A Study of RDF Storage Management and Query Evaluation Techniques
Guanghua Huang, Vahid Jalali, Pairoj Rattadilok, Jahnavi Virkar

What is RDF
- Resource Description Framework (RDF) is a standard model for data interchange on the Web.
- RDF statements could be stored as triples:

Subject | Predicate | Object
---|---|---
ID1 | type | Book
ID1 | title | "XYZ"
ID1 | author | "Fox, Joe"
ID1 | copyright | "2001"
ID2 | type | CD
ID2 | title | "ABC"
ID2 | artist | "On, Tim"
ID2 | language | "French"

How to store RDF
There are several approaches proposed for storing RDF data in Relational Databases.
1. Triple store: All statements are stored in a single table with three columns.
2. Vertical partitioning: For each predicate, a table with two columns (subject and object) would be created.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Object</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1</td>
<td>Book</td>
<td>&quot;XYZ&quot;</td>
</tr>
<tr>
<td>ID2</td>
<td>CD</td>
<td>&quot;ABC&quot;</td>
</tr>
</tbody>
</table>

3. Property table: Subjects with common properties are grouped into same tables.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Type</th>
<th>Title</th>
<th>copyright</th>
</tr>
</thead>
</table>
| ID1 | Book | "XYZ" | "2001"
| ID2 | CD | "ABC" | NULL

How to query RDF
- RDF query languages could be used for retrieving and manipulating data represented in RDF format.
- The RDF query language recommended by W3C is SPARQL.
- Example of a SPARQL query for getting the title of the book(s) written by Joe Fox written in 2001:

```
Select ?title
```

Q: What is the most expensive operation in SPARQL?
A: Join.
Q: How many different types of SPARQL join exist?
A: 6: subject-subject (s-s), object-object (o-o), property-property (p-p), subject-object (s-o), subject-property (s-p), property-object (p-o).
Q: What are the possible characteristics of the data being queried?
A: Multi-valued properties.
Large number of unique properties.
Heterogeneity (Null values in the tables).

How to evaluate RDF storage approaches
- Different benchmarks have been created for RDF storage evaluation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of triples</th>
<th># of unique properties</th>
<th>Used joins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barton</td>
<td>51 million</td>
<td>285</td>
<td>s-s, s-o</td>
</tr>
<tr>
<td>Yago</td>
<td>40 million</td>
<td>93</td>
<td>s-s, s-o, o-o</td>
</tr>
<tr>
<td>LibraryThing</td>
<td>36 million</td>
<td>380000</td>
<td>s-s, s-o, o-o</td>
</tr>
<tr>
<td>LUBM</td>
<td>6.8 million</td>
<td>32</td>
<td>s-o, o-o</td>
</tr>
<tr>
<td>JenA</td>
<td>Not reported</td>
<td>Not reported</td>
<td>s-s, s-o</td>
</tr>
</tbody>
</table>

Q: Which benchmarks are used for evaluating different RDF storage methods?
A: Different benchmarks are used depending on the storage method:
- Vertical partitioning: Barton
- Triple store: Barton, LUBM
- RDF-3X: Barton, Yago, LibraryThing
- Property table: JenA

Q: Is there something missing?
A: Yes, no property join is used for evaluating the proposed methods. Also, not all aspects of data characteristics are covered in existing evaluations.

Evaluations
Goals:
- Evaluating different storage methods:
  1. With different types of SPARQL queries.
  2. On datasets with different characteristics.

- Triple: Standard triple store
  - vp: vertical partitioning
  - hex: Standard triple store enhanced with 6 different clustered indexes

- Future works:
  1. Increasing the number of unique properties.
  2. Increasing the number of records.
  3. Performing evaluations on property table.

- SELECT A.obj count(*) FROM triple AS A WHERE A.predicate = "type" GROUP BY A.obj
- SELECT B.prop count(*) FROM triple AS A, triple AS B WHERE A.predicate = "type" GROUP BY B.prop