Query Heterogeneous Personal Information On The Go

Danh Le Phuoc, Anh Le Tuan, Gregor Scheile And Manfred Hauswirth
Insight Centre for Data Analytics
National University of Ireland, Galway

ISWC 2014, Riva Del Garda, Trentino, Italy
Heterogeneous Personal Information Data Sources
Integration of heterogeneous personal information using Linked Data approach

ID consolidation issues across data silos
Integration of heterogeneous personal information on the PCs/Servers

- **Identity consolidation:**
  - Explicit property owl:sameAs (transitive reasoning)
  - Implicit rules defined from properties: an *inverse-identification* property (ex. same email → same person).

- **Aggregated querying:**
  - Entailment regimes;
  - Store all computed possible explicit RDF statements.

Expensive and Resource-Intensive
It does not work on mobile devices!!!

Insight Centre for Data Analytics

Chapter 3. Empirical Evaluation and Experiment Analysis

Increases. TDBoid inserts data faster than OTG-BDB; however it suffers an out-of-memory error, and does not store all the experimental dataset. OTG-BDB finishes the experiment and stores all the dataset of 2200 products' profiles; however its updating throughput is very low.

As shown in Figure 3.2, the writing throughput of TDBoid first starts at 240 triples/second and then goes down slightly to 180 triples/second. After inserting 200 thousand triples TDBoid crashes and no more triples are added into storage. The dataset of 2200 products contains about 350 thousand triples. This is an average of 160 triples per product. Hence, in our mobile application's context, TDBoid can update from 1 to 1.5 products per second and only can store data for 1250 products. The updating throughput of OTG-BDB is about 140 triples/second in the beginning of the experiment. This gradually drops down to 50 after adding 100 thousand triples and then to 10 after 200 thousand triples. So, after adding a thousand products, it takes about 15 seconds for OTG-BDB to add one more product.

When adding the first 50 thousand triples, OTG-BDB and TDBoid consume the same amount of memory. Their memory heap is rising from 0MB to 30MB with the same shape. However, the data size that can be loaded by both system configurations. Then we measure the time response of each SPARQL query from the queries mentioned above.

• Experiment 3 - Memory consumption:
  In this experiment, we measured the memory consumption of two system configurations while performing the mix query. For conducting this test, we use datasets with different size. With each system, we run the query mix on each dataset and record the maximum memory heapsize that the operating system is allocated.
Only use basic functionality of RDF Store: Compose and Maintain Read-to-Query Graphs

Consolidated graph

SameAs Graph
Simple SPARQL query on Unified View linked by Consolidated Graphs

"find all friends in Facebook that are tagged with me in a photo posted in Google+ or Facebook or other data spaces."

```
SELECT ?fbfriend
FROM NAMED ds:facebook
FROM NAMED ds:cg
FROM NAMED ds:sameas
WHERE{
  GRAPH ds:facebook{fb:me foaf:knows ?fbfriend}
  GRAPH ds:sameas{?cgfriend owl:sameAs ?fbfriend. ?cgme owl:sameAs fb:me.}
}
```

Shorten the queries using Automatic ID Resolving

```
SELECT ?fbfriend
FROM NAMED ds:facebook
FROM NAMED ds:cg
WHERE{
  GRAPH ds:facebook{fb:me foaf:knows ?fbfriend.}
  GRAPH ds:cg{?cgfriend pim:tagged ?photo. fb:me pim:tagged ?photo.}
}
```
A framework for RDF-based data integration on mobile devices.
Boosting RDF Store On Android Devices (RDF On The Go – RDF-OTG)

- **Node Encoding**
  - Node label is stored as String in flash store;
  - Use 32 bit (an integer) for node identifier.

- **Flash friendly triple store**
  - Adaptive block based memory management for memory resilient
    For Query Processing Engine
  - Indexing scheme:
    - Two level of index.
      - A sorted list per block. (on flash)
      - Sparse index. (on memory)
  - Block’s slicing operations to improve writing speed.

- **Cache replacement policy**:
  - Release the block with highest chance not being changed.
Evaluation setup

- Devices:
  - HTC Desire:
    - Android OS 2.3.3;
    - Ram : 404 MB with 32 MB JVM heap;
    - 998Mhz CPU.
  - Samsung Galaxy:
    - Android OS 4.2.2;
    - Ram : 694 MB with 96 MB JVM heap;
    - 1200Mhz CPU.
  - Nexus 7 Tablet:
    - Android OS 4.2.2;
    - Ram : 974 MB with 64 MB JVM heap;
    - 1200Mhz CPU.

- Compare with TDBoid (Android version of Jena TDB)
- Datasets: Simulated user profiles for Google+, LinkedIn and Facebook
- Queries: 8 types of queries
Updating throughputs of RDF-OTG and TDBDoid:

(a) HTC Desire

(b) Samsung Galaxy Nexus

(c) Nexus 7 Tablet
Comparing the query response time of RDF-OTG and TDBoid
Querying Heterogeneous Personal Information On The Go

Related work

Many works have aimed to better allow a Semantic Web and RDF have long been used as a solution for modeling and integrating heterogeneous personal data. This is due to the requirements of data portability in terms of identification, personal profiles and friend networks. SemanticLife is one of the early attempts to employ user’s access to multiple data silos by using Semantic Web technologies to satisfy the user’s access to multiple data silos by using Semantic Web technologies to satisfy the requirements of data portability in terms of identification, personal profiles and friend networks.

Query response time (second) and memory footprint on maximum datasets for RDF-OTG

<table>
<thead>
<tr>
<th>Device</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTC (900 Prof/1M tr)</td>
<td>0.114</td>
<td>19.035</td>
<td>0.402</td>
<td>3.714</td>
<td>failed</td>
<td>6.093</td>
<td>22.652</td>
<td>24.345</td>
</tr>
<tr>
<td>GALAXY (1.2K Prof/1.5M tr)</td>
<td>0.322</td>
<td>14.705</td>
<td>0.341</td>
<td>3.490</td>
<td>7.858</td>
<td>1.713</td>
<td>16.111</td>
<td>19.223</td>
</tr>
<tr>
<td>NEXUS 7 (1.2K Prof/1.5M tr)</td>
<td>0.113</td>
<td>11.638</td>
<td>0.242</td>
<td>2.458</td>
<td>6.649</td>
<td>1.579</td>
<td>12.769</td>
<td>17.044</td>
</tr>
</tbody>
</table>

Query response time (second)

<table>
<thead>
<tr>
<th>Size</th>
<th>RDF-OTG</th>
<th>RDF-OTG</th>
<th>RDF-OTG</th>
<th>RDF-OTG</th>
<th>RDF-OTG</th>
<th>RDF-OTG</th>
<th>RDF-OTG</th>
<th>RDF-OTG</th>
</tr>
</thead>
<tbody>
<tr>
<td>100k</td>
<td>4MB</td>
<td>9MB</td>
<td>17MB</td>
<td>34MB</td>
<td>46MB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200k</td>
<td>11MB</td>
<td>26MB</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory footprint
Query Heterogeneous Personal Information On The Go

• RDF-On-The-Go: Store and query up to 4 million triples on the secondary storage

• Low memory footprint: 4-32MB heap size to embed to any Android application

• Full SPARQL 1.1 support

• RDF-On-The-Go on Android at https://code.google.com/p/rdfontheego/