RDF Graph Stores

Christopher J. Matheus
Bell Labs / Alcatel-Lucent

Message

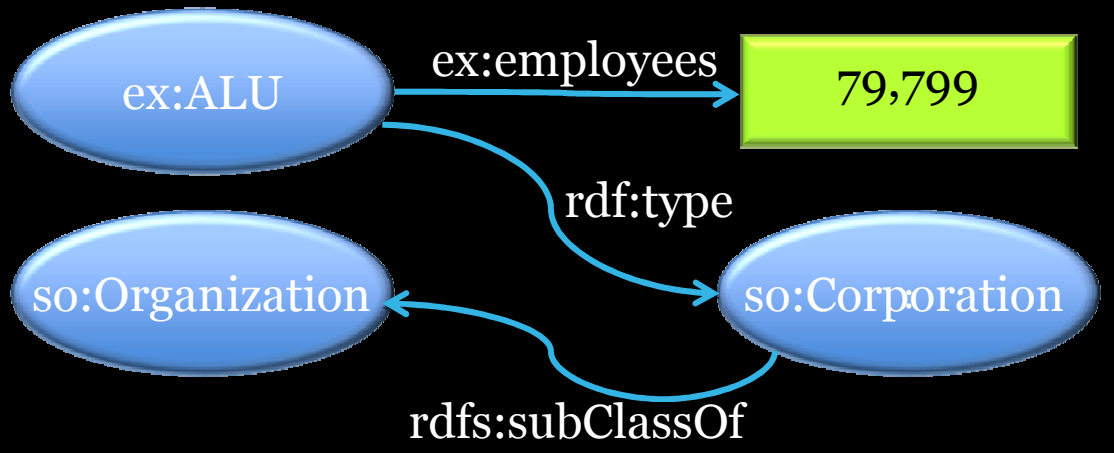
RDF Graph Stores (aka “triple stores”) store RDF triples (often with “context”) and permit querying (and maybe updating) via the SPARQL query language.

Many commercial and free open source implementations, becoming faster and more scalable and all standards-based.

RDF Basics

- RDF Triples:
- RDF Graphs:
- Serializations:
  - XML
  - Turtle
  - N-Triples
  - JSON-LD

subject - predicate – object



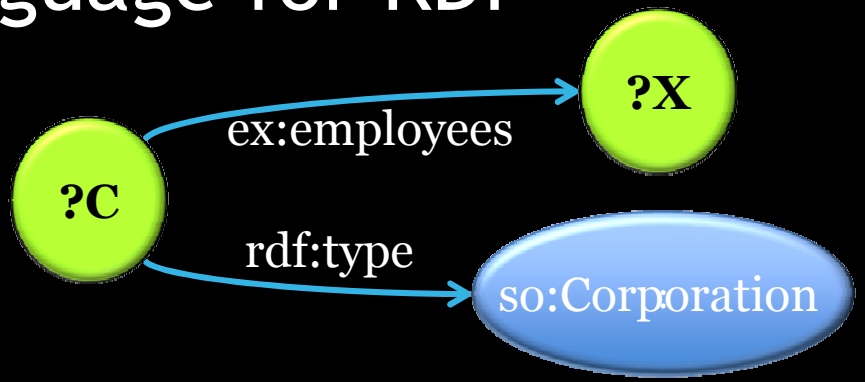

```
ex:ALU
   ex:employees    "79799";
   rdf:type        so:Corporation .
so:Corporation
   rdfs:subclassOf so:Organization .
```

RDF and the Semantic Web

- Linked Data
  - use URIs for all resources
  - use HTTP for de-referencing
  - use RDF to describe resources
  - cross-reference data sets
- Metadata and Semantics
  - Ontologies
  - Inferencing
  - Ontology Languages:
    - RDF Schema (RDFS)
    - The Web Ontology Language OWL

SPARQL

- W3C query language for RDF
- sub-graph matching with variables




```
PREFIX rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ...
SELECT ?C ?X
FROM <http://example.org/companies>
WHERE { ?C ex:employees ?X .
        ?C rdf:rdf so:Corporation . }
---------------
  ?C       ?X
ex:ALU  “79799”
```

What's exciting in SPARQL 1.1?

- Property paths
- Nested queries
- UPDATE
- HTTP Protocol
- Federated Queries
- Entailment Regimes

SPARQL Inferencing Notation (SPIN)

- W3C Member Submission (Feb. 2011)
- SPARQL-based rule & constraint language
- Enables OWL 2 RL inferencing via SPARQL
- Components:
  - RDF based syntax for SPARQL queries
  - constraint language
  - rule language
  - SPARQL function definition mechanism

RDF Graph/Triple Stores

- Triple stores – not! Need at least quads
  - named graphs, time, provenance, access rights
- Standards-based: RDF, SPARQL, RDFS/OWL
- Notable Examples:
  - OpenLink Software Virtuoso
  - Franz AllegroGraph
  - Apache Fuseki (Incubator)
  - Ontotext OWLim
  - Systap BigData
  - Garlik 4Store/5Store
  - Antonio Garrote rdfstore.js
  - many more:

35 on *http://www.w3.org/wiki/SparqlImplementations*

Which RDF Store is best?

Depends on what you need:

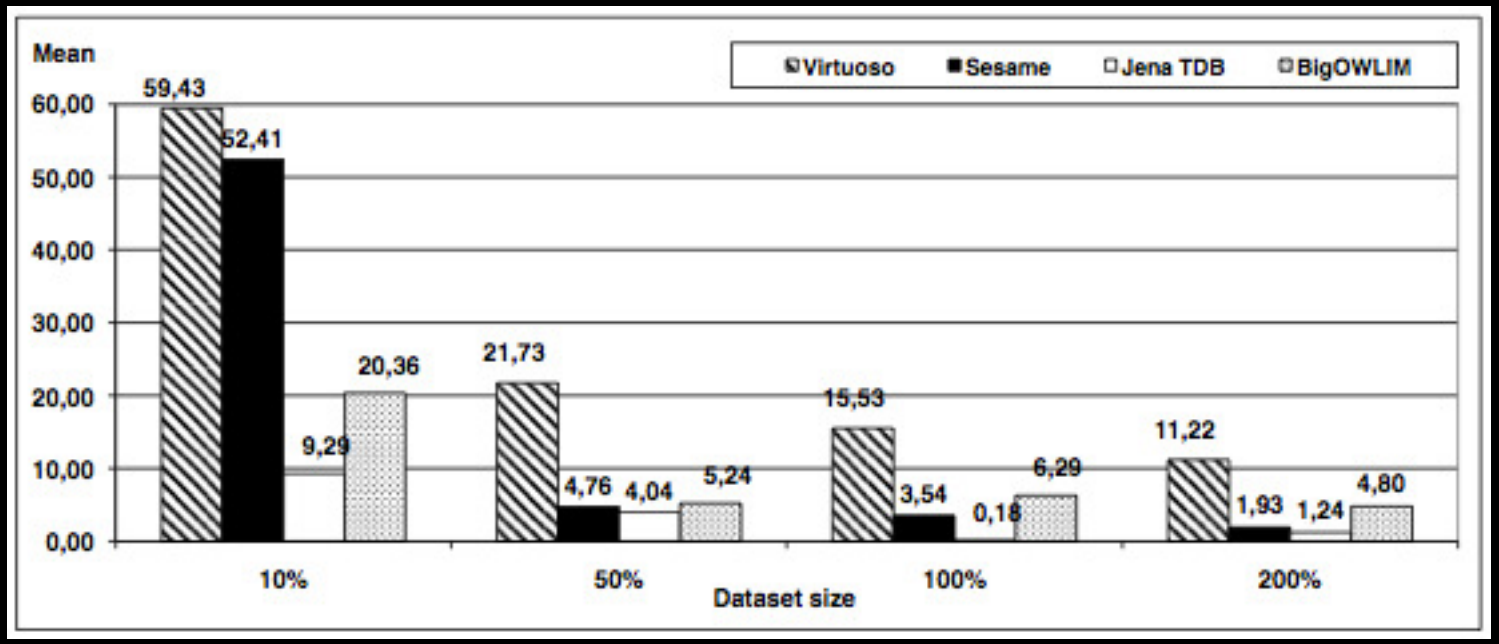
- Large data sets?
- Query performance?
- Quick, easy up and running?
- Production environment?
- Distributed, scalable?
- Rapid prototyping?

Large Data Sets: AllegroGraph

- August 16th, 2011:
  1 Trillion triples loaded and queried on an Intel-based HPC cluster-in-the-cloud provided by Stillwater SuperComputing
- Load:
  - 1,009,690,381,946 triples
  - just over 338 hours
  - average rate: 829,556 triples/second

Query Performance: Virtuoso

- OpenLink Software's RDF Store built on top of their "Universal Server" (rdbms)
- DBpedia SPARQL Benchmark (ISWC 2011)



153,737,776 triples 27,665,352 resources 25 query templates

Ease of Use: Fuseki

download, unzip, chmod, execute:

```
./fuseki-server --update --mem /ds
./s-put http://localhost:3030/ds/data default config.ttl
./s-get http://localhost:3030/ds/data default
./s-query --service http://localhost:3030/ds/query
      'SELECT * {?s ?p ?o}'
./s-update --service http://localhost:3030/ds/update
                        'CLEAR DEFAULT'
```

http://incubator.apache.org/jena/documentation/serving_data/

Production Environment

- 4Store/5Store
- AllegroGraph
- BigData
- BigOWLim
- Oracle
- Virtuoso

Distributed, Scalable?

- LOD2 Large Scale Experiments (in progress)
  - Berlin SPARQL Benchmark (BSBM) 3.1
  - SARA Lisa cluster with 64 nodes
  - VectorWise server with 40 cores and 1TB
  - Systems under evaluation:
    - BigData, BigOWLim, Jena TDB, Fuseki, Virtuoso, 4Store

Rapid prototyping: rdfstore.js

- open source Javascript RDF Store
- runs in node.js or browser
- supports most of SPARQL 1.1
- reasonable performance on LUBM
- experimental feature: watch triples

https://github.com/antoniogarrote/rdfstore-js

RDF vs. NoSQL

RDF Stores

- all data as triples
+ standards-based:
  + data model (RDF)
  + query language (SPARQL)
  + serializations (XML, etc)
+ formal semantics
+ standard tools
  + frameworks and APIs
  + editors
  + reasoners

NoSQL Stores

+ not restricted to RDF
- no standards
- no formal semantics
- store-specific tools

Key References

- W3C SPARQL Implementations
  - *http://www.w3.org/wiki/SparqlImplementations*
- W3C Wiki: Large Triple Stores
  - *http://www.w3.org/wiki/LargeTripleStores*
- W3C Wiki: RDF Store Benchmarking
  - *http://www.w3.org/wiki/RdfStoreBenchmarking*
- W3C RDF Working Group Wiki
  - *http://www.w3.org/2011/rdf-wg/*
- W3C SPARQL 1.1 Draft Recommendation
  - *http://www.w3.org/TR/sparql11-query/*

chris.matheus@alcatel-lucent.com