Massive-scale RDF Processing UsingCompressed Bitmap Indexes

Kamesh Madduri and John Wu
Scientific Data Management
Lawrence Berkeley National Laboratory
SSDBM 2011

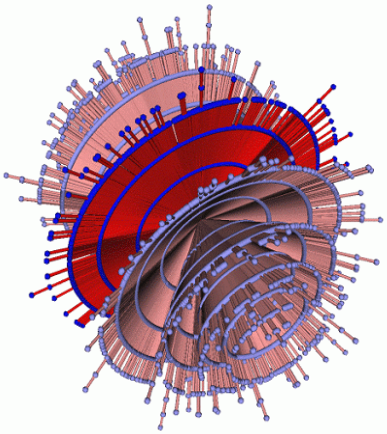
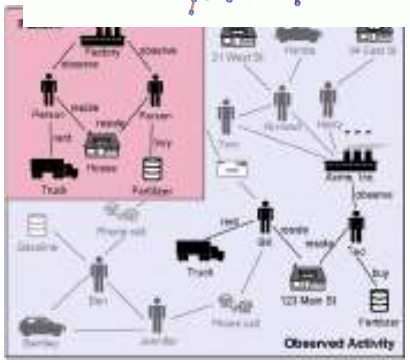
Talk Outline

- •Introduction to RDF and SPARQL queries
- •Bitmap Index Construction for RDF data
- •Query evaluation scheme using compressed bitmap indexes
- •Performance results

Semantic Data Analysis and RDF

- •The RDF (Resource Description Framework) data model is a popular abstraction for linked data repositories
  - –Records are in *triple* form [<subject> <predicate> <object>]
  - –Data sets with a few billion triples
- 

pres


st



SPARQL

- •Query language expressing conjunctions and disjunctions of triple patterns
- •Each conjunction corresponds to a join
- •SPARQL queries can be viewed as graph pattern-matching
- •Example query from the Lehigh University Benchmark Suite (LUBM):
  - **–select ?x ?y ?z where {**
    ?x rdf:type ub:GraduateStudent .
    ?y rdf:type ub:University .
    ?z rdf:type ub:Department .
    ?x ub:memberOf ?z .
    ?z ub:subOrganizationOf ?y .
    ?x ub:undergraduateDegreeFrom ?y .
    }

FastBit-RDF: Our Contributions

- •We use the compressed bitmap indexing software FastBit to index RDF data
  - –Several different types of bitmap indexes
  - –Fast parallel index construction
- •We present a new SPARQL query evaluation approach
  - –Pattern-matching queries on RDF data are modified to use bitmap indexes
- •Our approach is up to an order of magnitude faster than the RDF-3X SPARQL query software
  - –Speedup insight: The nested joins in SPARQL queries can be expressed as fast and I/O optimal bit vector operations

Bitmap Index Construction: Data structures

- •RDF data is commonly expressed as triples
  - –(subject, predicate, object)
- •We create and maintain two string to integer dictionaries
  - –Predicate strings to integer IDs (PDict)
  - –A combined subject and object dictionary (SODict)
- •We construct three Column Indexes, one for each column
  - –Keys are distinct values, bit vectors are the size of the number of records, and a bit is set if the value appears in a particular record
  - –Analogous to traditional bitmap indexes
- • We construct three Composite Indexes
  - –Keys are composite values of subject-object, predicate-subject, and predicate-object
  - –Each composite key has a bit vector associated with it

Column Index Data Structures: Illustration

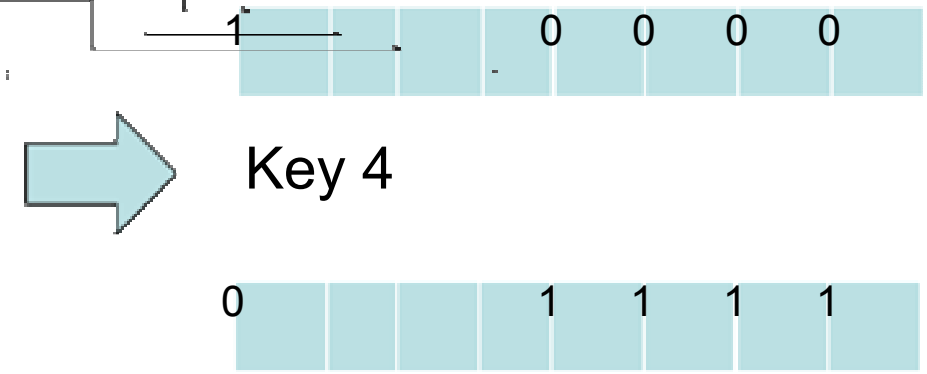
Triple data

S	P	O
0	0	0
0	1	1
0	2	2
0	3	3
4	0	5
4	1	6
4	2	7
4	3	3

nSO = 8
nP = 4

Subject
Column Index

Two keys: 0, 4



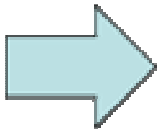
Object index (8 bit vectors) and predicate index (4 bit vectors) can be similarly constructed.

Composite Index: Illustration

Triple data

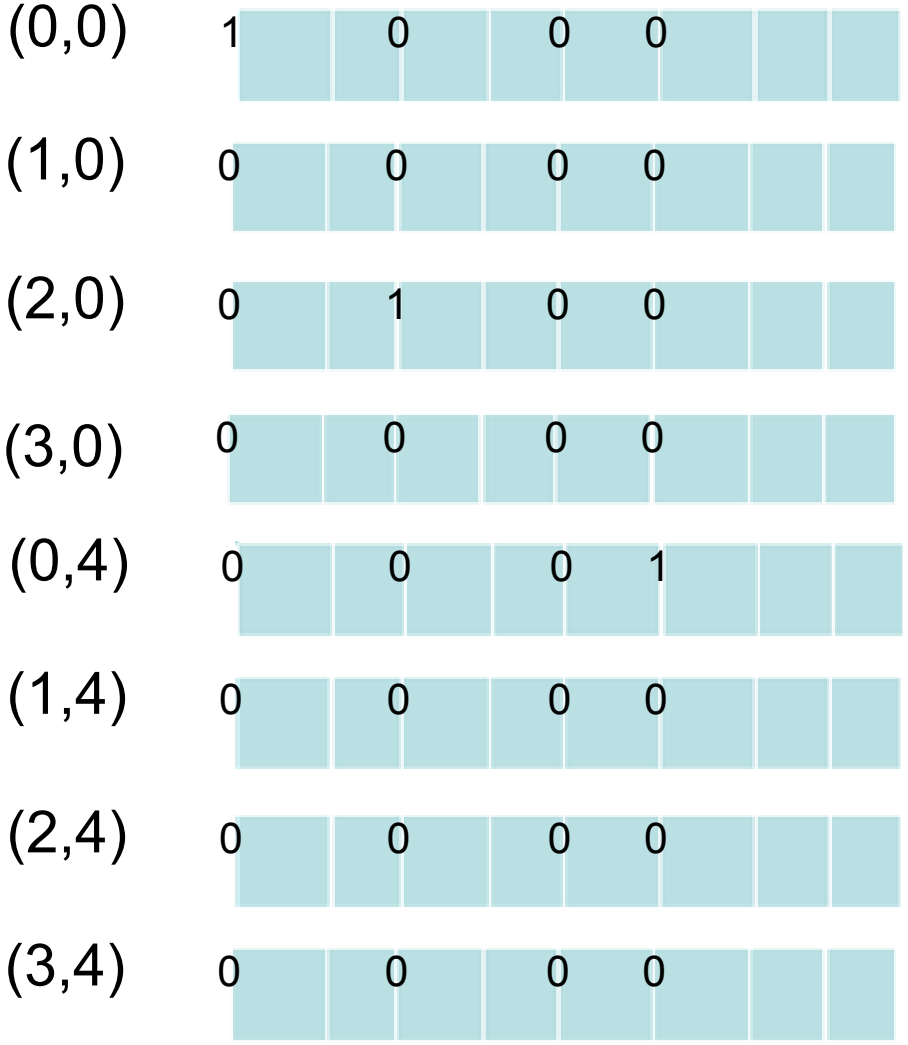
S	P	O
0	0	0
0	1	1
0	2	2
0	3	3
4	0	5
4	1	6
4	2	7
4	3	3

nSO = 8
nP = 4



PSIndex

Eight composite keys



Note: Bit vectors are further compressed with FastBit.

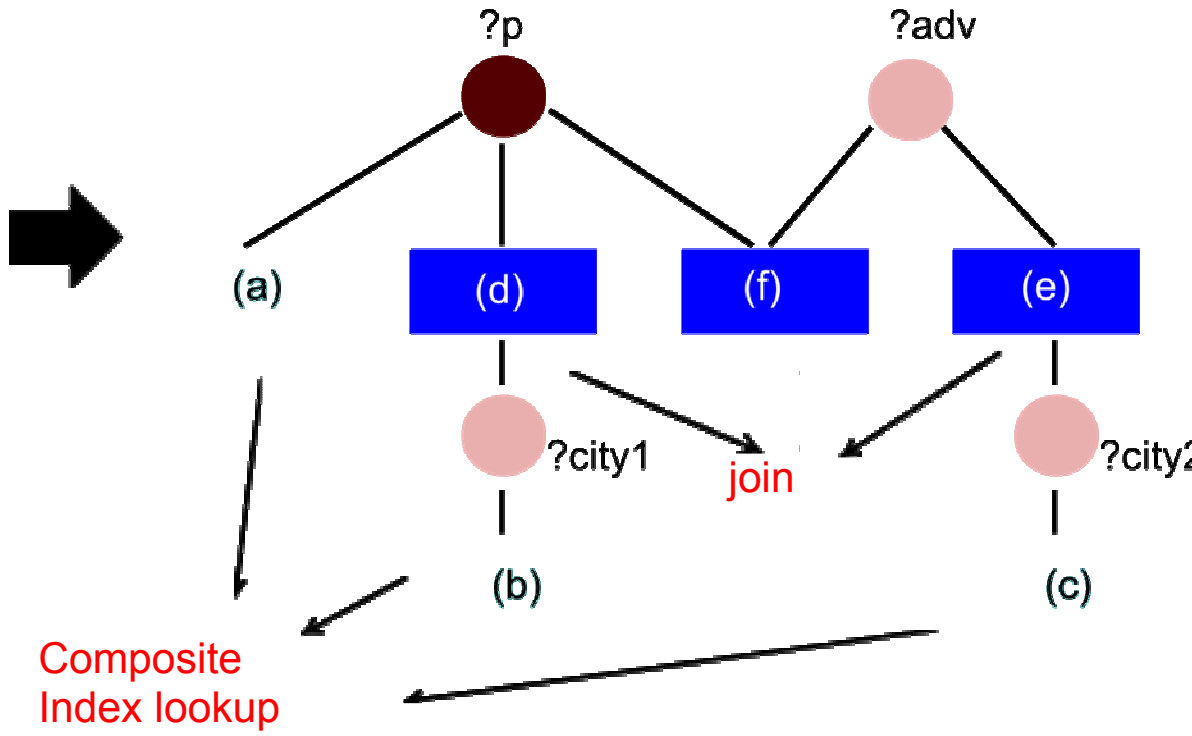
Answering a SPARQL Query with Bitmap Indexes

Example Search Query: list of all scientists born in a city in USA, who have/had a doctoral advisor born in Chinese city.

Query in SPARQL

Select ?p where {
1. (a)?p <type> 'scientist' .
(b) ?city1 <locatedIn> 'USA' .
(c) ?city2 <locatedIn> 'China' .
(d) ?p <bornInLocation> ?city1 .
(e) ?adv <bornInLocation> ?city2 .
(f) ?p <hasDoctoralAdvisor> ?adv .

Query Graph



The ordering of bit vector operations determines query work performed.

Index Size Comparison

Data Set #triples	LUBM 1M	LUBM 50M	Yago 40M	UniProt 220M
Raw data (GB)	0.125	6.27	3.56	30.58
FastBit dictionaries (GB)	0.032	0.79	1.30	3.05
FastBit Indexes (GB)	0.016	1.59	1.20	6.30
RDF-3X (GB)	0.058	2.83	2.75	---

- FastBit indexes 1.78-3.6X smaller than RDF-3X B-tree based index for various data sets.
- FastBit indexes are much smaller than the raw data.

Performance Results: LUBM Benchmark

LUBM/50M records SPARQL test query evaluation time in milliseconds, 'warm caches' performance on a 2.67 GHz Intel Xeon system.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
FastBit	0.167	1311	0.92	0.40	0.19	135	0.46
RDF-3X	0.31	544	0.193	0.70	1.95	4021	1.52
Speedup	1.86X	.42X	0.21X	1.75X	10.3X	**29.8X**	3.3X

	Q8	Q9	Q10	Q11	Q12	Q13	Q14
FastBit	6.34	9288	0.179	0.148	2.34	0.34	467
RDF-3X	50.4	1369	0.336	0.35	7.44	1.7	13770
Speedup	**7.95X**	.15X	1.87X	2.36X	3.17X	5.0X	**29.5X**

```
select ?x ?y ?z where {
?x ub:subOrganizationOf <http://www.University0.edu> .
?x rdf:type ub:Department .
?x ub:memberOf ?y .
?x rdf:type ub:UndergraduateStudent .
?x ub:emailAddress ?z .
}
```

```
select ?x where {
?x rdf:type ub:UndergraduateStudent .
}
```

Performance Results: Summary

FastBit query evaluation performance improvement achieved (geometric mean of individual query speedup) over RDF-3X for various data sets.

	LUBM-5M	LUBM-50M	LUBM-500M	Yago-40M
Speedup	12.96X	2.62X	2.81X	1.38X

Conclusions

- •We utilize compressed bitmap indexes to accelerate RDF SPARQL queries
- •Our new approach is 1.4-13X faster than RDF-3X, a state-of-the-art RDF storage and retrieval system.

Future Work

- •Develop join indexes for SPARQL queries
- •Automate SPARQL query parsing and evaluation
- •Speed up index and dictionary creation
- •Support incremental index updates

Thank you!

Questions?

Information about FastBit
http://sdm.lbl.gov/fastbit/