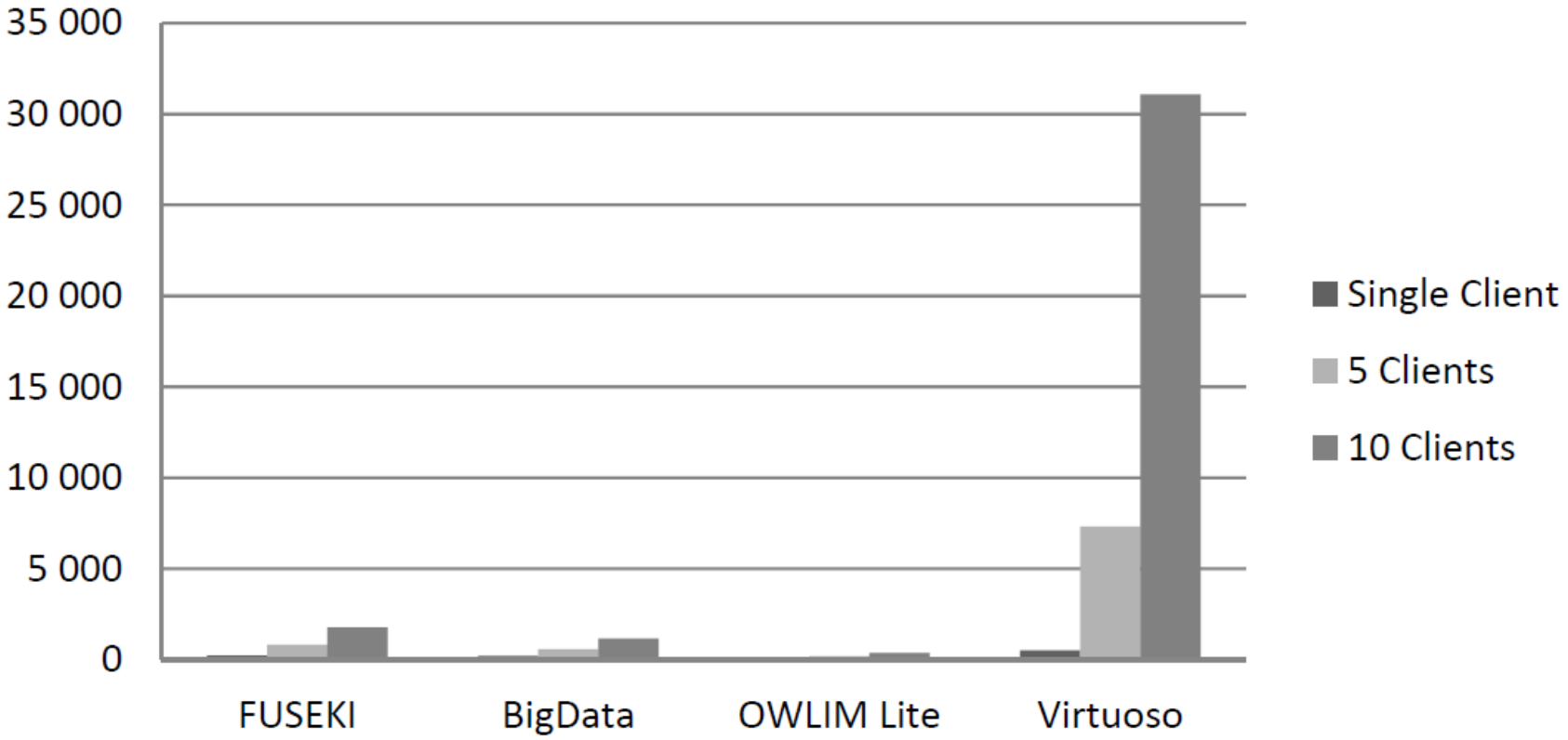
- Per-query type performance:
  - main focus on comparison of average query performance of the stores, but distinguish between the generic, the dataset-specific, and the UPDATE queries
  - average execution time from 20 runs for each query (15 queries per store in round-robin manner)
  - recorded the success rate of each query (if it delivers the expected results without any error or timeout)
  - tested query performance without and with reasoning

Experimental Results :: Query Execution



Experimental Results

- **Multi-client performance**: average query execution time (in ms), *NYTimes* dataset



Conclusion

→ http://mt.inf.tu-dresden.de/topics/bench

- comprehensive tests with up-to-date stores **and** real-world data are necessary.
  - otherwise not possible to detect anomalies
- RDFS inference
  - result set may differ from store to store!
- SPARQL 1.1
  - well implemented nowadays, but performance on UPDATE queries is varying
    → here, Virtuoso stands out
- General recommendation: NONE
  - no store could win on all fields
  - selection strongly depends on specific project requirements

Future Work

- Decision for Topic/S: OWLIM Lite
  - Good average performance for all queries and RDFS reasoning
  - Fastest in a multi-client scenario
- Current benchmark work
  - Testing YAGO2 on OWLIM Lite with RDFS reasoning
  - Benchmarking Oracle Spatial and Graph with YAGO2 and Comparison of the results
    → maybe for productive use-case (seems to be slower)
- Future Work
  - evaluate the performance of OWL reasoning for some common constructs, e. g., cardinalities
    → need to identify suitable real-world datasets

TECHNISCHE UNIVERSITÄT DRESDEN

Thank you for your attention.

For further details please contact:
Martin Voigt, martin.voigt@tu-dresden.de

References

[1] http://www.topic-s.de
[2] Duan, S., Kementsietsidis, A., Srinivas, K., Udrea, O.: Apples and oranges: a comparison of rdf benchmarks and real rdf datasets. In: Procs. of the Intern. Conf. on Management of Data, ACM (2011)
[3] Bizer, C., Schultz, A.: The berlin sparql benchmark. In: International Journal on Semantic Web & Information Systems. Volume 5. (2009)
[4] Berlin SPARQL Benchmark: http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/.
[5] Haslhofer, B., Roochi, E.M., Schandl, B., Zander, S.: Europeana rdf store report. Technical report, University of Vienna, Vienna (March 2011)
[6] Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: Sp2bench: A sparql performance benchmark. CoRR abs/0806.4627 (2008)
[7] Schmidt, M., Gorlitz, O., Haase, P., Ladwig, G., Schwarte, A., Tran, T.: Fedbench: A benchmark suite for federated semantic data query processing. In: The Semantic Web ISWC 2011. Volume 7031 of LNCS. Springer (2011)
[8] Harris, S., Seaborne, A.: SPARQL 1.1 Query Language (October 2010)
[9] New York Times Dataset: http://data.nytimes.com/.
[10] YAGO2 Dataset: http://www.mpi-inf.mpg.de/yago-naga/yago/index.html.
[11] Jamendo RDF Dump: http://dbtune.org/jamendo/
[12] Linked MovieDB Dump: http://queens.db.toronto.edu/~oktie/linkedmdb/linkedmdb-latest-dump.zip