SchemEX
Creating the Yellow Pages of the LOD Cloud
Mathias Konrath, Thomas Gottron, Ansgar Scherp

WeST
UNIVERSITÄT KOBLENZ · LANDAU
Scenario

• People who are politicians and actors

• Who else?
• Where do they live?
• Whom do they know?
Problem

• Execute those queries on the LOD cloud
• No single federated query interface provided

“politicians and actors”
Problem

• Execute those queries on the LOD cloud
• No single federated query interface provided

“politicians and actors”
Principle Solution

• Suitable index structure for looking up sources

“politicians and actors”
The Naive Approach

1. Download the entire LOD cloud
2. Put it into a (really) large triple store
3. Process the data and extract schema
4. Provide lookup

- Big machinery
- Late in processing the data
- High effort to scale with LOD cloud
The Naive Approach

1. Download the entire LOD cloud
2. Put it into a (really) large triple store
3. Process the data and extract
4. Provide lookup

- Big machinery
- Late in processing the data
- High effort to scale with LOD cloud

Can we do smarter?
Yes, we can ...
The SchemEX Approach

- Stream-based schema extraction
- While crawling the data
The SchemEX Approach

- Stream-based schema extraction
- While crawling the data

LOD-Crawler
RDF-Dump
Triple Store

Nquad-Stream

FIFO

Instance-Cache

NxParser

Parser

Schema-Extractor

RDF
RDBMS

Schema
Efficient Instance Cache

- Observe a quadruple stream from LD spider

  Triple-/NQuad-Stream

  observed window

- Ring queue, backed up by a hash map
- Organizes triples with same subject URI
- Dismiss oldest, when cache full (FIFO)

→ Runtime complexity $O(1)$
Building the Schema and Index

ConsistsOf

hasDataSource

hasEQClass

hasDataSources

RDF classes

Type clusters

Equivalence classes

Data sources

Building the Schema and Index
Computing SchemEX: TimBL Data Set

- Analysis of a smaller data set
- 11 M triples, TimBL’s FOAF profile
- LDspider with ~ 2k triples / sec

- Different cache sizes: 100, 1k, 10k, 50k, 100k
- Compared SchemEX with reference schema
- Index queries on all Types, TCs, EQCs
- Good precision/recall ratio at 50k+
## Computing SchemEX: Full BTC 2011 Data

<table>
<thead>
<tr>
<th></th>
<th>1st billion</th>
<th>2nd billion</th>
<th>full dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>#triples</td>
<td>1 billion</td>
<td>1 billion</td>
<td>2.17 billion</td>
</tr>
<tr>
<td>#instances</td>
<td>187.7M</td>
<td>222.6M</td>
<td>450.0M</td>
</tr>
<tr>
<td>#data sources</td>
<td>13.5M</td>
<td>9.5M</td>
<td>24.1M</td>
</tr>
<tr>
<td>#type clusters</td>
<td>208.5k</td>
<td>248.5k</td>
<td>448.6k</td>
</tr>
<tr>
<td>#equivalence classes</td>
<td>0.97M</td>
<td>1.14M</td>
<td>2.12M</td>
</tr>
<tr>
<td>#triples index</td>
<td>29.1M</td>
<td>24.8M</td>
<td>54.7M</td>
</tr>
<tr>
<td>Compression ratio</td>
<td>2.91%</td>
<td>2.48%</td>
<td>2.52%</td>
</tr>
<tr>
<td>runtime (hh:mm)</td>
<td>6:51</td>
<td>6:05</td>
<td>15:16</td>
</tr>
<tr>
<td>average runtime per 10M chunk</td>
<td>247 s</td>
<td>219 s</td>
<td>252 s</td>
</tr>
<tr>
<td>standard deviation</td>
<td>80 s</td>
<td>12 s</td>
<td>57 s</td>
</tr>
<tr>
<td>#triples/sec.</td>
<td>40.5k</td>
<td>45.6k</td>
<td>39.5k</td>
</tr>
</tbody>
</table>

Cache size: 50 k
## Computing SchemEX: Full BTC 2011 Data

<table>
<thead>
<tr>
<th></th>
<th>1st billion</th>
<th>2nd billion</th>
<th>full dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>#triples</td>
<td>1 billion</td>
<td>1 billion</td>
<td>2.17 billion</td>
</tr>
<tr>
<td>#instances</td>
<td>187.7M</td>
<td>222.6M</td>
<td>450.0M</td>
</tr>
<tr>
<td>#data sources</td>
<td>13.5M</td>
<td>9.5M</td>
<td>24.1M</td>
</tr>
<tr>
<td>#type clusters</td>
<td>208.5k</td>
<td>248.5k</td>
<td>448.6k</td>
</tr>
<tr>
<td>#equivalence classes</td>
<td>0.97M</td>
<td>1.14M</td>
<td>2.12M</td>
</tr>
<tr>
<td>#triples index</td>
<td>29.1M</td>
<td>24.8M</td>
<td>54.7M</td>
</tr>
<tr>
<td>Compression ratio</td>
<td>2.91%</td>
<td>2.48%</td>
<td>2.52%</td>
</tr>
<tr>
<td>runtime (hh:mm)</td>
<td>6:51</td>
<td>6:05</td>
<td>15:16</td>
</tr>
<tr>
<td>average runtime per 10M chunk</td>
<td>247 s</td>
<td>219 s</td>
<td>252 s</td>
</tr>
<tr>
<td>standard deviation</td>
<td>80 s</td>
<td>12 s</td>
<td>57 s</td>
</tr>
<tr>
<td>#triples/sec.</td>
<td>40.5k</td>
<td>45.6k</td>
<td>39.5k</td>
</tr>
</tbody>
</table>

Cache size: 50 k
Conclusions: SchemEX

- Stream-based approach to schema extraction
- Scalable to arbitrary amount of Linked Data
- Applicable on commodity hardware (4GB RAM, standard single CPU)

- Lookup-index to find relevant data sources
- Support federated queries on the LOD cloud