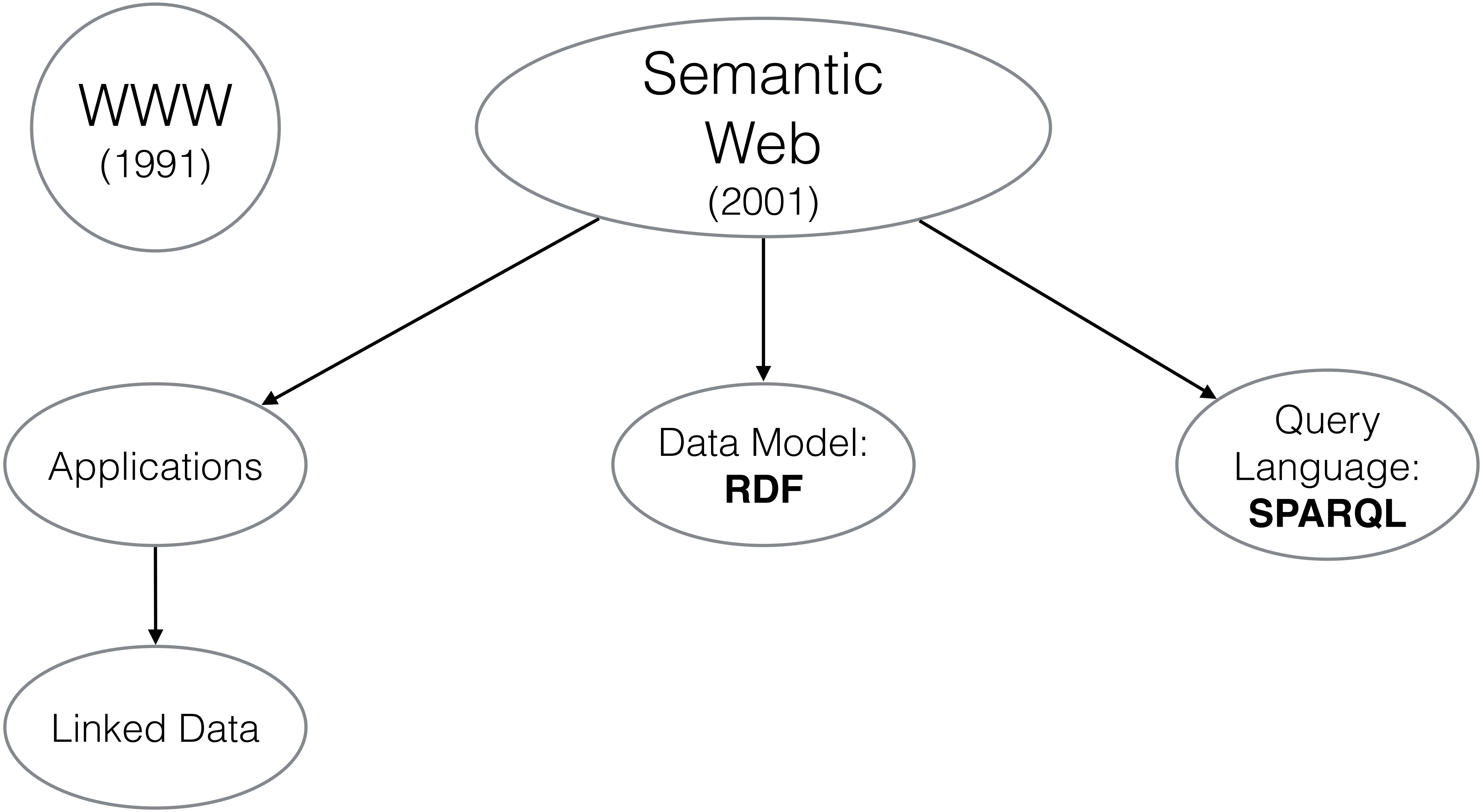
Fast Processing of SPARQL Queries on RDF Quadruples

Vasil Slavov, Anas Katib, Praveen Rao,
Srivenu Paturi, Dinesh Barenkala
University of Missouri-Kansas City

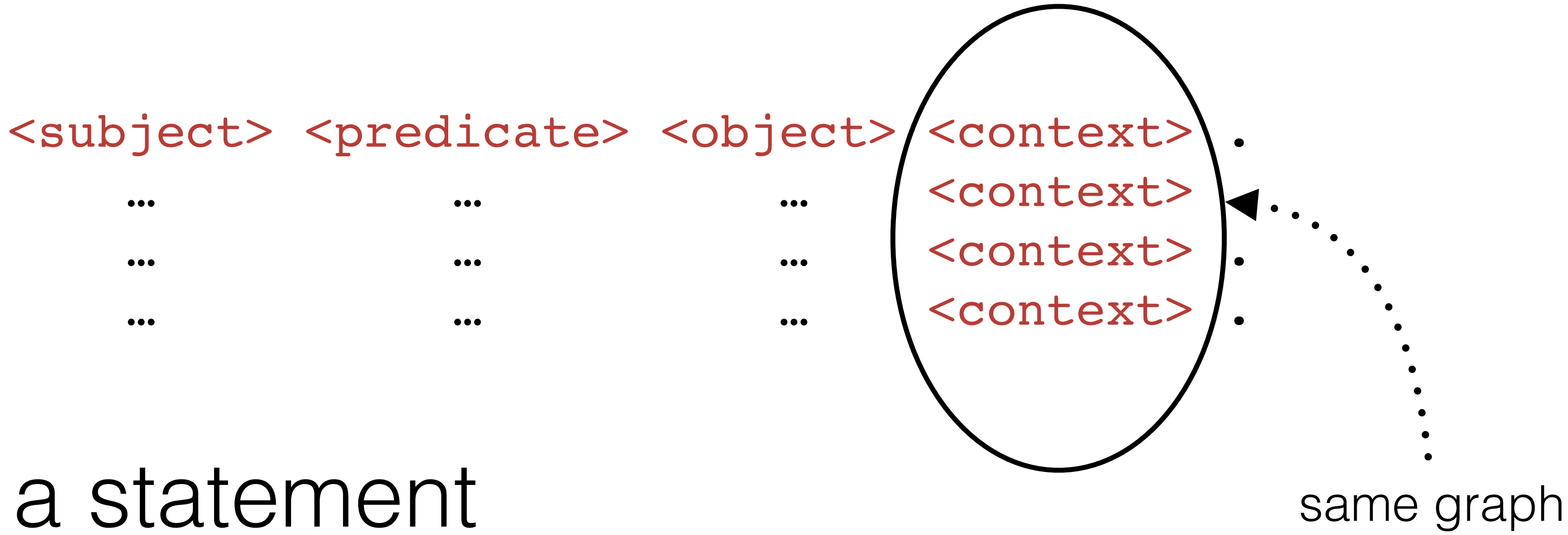
WebDB 2014

Acknowledgements
National Science Foundation (IIS-1115871)

Semantic Web



Quads



```
<subject> <predicate> <object> <context> .
    …           …          …     <context>
    …           …          …     <context> .
    …           …          …     <context> .
```

- Origin of a statement

```
foaf:me foaf:name "Alice" <http://ex.org/alice/foaf.rdf> .
foaf:me foaf:name "Bob" <http://ex.org/bob/foaf.rdf> .
```

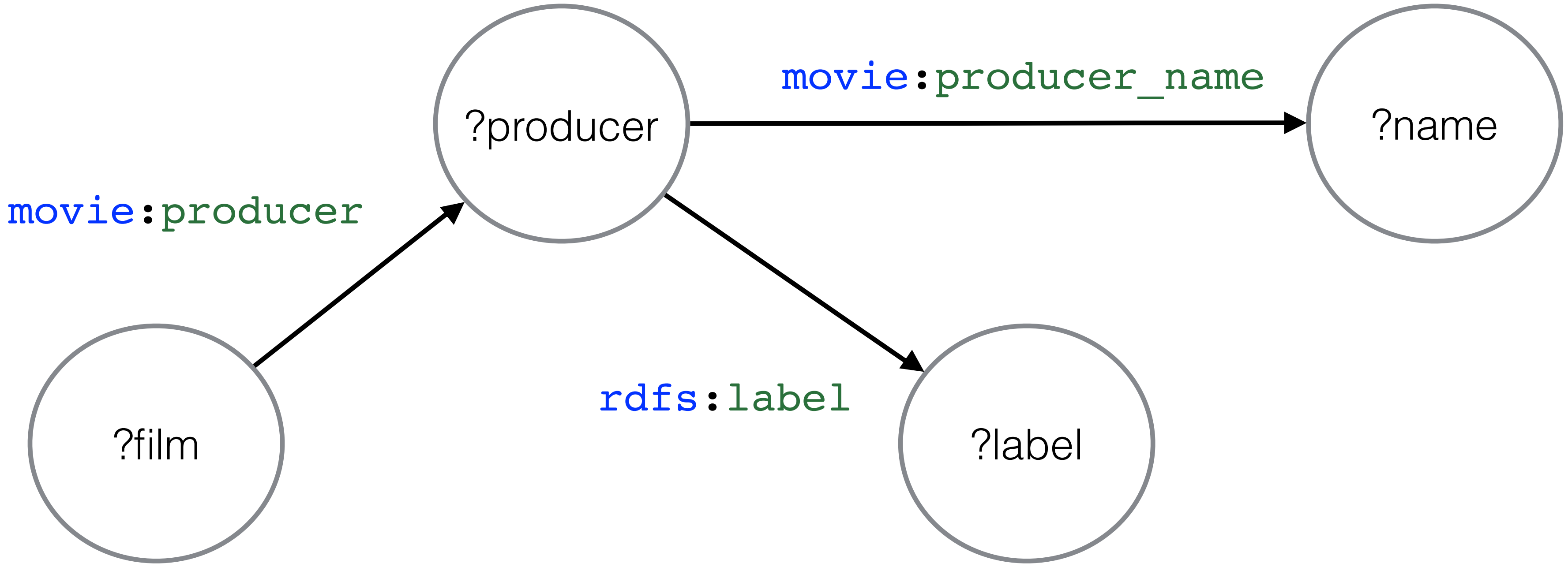
- Differentiate b/w identical statements

```
foaf:alice foaf:knows foaf:bob <http://ex.org/graphs/john> .
foaf:alice foaf:knows foaf:bob <http://ex.org/graphs/james> .
```

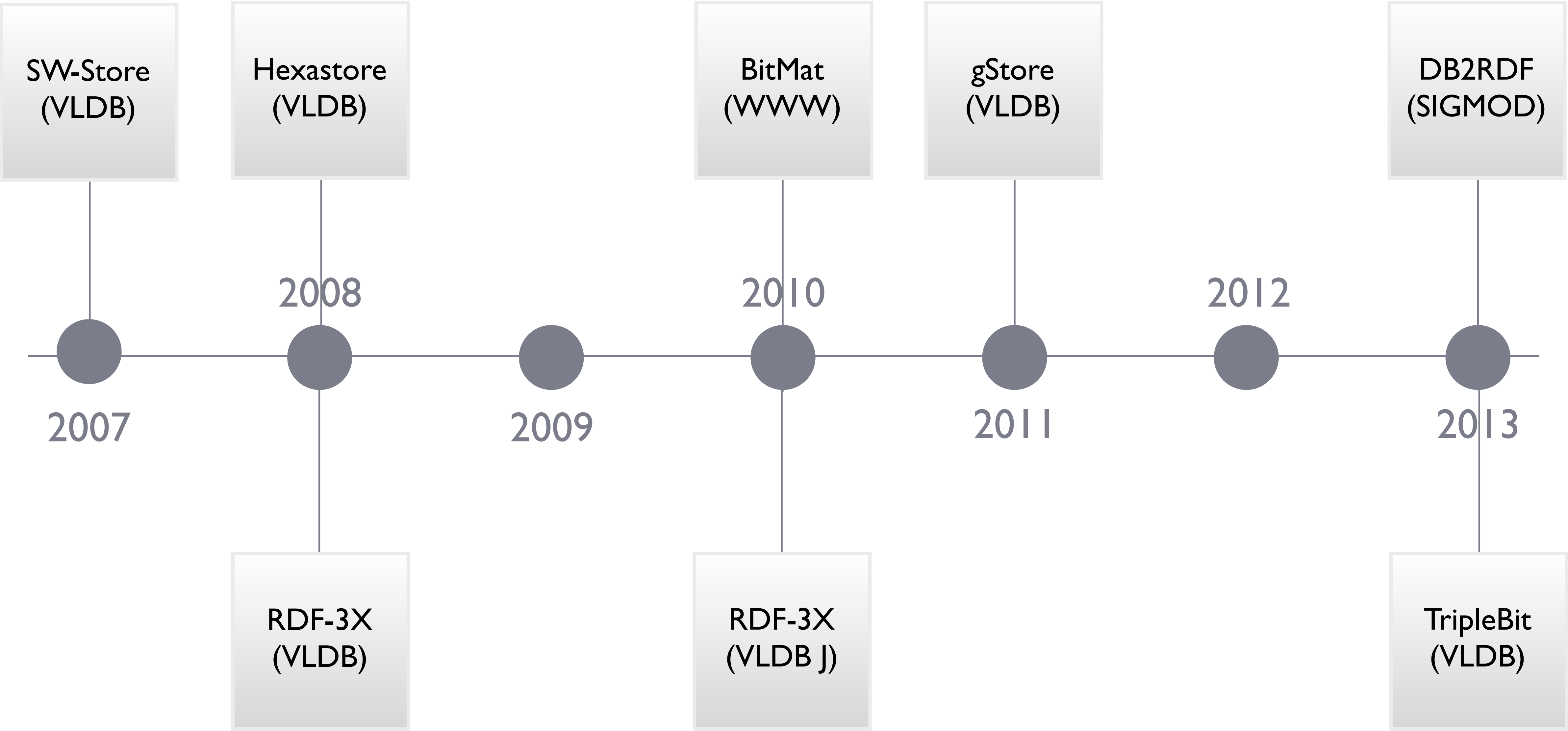
GRAPH query

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX movie: <http://data.linkedmdb.org/resource/movie/>

SELECT ?g ?producer ?name ?label ?page ?film WHERE {
    GRAPH ?g {
        ?producer movie:producer_name ?name .
        ?producer rdfs:label ?label .
        ?film movie:producer ?producer .
    }
}
```



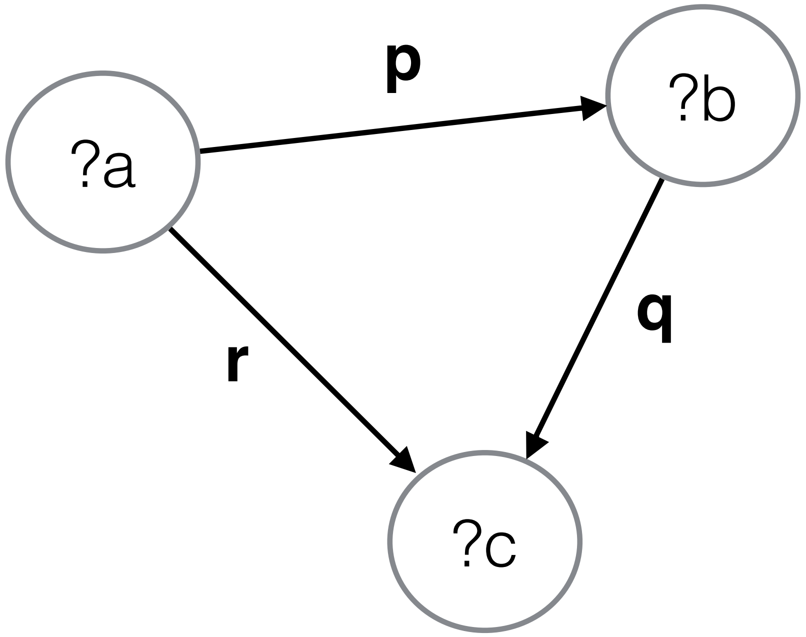
Related work



What's missing in them?

1. No support for quads
2. No large BGP queries (over 8 triple patterns)
3. No complex BGP queries (undirected cycles):

```
SELECT * WHERE {
  ?a p ?b .
  ?b q ?c .
  ?a r ?c .
}
```



Why not use triple stores for quads?

INCORRECT RESULTS

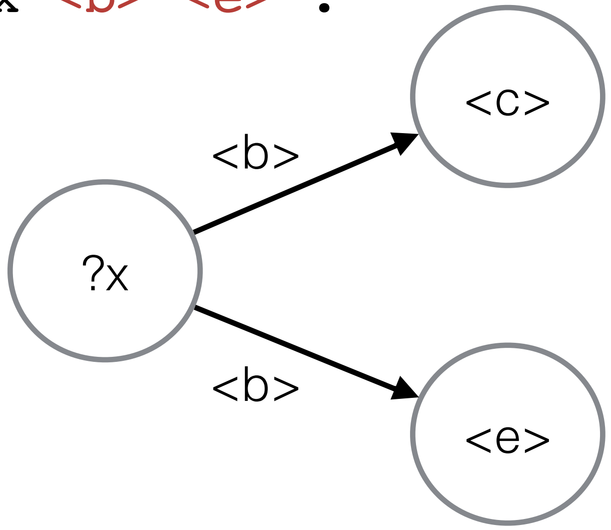
Triple vs. Quad

```
<a> <b> <c> <g1> .
<a> <b> <e> <g2> .
```

Data

Query

```
SELECT ?x WHERE {
  GRAPH ?g {
    ?x <b> <c> .
    ?x <b> <e> .
  }
}
```



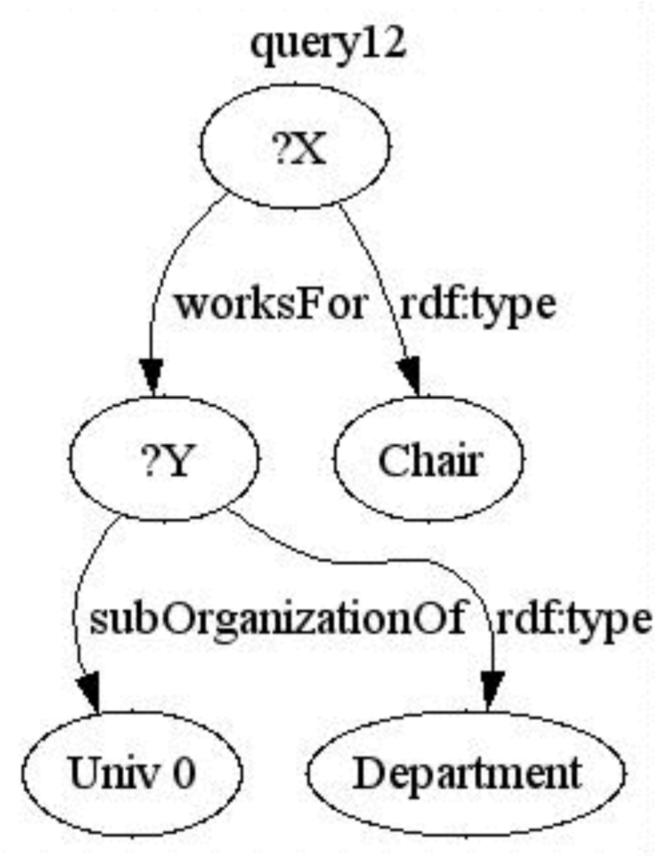
`<a>`

Triple store results

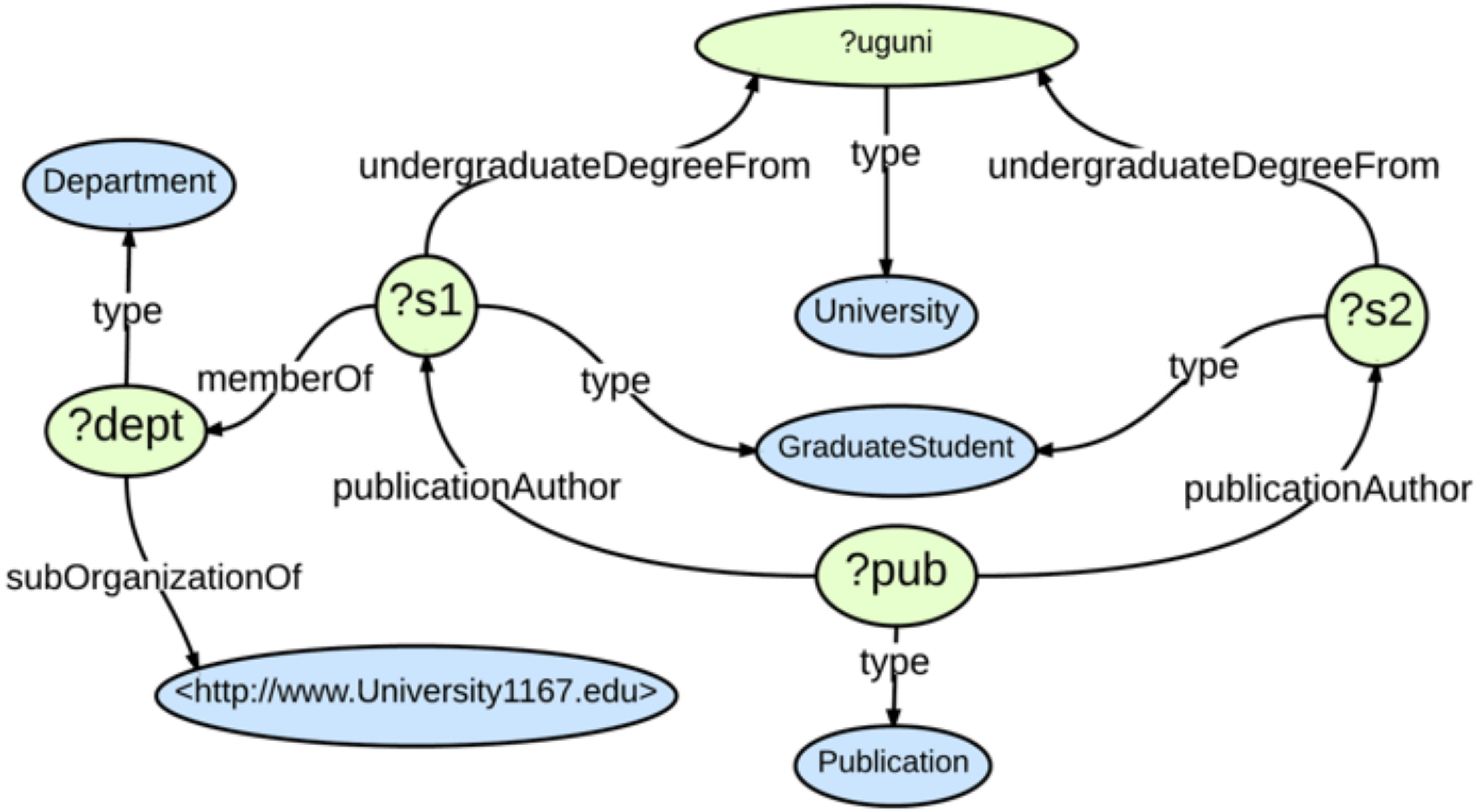
Quad store results

`<empty>`

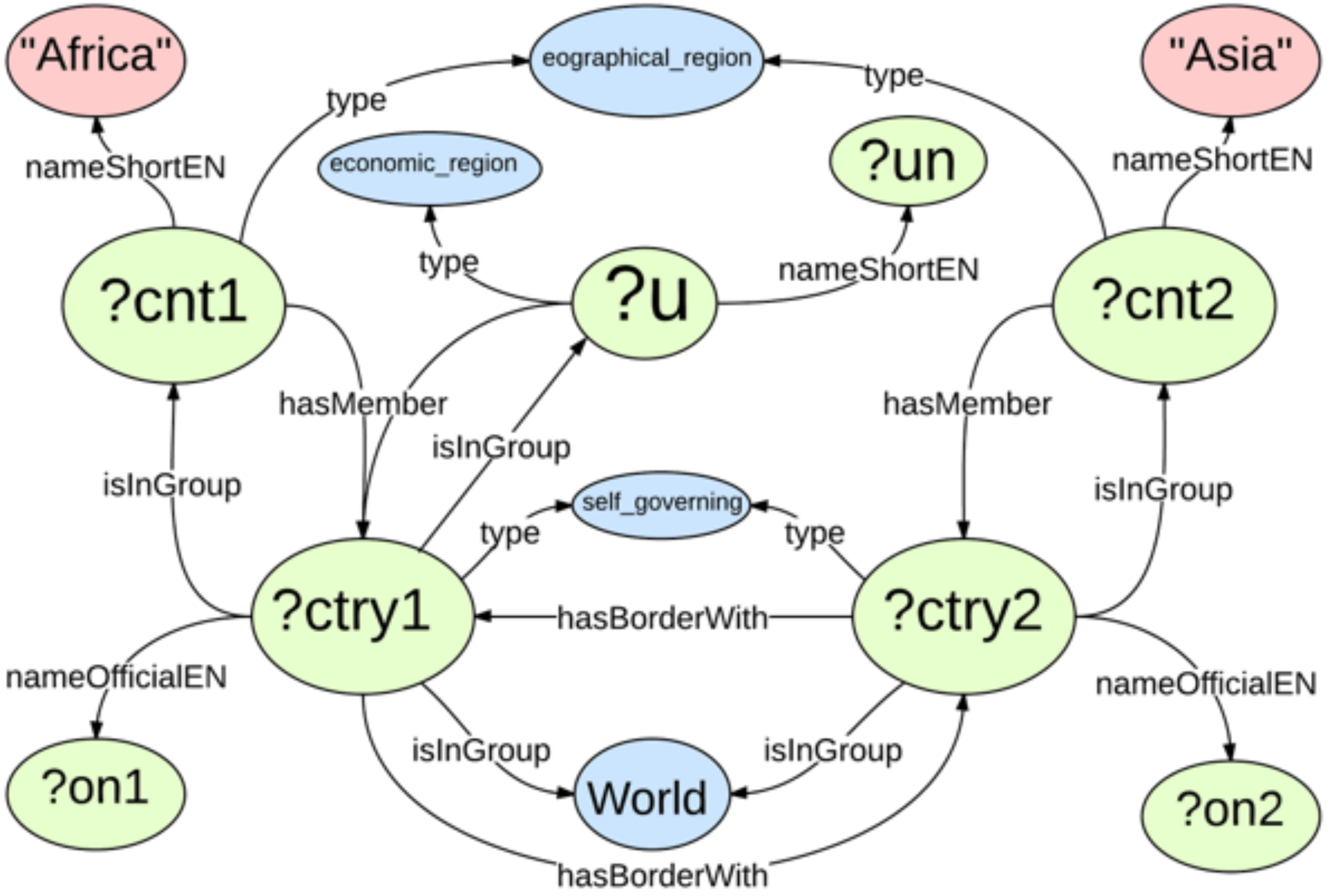
State-of-the-art technologies are… fast



State-of-the-art technologies are… slow



State-of-the-art technologies are… really slow

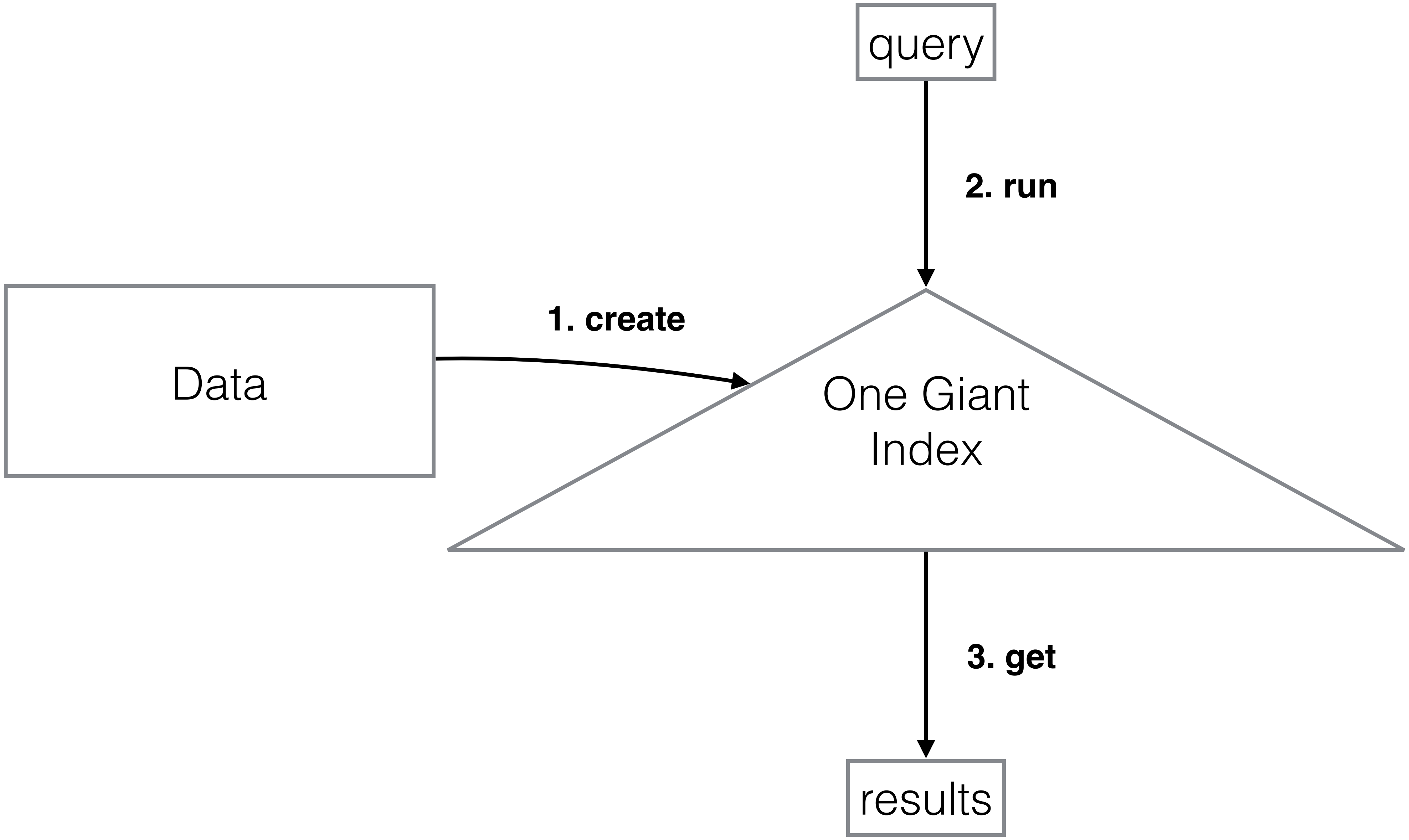


Comparison

	quads	max triples/ quads	max triple patterns
RIQ	yes	1.38B	22
RDF-3X	no	845M	13
BitMat	no	1.33B	8
Jena TDB	yes	333M	6
DB2RDF	no	333M	6
TripleBit	no	2.95B	12

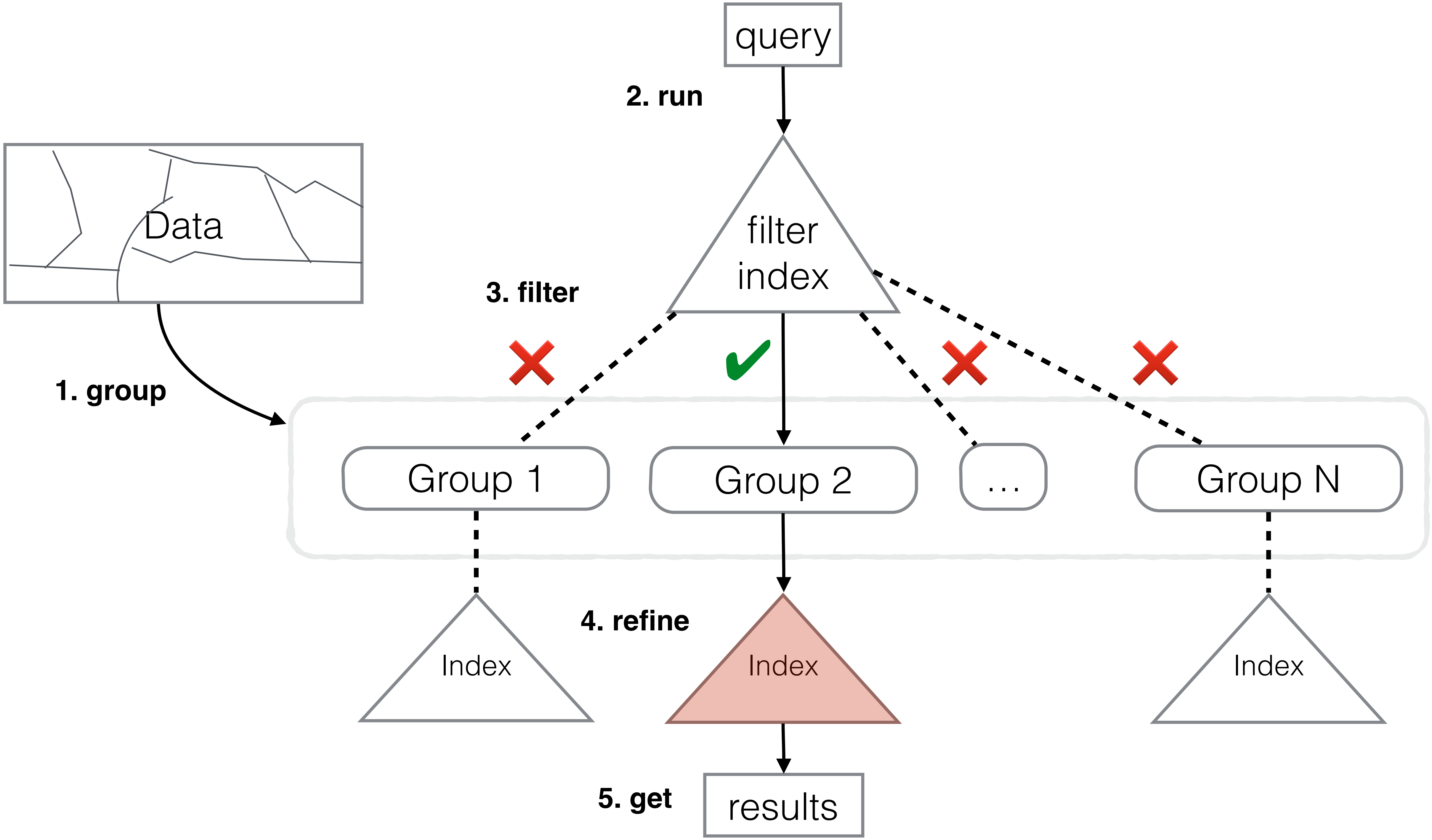
Query processing

(traditional)

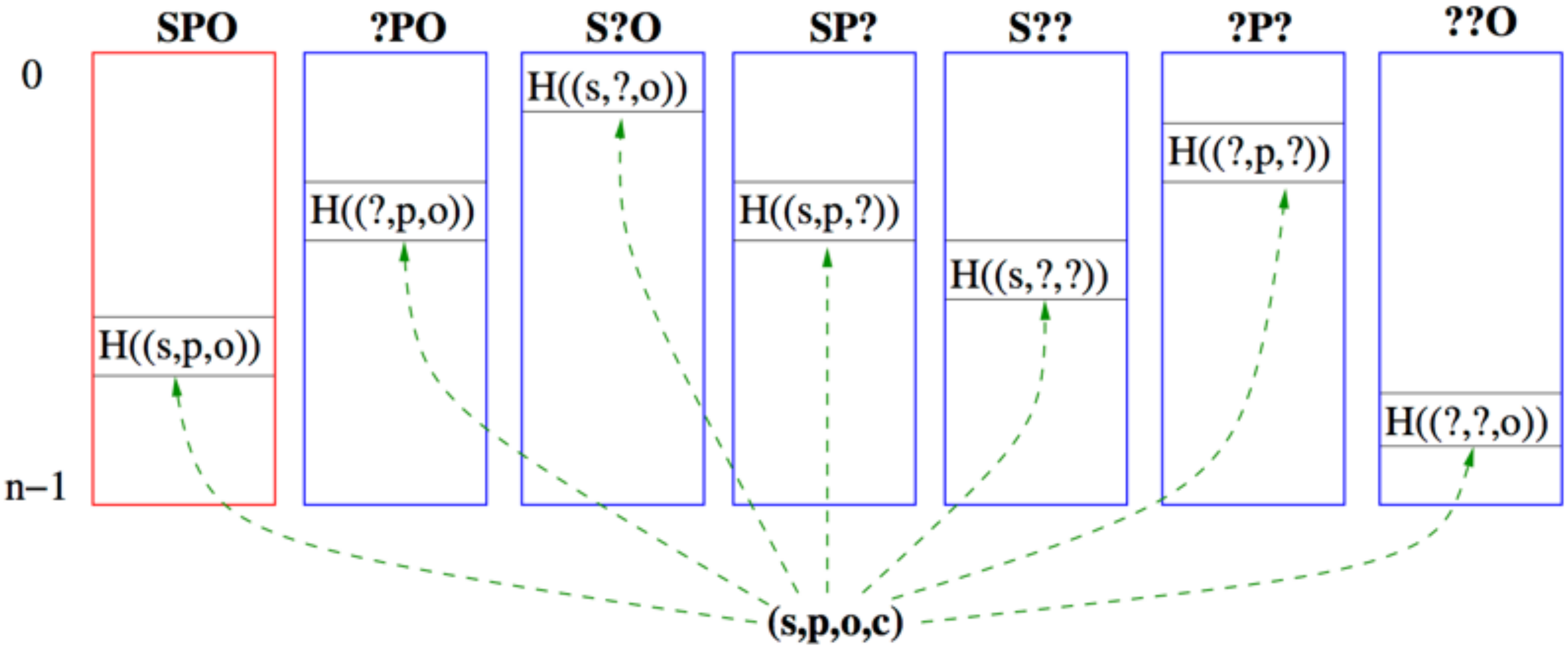


Query processing

(our ‘decrease-and-conquer’ approach)



Pattern Vectors (PVs)



$\mathbb{H} : B \to \mathbb{Z}^*$

$\mathbb{P} = \{SPO, SP?, S?O, ?PO, S??, ?P?, ??O\}$

Filter Index construction

Steps:

1. Create groups of similar PVs

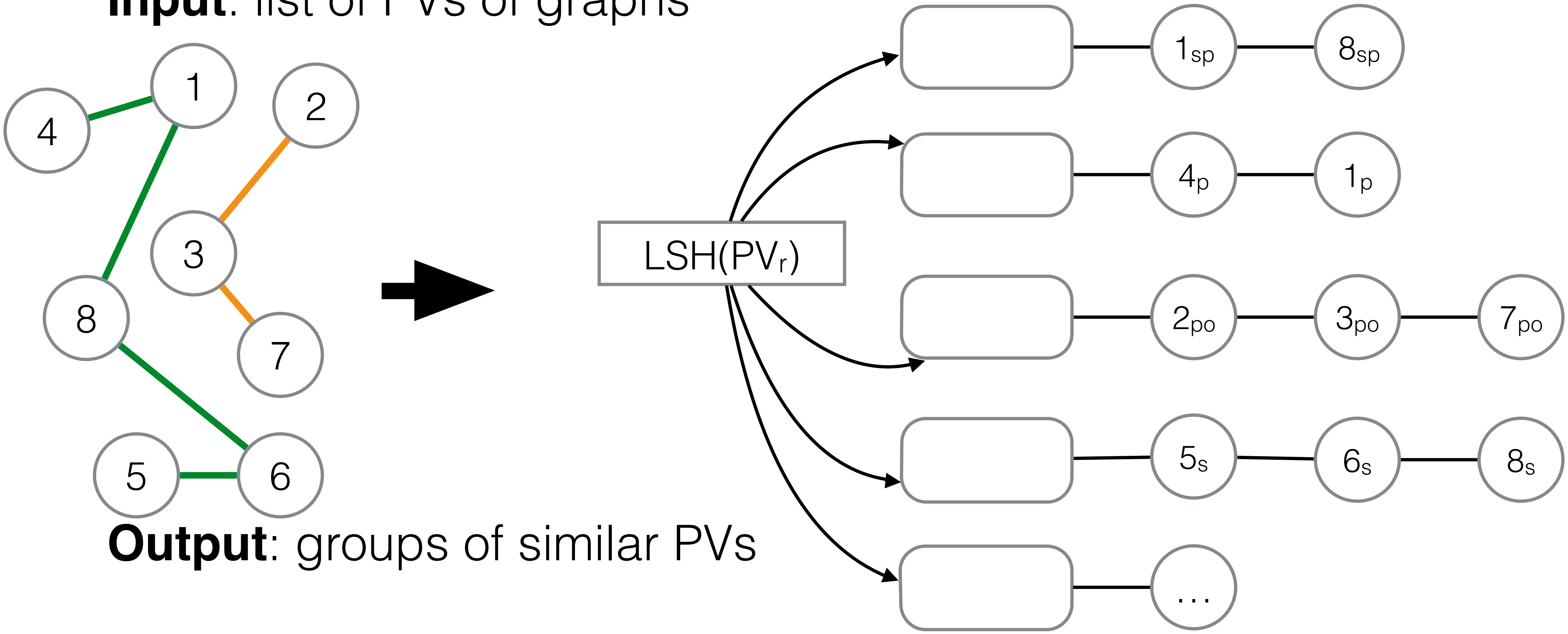
   Locality Sensitive Hashing

2. Compactly store Filter Index

   Bloom Filters and Counting Bloom Filters

Grouping PVs

Input: list of PVs of graphs



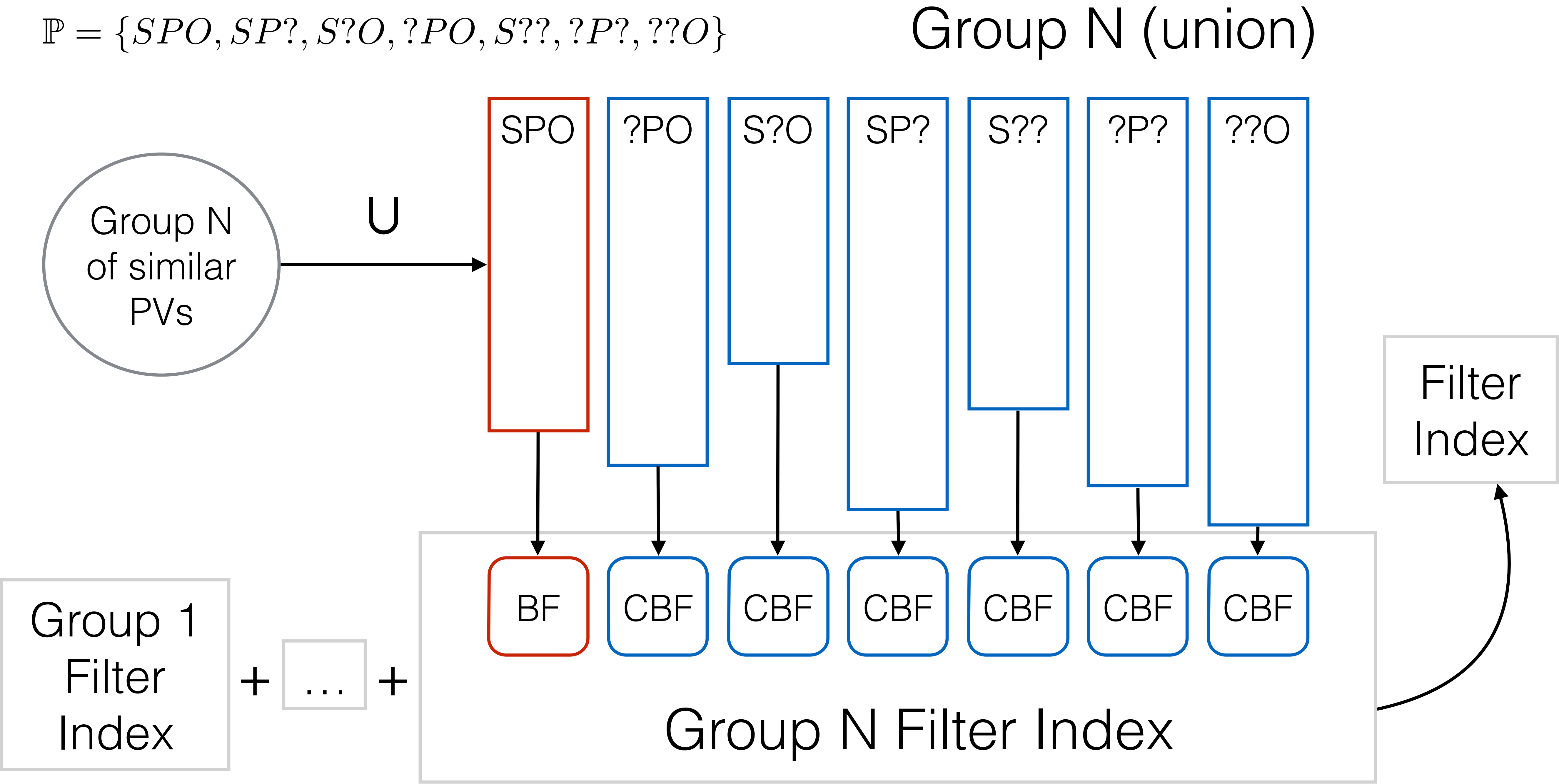
Output: groups of similar PVs

$$sim(PV_a, PV_b) = \max_{r \in \mathbb{P}} \frac{|PV_{a,r} \cap PV_{b,r}|}{|PV_{a,r} \cup PV_{b,r}|} \qquad \mathbb{P} = \{SPO, SP?, S?O, ?PO, S??, ?P?, ??O\}$$

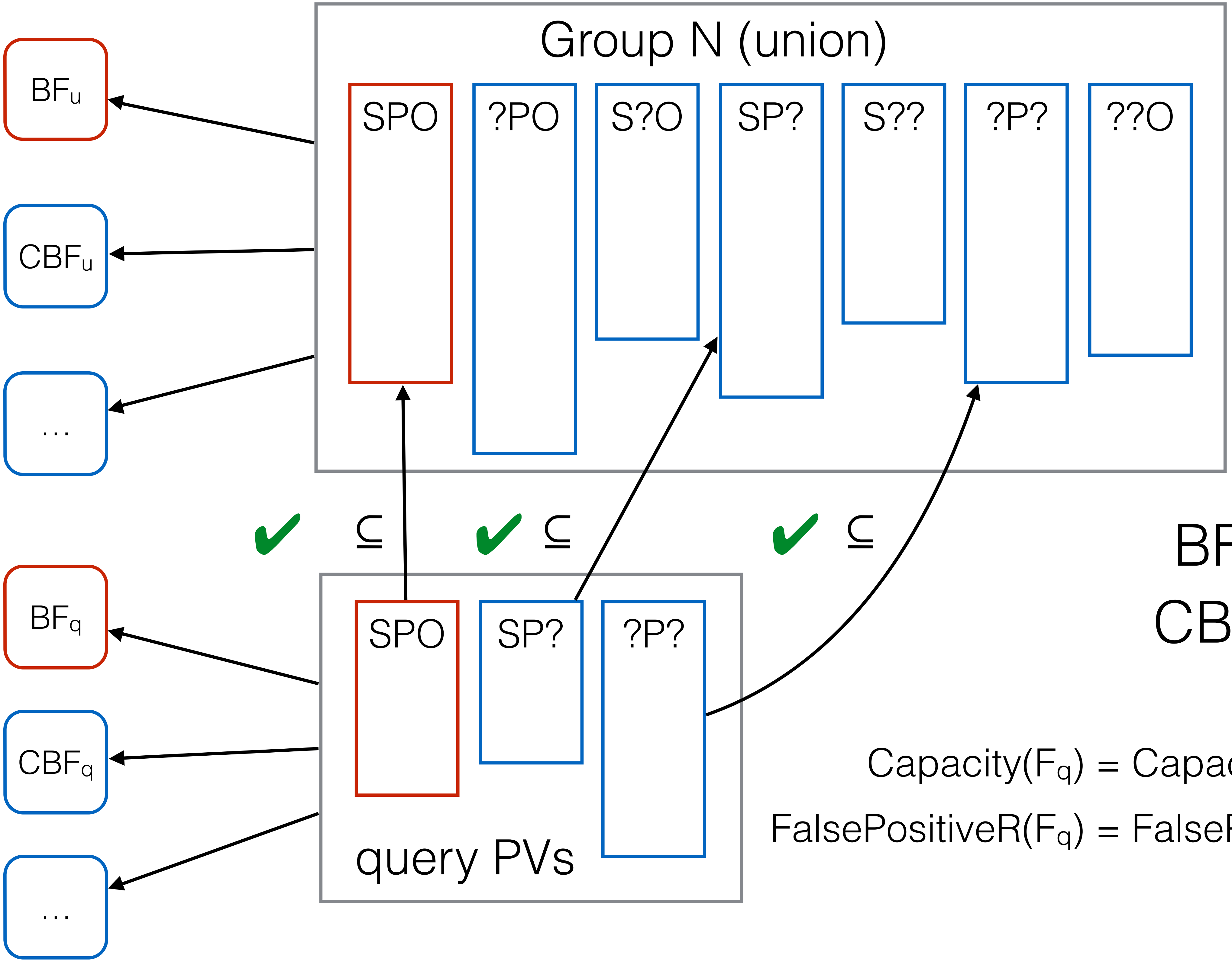
Filter Index

$PV_{c,r} \leftarrow PV_{a,r} \cup PV_{b,r}\ and\ r \in \mathbb{P}$

$\mathbb{P} = \{SPO, SP?, S?O, ?PO, S??, ?P?, ??O\}$



Query execution



$$BF_q[i] == BF_u[i]$$

$$CBF_q[i] \leq CBF_u[i]$$

$\text{Capacity}(F_q) = \text{Capacity}(F_u)$

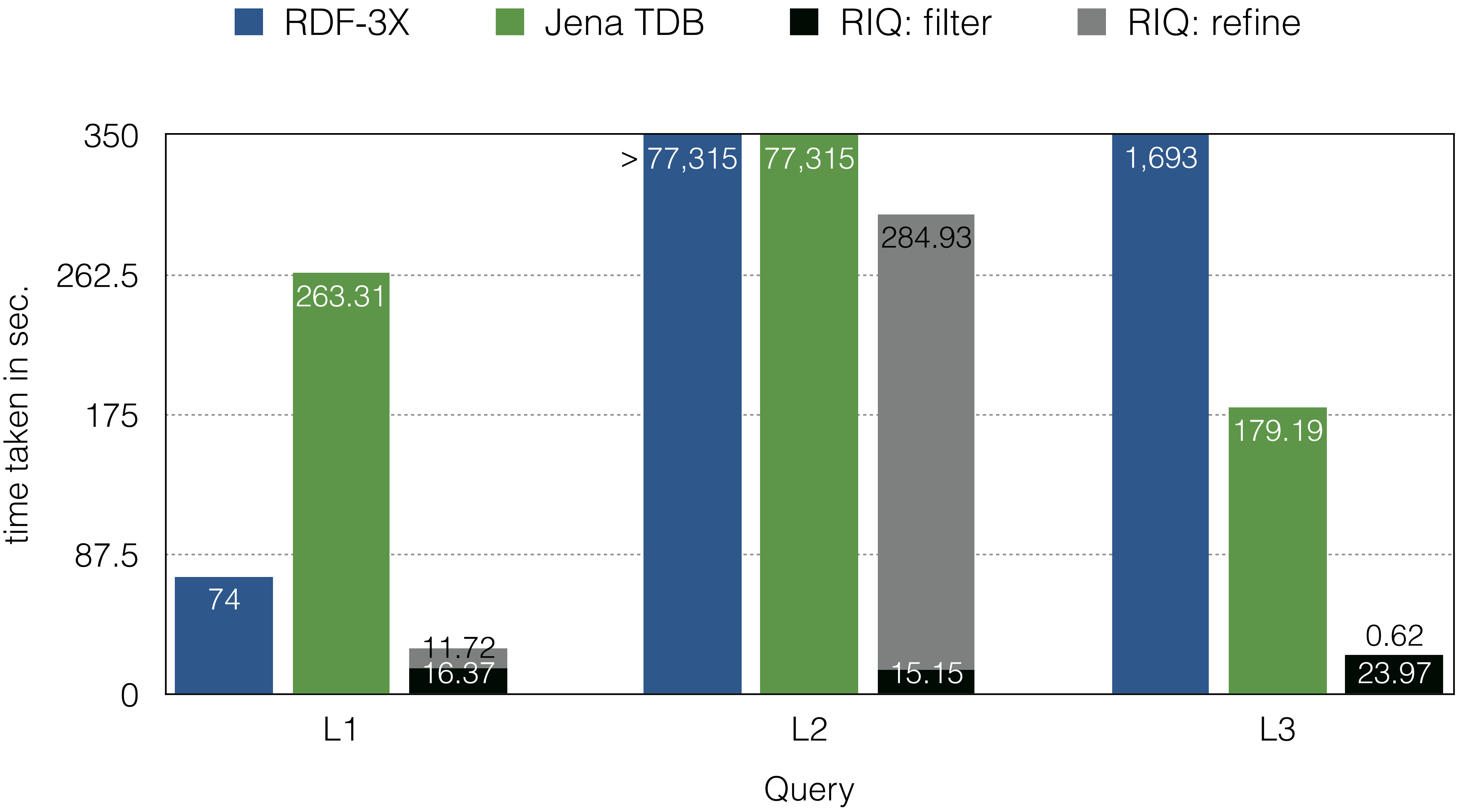
$\text{FalsePositiveR}(F_q) = \text{FalsePositiveR}(F_u)$

Initial performance evaluation

- Datasets
  - Synthetic: LUBM [Web Semantics ‘05], 1.38 billion triples
  - Real: BTC-2012 [http://challenge.semanticweb.org], 1.36 billion quads
- Queries with single BGP
  - Large: up to 22 patterns
  - Small: up to 8 patterns

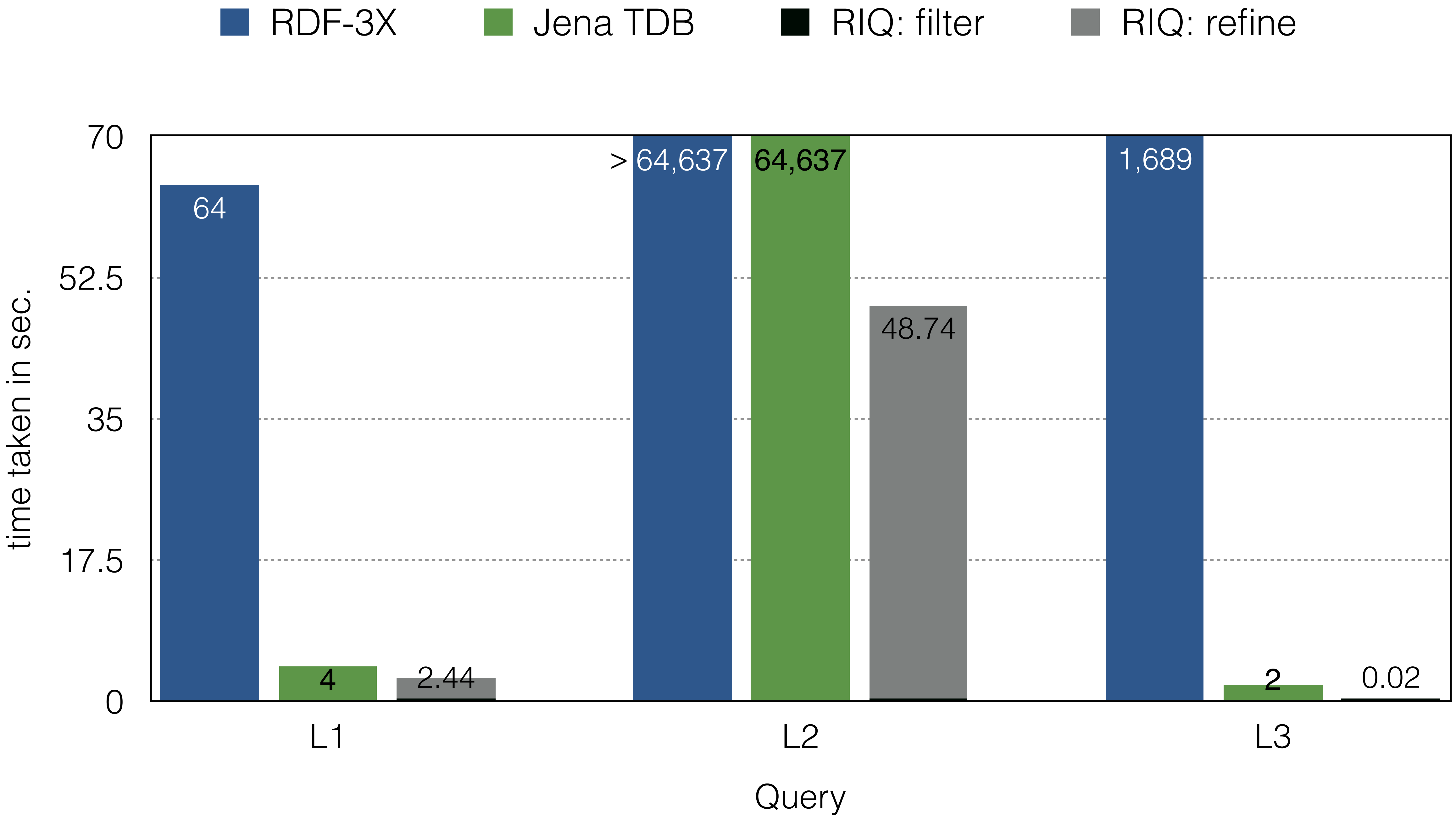
Large BGPs

(LUBM, cold cache)



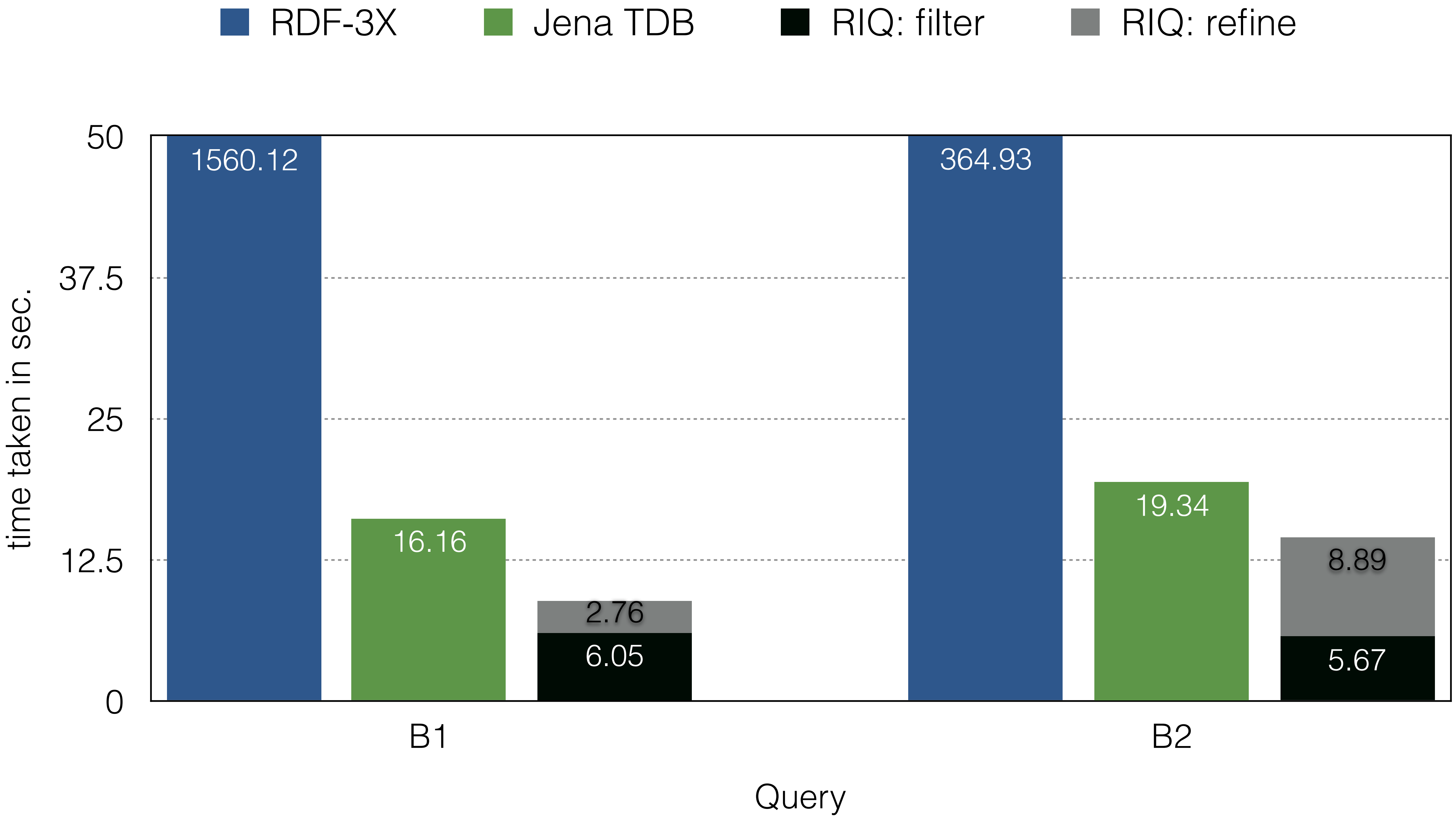
Large BGPs

(LUBM, warm cache)



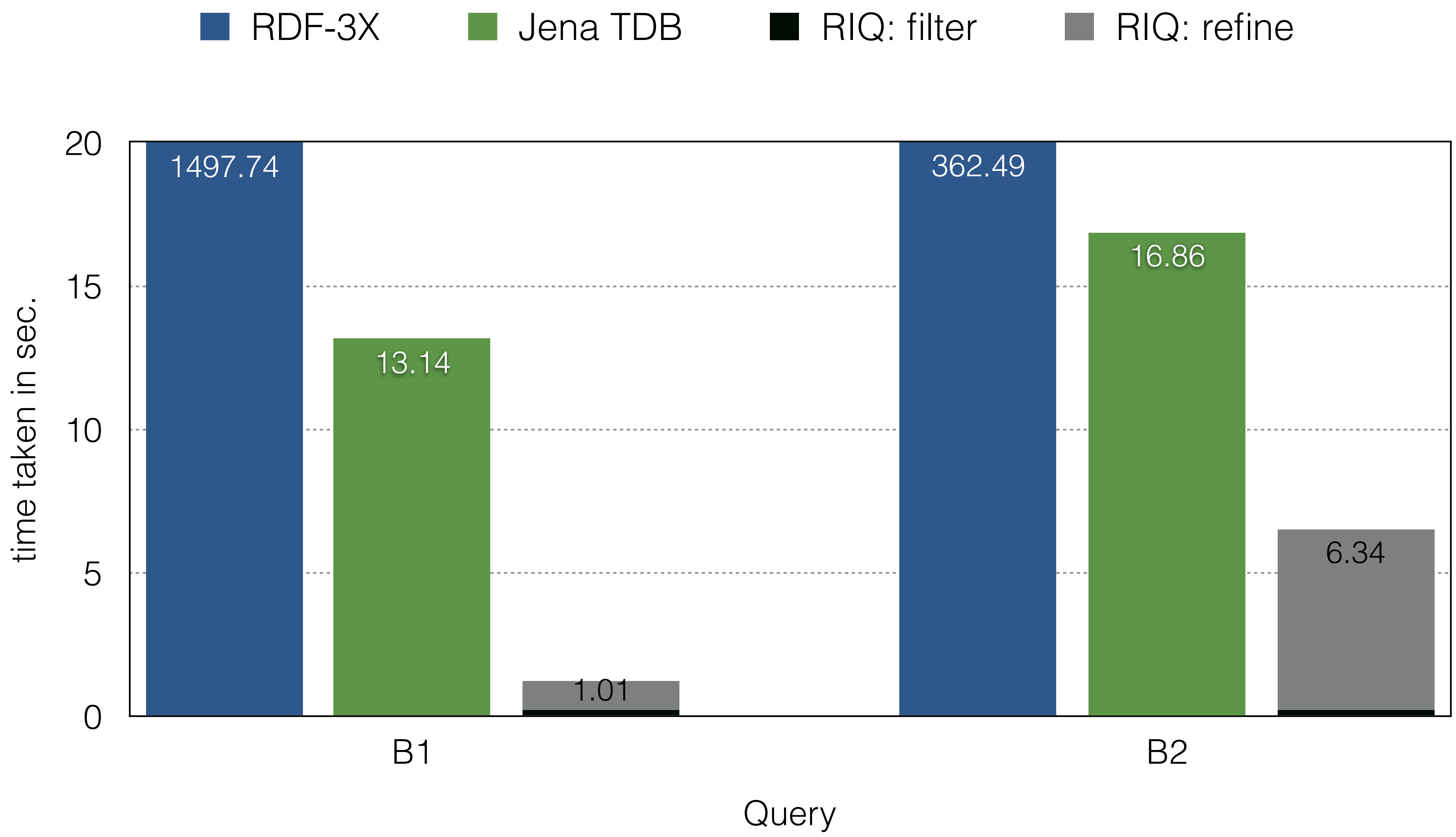
Large BGPs

(BTC-2012, cold cache)



Large BGPs

(BTC-2012, warm cache)



Small BGPs

(LUBM)

	Cold cache			Warm cache		
Query	**RIQ**	**RDF-3X**	**Jena TDB**	**RIQ**	**RDF-3X**	**Jena TDB**
L4	229.95	1986.21	698.08	27.46	1899.1	664.75
L5	576.96	995.26	1130.43	567.2	948.53	1127.37
L6	506.93	888.84	1119.31	489.36	847.59	1144.11
L7	892.7	1215.53	aborted	871.12	1153.31	aborted
L8	507.43	805.41	1346.17	497.69	70.35	1395.48
L9	538.99	979.79	1137.38	519.22	947.07	1142.73
L10	18.72	11.11	7.15	0.51	6.39	3.19
L11	12.19	1.98	5.79	0.41	0.25	1.13
L12	103.14	22.33	725.93	26.76	19.83	703.26
Geo. mean	**193.85**	**210.97**	**282.57**	**59.68**	**115.7**	**207.72**

Small BGPs

(BTC-2012)

	Cold cache			Warm cache		
Query	**RIQ**	**RDF-3X**	**Jena TDB**	**RIQ**	**RDF-3X**	**Jena TDB**
B3	41.01	56.42	373.59	1.83	0.82	20.13
B4	42.17	48.55	321.56	3.59	2.37	35.99
B5	70.15	74.86	3541.99	32.38	28.64	3540.28
B6	20.39	> 40,140	14.89	0.64	> 40,140	12.83
B7	221.86	210.37	1925.27	184.86	118.84	1817.85
Geo. mean	**55.96**	**280.34**	**414.25**	**7.59**	**48.4**	**143.01**

Queries with multiple BGPs

- Keywords like UNION, EXISTS, OPTIONAL, etc.
- See paper for more details

Q&A

Acknowledgements:

- National Science Foundation (IIS-1115871)