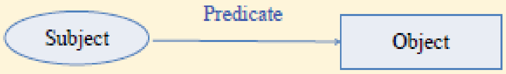
A Study of RDF Storage Management and Query Evaluation Techniques

Guanghua Huang, Vahid Jalali, Pairoj Rattadilok, Janhavi 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:



- An example:

Subj.	Prop.	Obj.
ID1	type	Book
ID1	title	"XYZ"
ID1	author	"Fox, Joe"
ID1	copyright	"2001"
ID2	type	CD
ID2	title	"ABC"
ID2	artist	"Orr, 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.

type

Subject	Object
ID1	Book
ID2	CD

author

Subject	Object
ID1	"Fox, Joe"

title

Subject	Object
ID1	"XYZ"
ID2	"ABC"

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

Property table

Subject	Type	Title	copyright
ID1	Book	"XYZ"	"2001"
ID2	CD	"ABC"	NULL

Left-over triples

Subj.	Prop.	Obj.
ID1	author	"Fox, Joe"
ID2	artist	"Orr, Tim"
ID2	language	"French"

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    Where {?book author "Fox, Joe"
                        ?book copyright "2001"
                        ?book title ?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

Dataset	# of triples	# of unique properties	Used joins
Barton	51 million	285	s-s, s-o
Yago	40 million	93	s-s, s-o, o-o
LibraryThing	36 million	382000	s-s, s-o, o-o
LUBM	6.8 million	32	s-o, o-o
Jena	Not reported	Not reported	s-s, s-o

- Which benchmarks are used for evaluating different RDF storage methods?

Storage method	Benchmark
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:

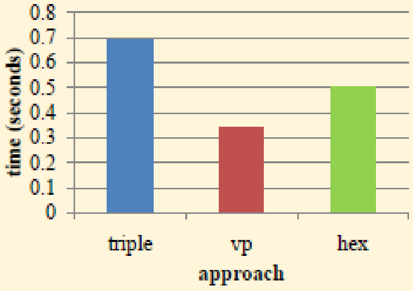
1. With different types of SPARQL queries.
2. On datasets with different characteristics.

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

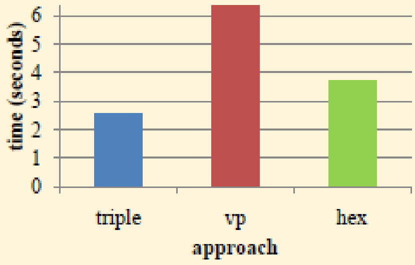
All experiments performed on 700000 records from Barton dataset

Future works:

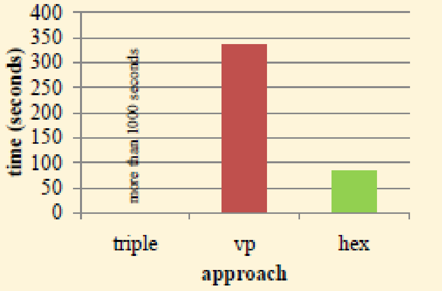
1. Increasing the number of unique properties.
2. Increasing the number of records.
3. Performing evaluations on property table



```
SELECT A.obj, count(*) FROM triples AS A
WHERE A.prop = "<type>" GROUP BY A.obj
```



```
SELECT A.obj, count(*) FROM triples AS A
WHERE A.prop = "<type>" GROUP BY A.obj
```



```
SELECT B.prop, count(*) FROM triples AS A, triples AS B,
properties AS P WHERE A.subj = B.subj
AND A.prop = "<type>" AND A.obj = "<Text>"
AND P.prop = B.prop GROUP BY B.prop
```

B669 Project