Iguana
A Generic Framework for Benchmarking the Read-Write Performance of Triple Stores

Felix Conrads¹  Jens Lehmann²  Muhammad Saleem¹  Mohamed Morsey³  Axel-Cyrille Ngonga Ngomo¹,⁴

¹AKSW, Germany
   University of Leipzig

²University of Bonn and
   Fraunhofer IAIS

³System and Network Engineering Group,
   University of Amsterdam

⁴Department of Computer Science,
   University of Paderborn

ISWC 2017
Motivation

Semantic Web Stack

User interface and applications

Trust

Proof

Unifying Logic

Ontologies: OWL

Rules: RIF/SWRL

Taxonomies: RDFS

Data interchange: RDF

Syntax: XML

Identifiers: URI

Character Set: UNICODE

Querying: SPARQL

Cryptography
Motivation

- RDF stores are backbone of Linked Data
- No realistic scenario (e.g. no stress test)
- No unified benchmark execution
Characteristics

- **Relevant**, testing typical operations in the specific domain

- **Portable**, executable on different platforms, benchmarks and datasets

- **Scalable**, possibility of small and large data sets, with variable rates of updates and concurrent users

- **Understandable**, returns results using standard measures
Architecture

Overview

Core

Experiments

Triple Stores

Query results

Result Processor

Metrics

Result Storages

Conrads et al.

ISWC 2017
Architecture

Stresstest

Core

Experiments

Triple Stores

Result Processor

Metrics

Result Storages

Query results
Architecture
Stresstest

- Provides a realistic scenario
- Simulates several Users, querying and updating
- Network latency can be simulated
- Warmup can be configured
- Time limit or Number of query mixes
Workflow:

1. Will start at a fair random query(pattern)
2. Executes each query(pattern) and starts at beginning
3. Does this until end of Task

supports **SPARQL 1.1** (and Patterns)
Architecture
Query Pattern Instantiating

```
PREFIX dct:<http://purl.org/dc/terms/>
SELECT ?s ?p FROM <http://dbpedia.org>
WHERE
{ ?s ?p %%v1%% .
  %%v1%% dct:creator %%v2%% }
LIMIT 10
```

will be converted to:

```
PREFIX dct:<http://purl.org/dc/terms/>
SELECT ?v1 ?v2 FROM <http://dbpedia.org>
WHERE
  ?v1 dct:creator ?v2}
LIMIT K
```
Architecture

UPDATE Worker

StressTest

SPARQL Worker

SPARQL Worker

Query Manager

System

Update Manager

UPDATE Worker

UPDATE Worker

Q

U
Architecture

UPDATE Worker

- Uses either a file with UPDATE queries

- Or a path with files containing SPARQL updates

- Files have to be in format:

  \([^0-9]+/\.(added|remove)\.sparql\)
Workflow:

1. Will start according to UPDATE strategy
2. will execute next update
3. Waits an amount of time
4. Does this until end of task
   or no more updates
Architecture
Metrics & Results

Core

Experiments

Triple Stores

Query results

Result Processor

Metrics

Result Storages
Architecture

Metrics

- Queries per Second (QPS)

- Query Mixes per Hour (QMPH)

- No. Of Queries per Hour (NoQPH)

- Each Query Execution (EQE)
Architecture

Results

- **Results** will be stored in files as **CSV**

- ... as **NTriple file**

- ... in a **Triple Store**
Architecture

Extensible

It can be extended through

- Tasks
- Data Generators
- Query Handlers
- Metrics (KPI)
- Result Storages
Configuration

- All test were executed on the **same machine**
- **Datasets**: DBpedia (100%, 50%, 10%) & SWDF

- 1 hour execution
- 20 min warmup
- 250 complex queries
- DBpedia Live changesets

- 1, 4, 16 **SPARQL Users**
- 0, 1 **Update Users**
Baseline

(a) DBpedia

(b) SWDF
Parallel Request

(a) DBpedia 10%

(b) DBpedia 50%

(c) DBpedia 100%

(d) SWDF 100%
Effect of Updates

(a) DBpedia

(b) SWDF

Conrads et al.
Update & Parallel Request

(a) DBpedia 10%
(b) DBpedia 50%
(c) DBpedia 100%
(d) SWDF 100%
Summary

**IGUANA** provides a unified SPARQL Benchmark Execution Framework...
... can provide *new insights* for triple stores
... can be *executed* on every Dataset, Endpoint, and SPARQL/UPDATE queries

Outlook

- RDF Streaming
- RDF Configuration
- Any suggestions?
Thats all folks!

Any Questions?

IGUANA is Open Source

Github: http://github.com/dice-group/IGUANA
Project Site: http://iguana-benchmark.eu
Contact: conrads@informatik.uni-leipzig.de

Feel free to code with us!