IIF
INSTITUT FÜR
INFORMATIK
FREIBURG

Optimization of SPARQL Queries Processing Using PRoST

UNI
FREIBURG

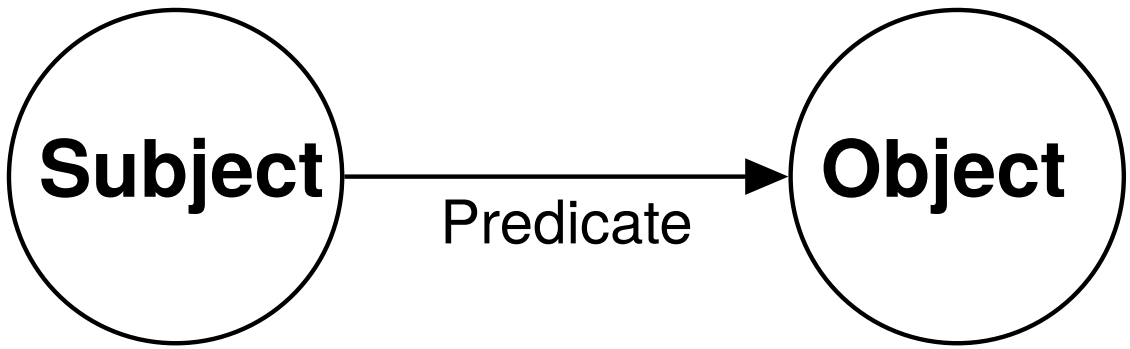
Guilherme Schievelbein

Databases and Information Systems
Master's Thesis, March 2019

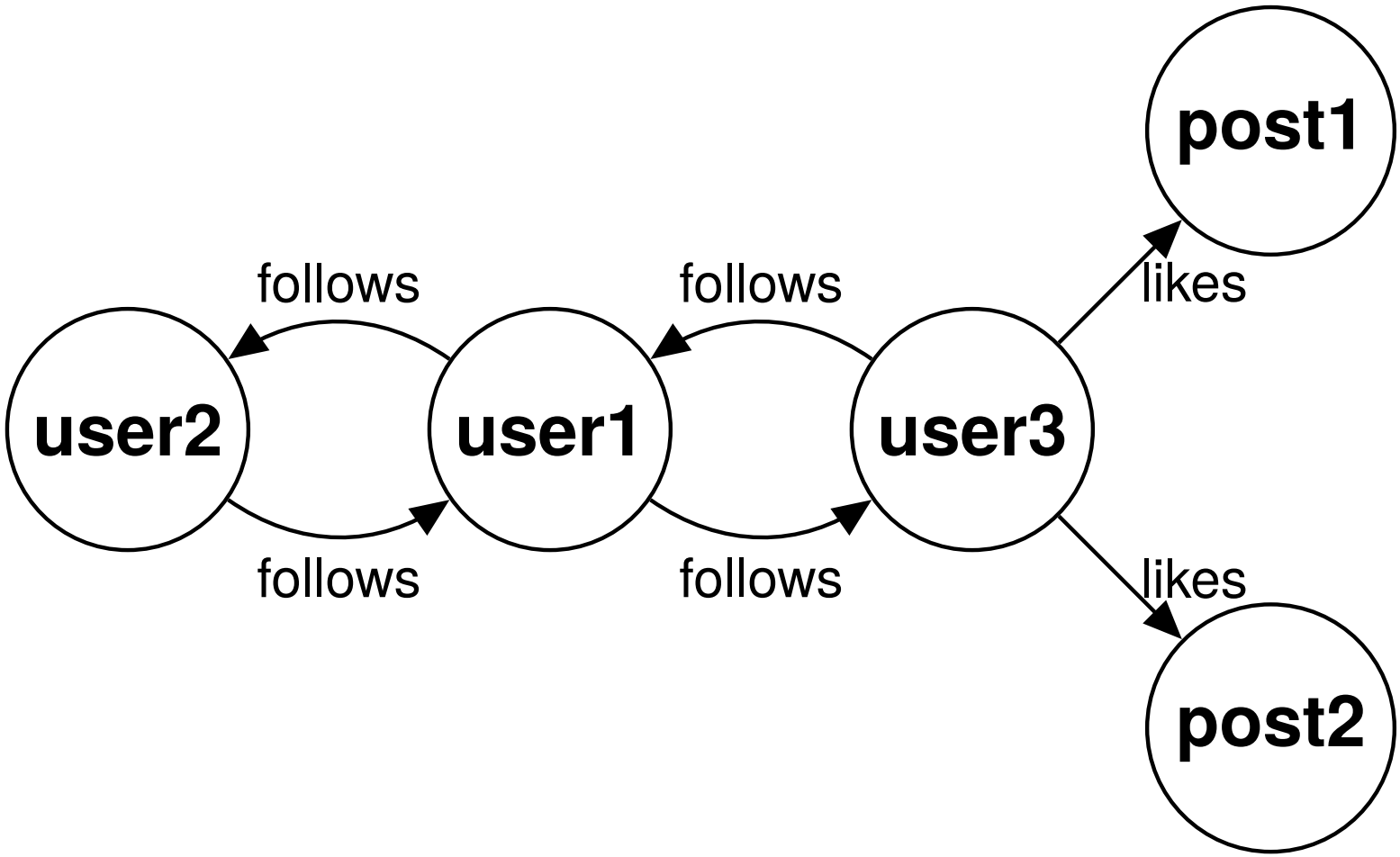
Outline

Outline

Resource Description Framework



RDF Graph


post1
follows
follows
likes
user2
user1
user3
follows
follows
likes
post2

SPARQL

```
SELECT  ?user
        ?post
WHERE {
        <user1> <follows> ?user .
        ?user   <likes>   ?post
}
```

SPARQL

```
SELECT  ?user
        ?post
WHERE {
<user1> <follows> ?user .
?user   <likes>   ?post
}
```

?user	?post
user3	post1
user3	post2

Triplestore

Triplestore		
subject	**predicate**	**object**
user1	follows	user2
user1	follows	user3
user2	follows	user1
user3	follows	user1
user3	likes	post1
user3	likes	post2

Vertical Partitioning

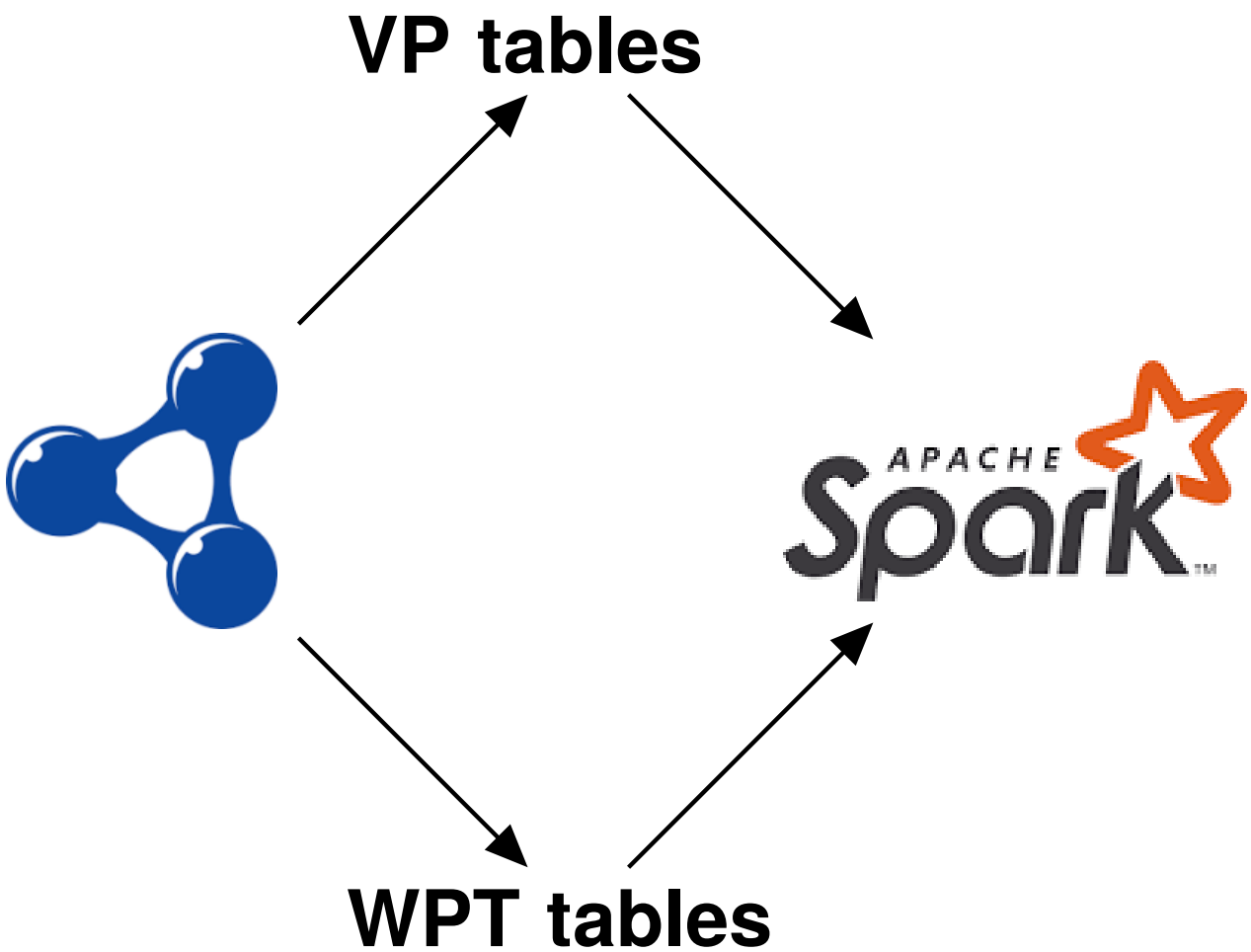
follows	
subject	**object**
user1	user2
user1	user3
user2	user1
user3	user1

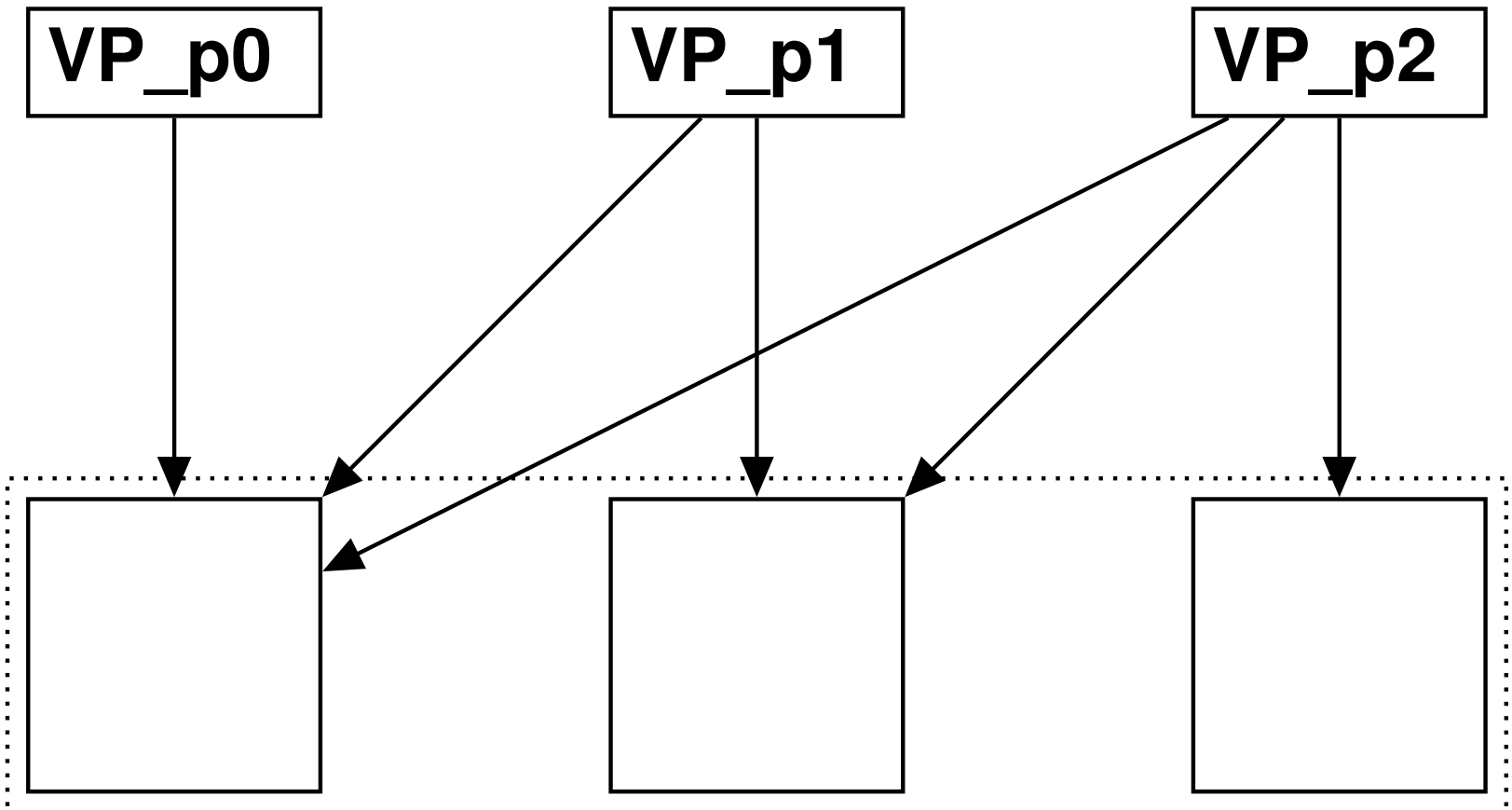
likes	
subject	**object**
user3	post1
user3	post2

Wide Property Table

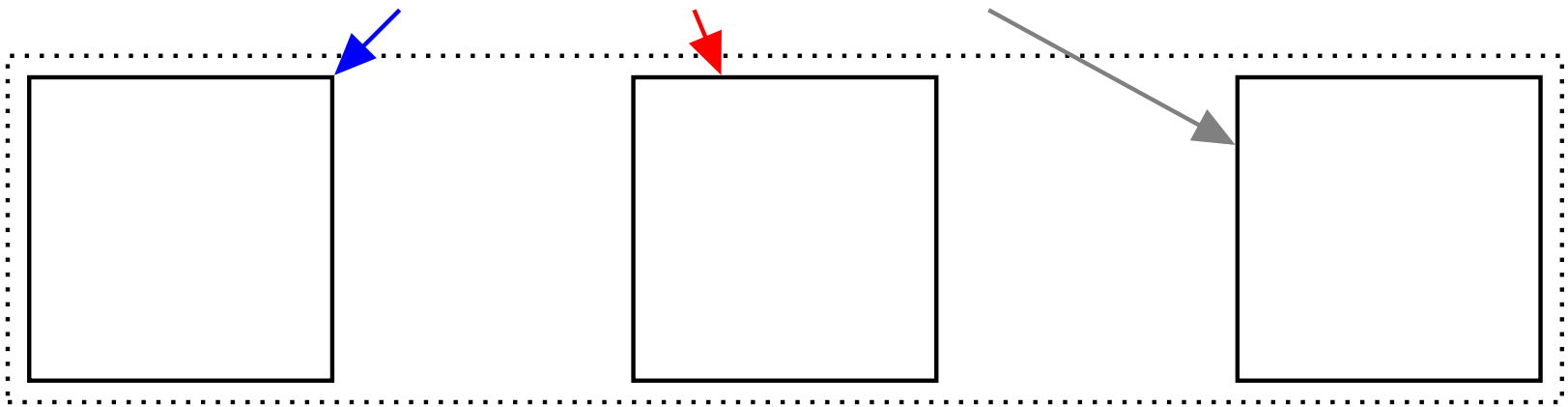
Wide Property Table		
subject	**follows**	**likes**
user1	[user2, user3]	NULL
user2	user1	NULL
user3	user1	[post1, post2]

Outline


VP tables
APACHE Spark
WPT tables

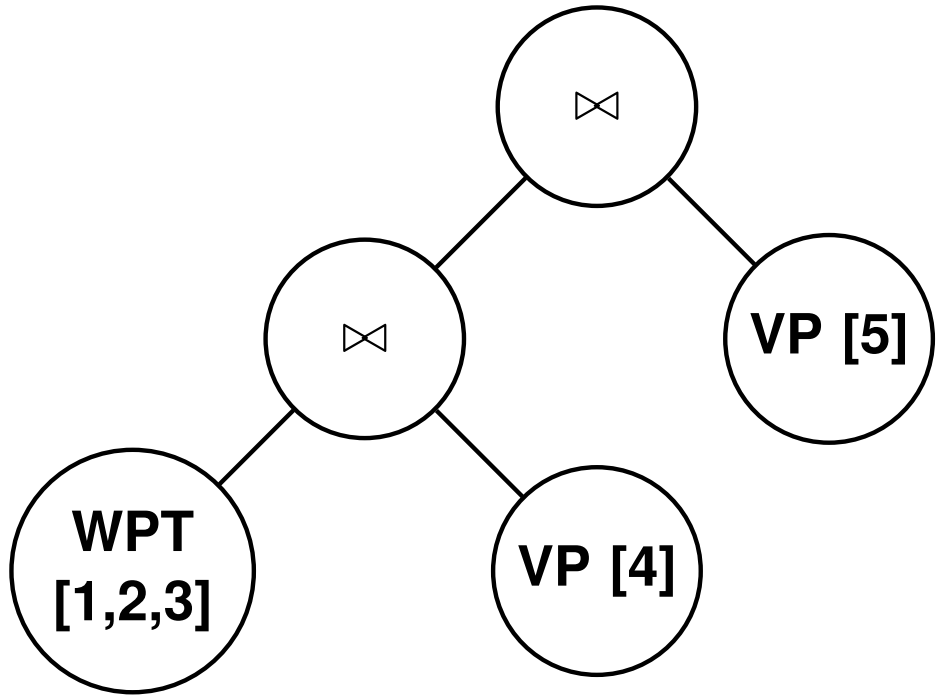

VP_p0
VP_p1
VP_p2

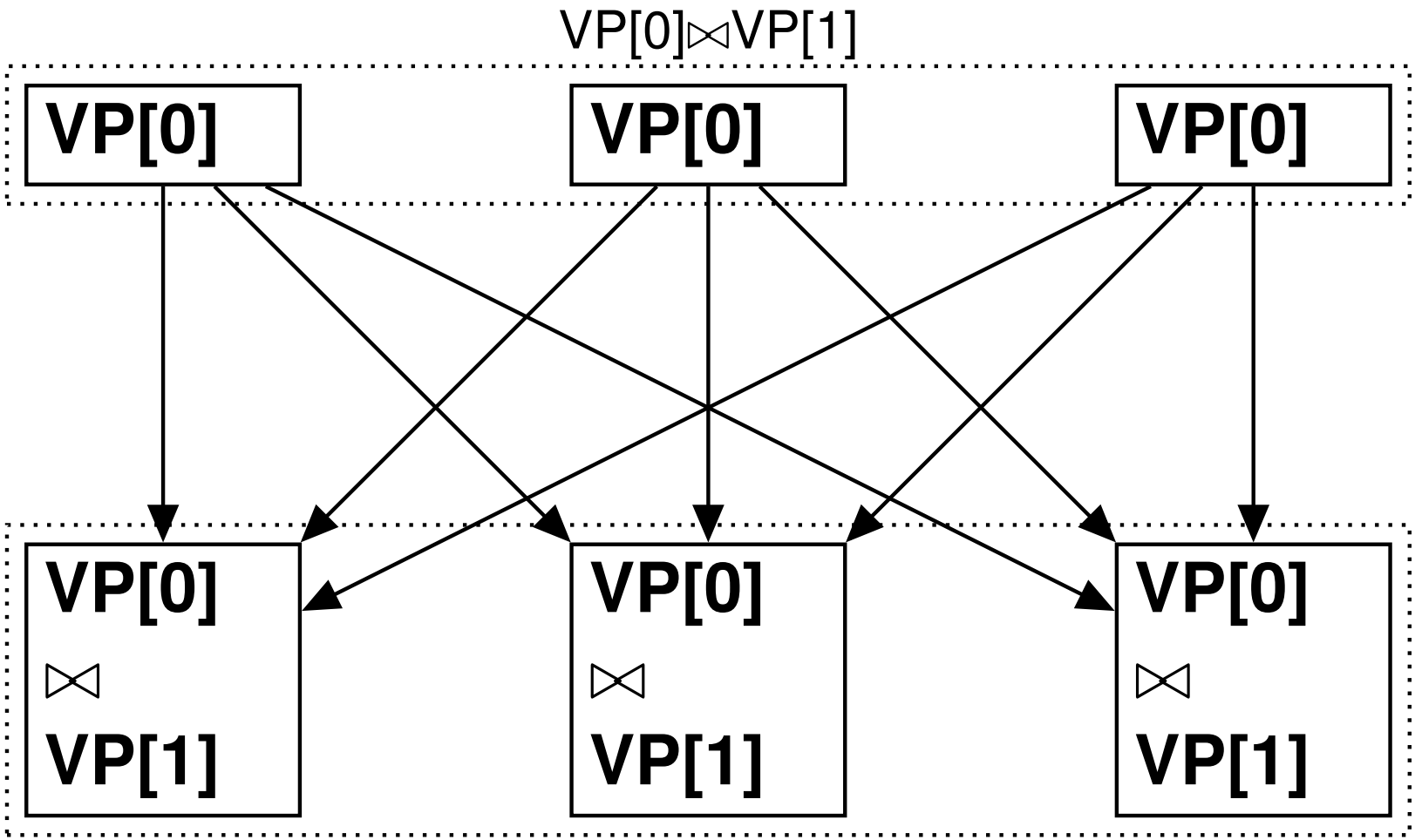
Wide Property Table		
subject	**follows**	**likes**
user1	[user2, user3]	NULL
user2	user1	NULL
user3	user1	[post1, post2]

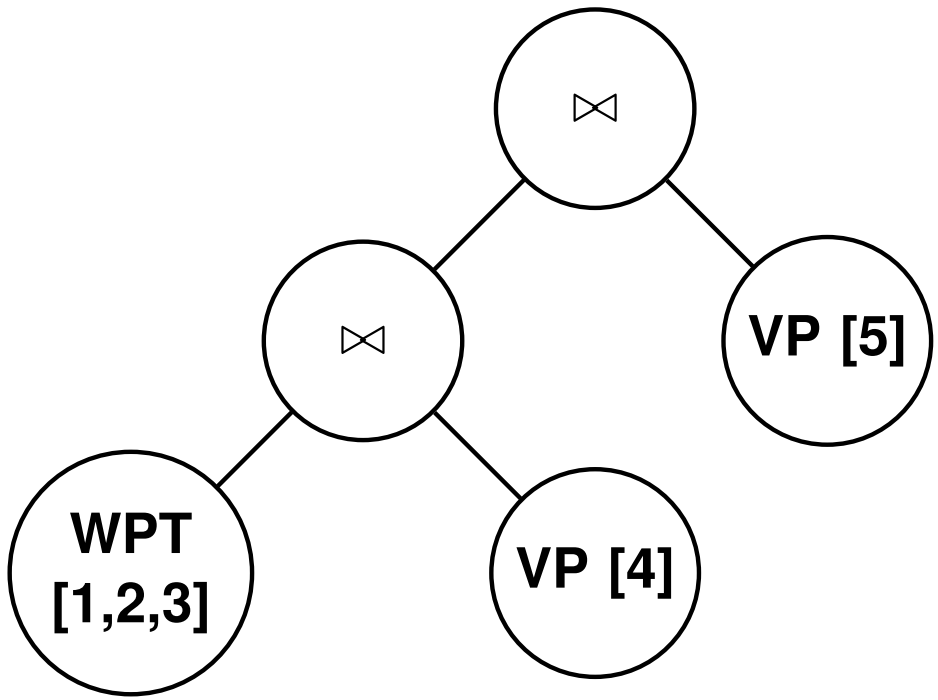


1. ?v0 <p0> <l0> .
2. ?v0 <p1> <l1> .
3. ?v0 <p2> ?v1 .
4. ?v1 <p3> ?v2 .
5. ?v2 <p4> ?v3

1. ?v0 <p0> <l0> .
2. ?v0 <p1> <l1> .
3. ?v0 <p2> ?v1 .
4. ?v1 <p3> ?v2 .
5. ?v2 <p4> ?v3

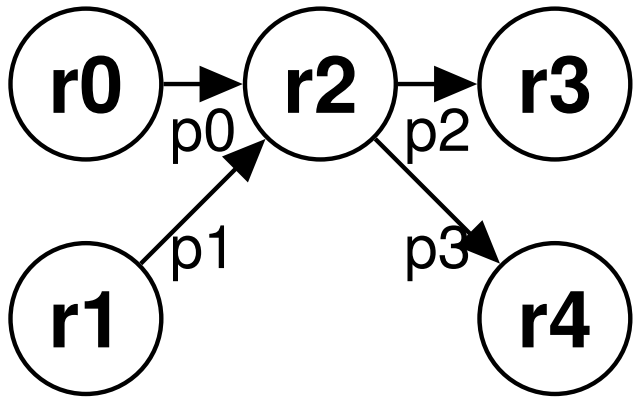



VP[0]⋈VP[1]
VP[0]
VP[0]
VP[0]
VP[0]
⋈
VP[1]
VP[0]
⋈
VP[1]
VP[0]
⋈
VP[1]


⋈
⋈
VP [5]
WPT
[1,2,3]
VP [4]

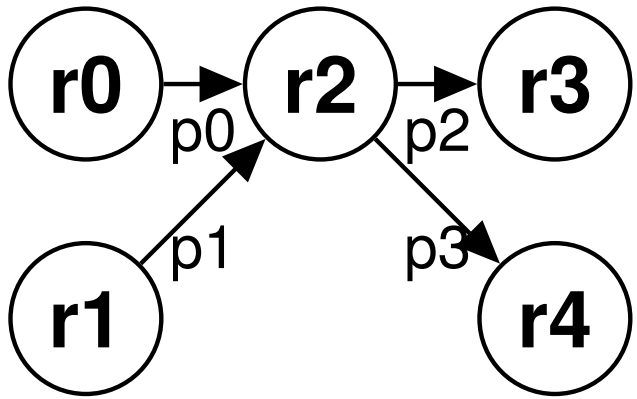
Outline

Wide Property Table and Inverse Wide Property Table



Wide Property Table				
s	**p0**	**p1**	**p2**	**p3**
r0	r2	NULL	NULL	NULL
r1	NULL	r2	NULL	NULL
r2	NULL	NULL	r3	r4

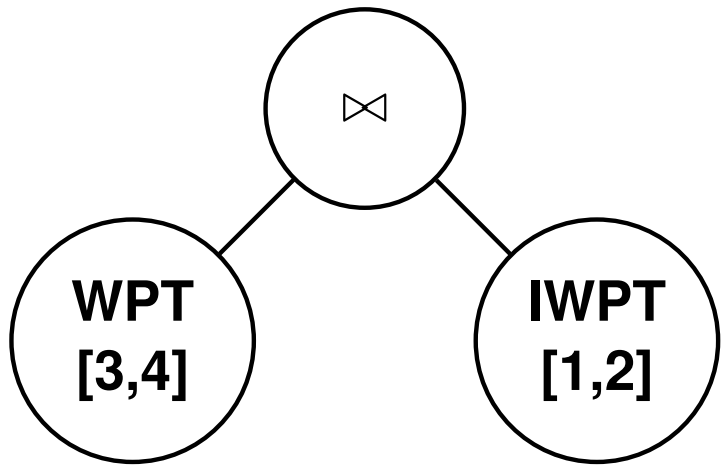
Wide Property Table and Inverse Wide Property Table



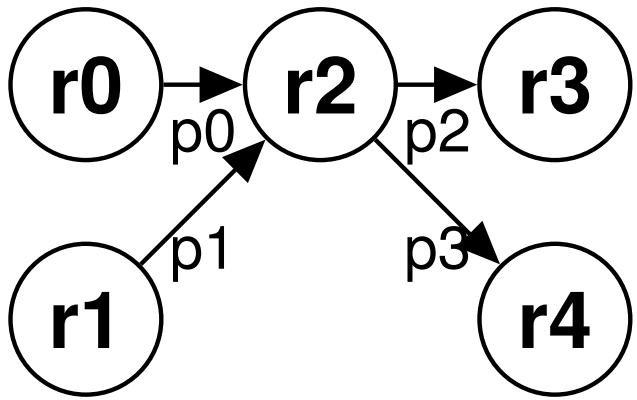
Inverse Wide Property Table				
o	**p0**	**p1**	**p2**	**p3**
r2	r0	r1	NULL	NULL
r3	NULL	NULL	r2	NULL
r4	NULL	NULL	NULL	r2

Wide Property Table and Inverse Wide Property Table

1. ?v0 <p0> **?v2** .
2. ?v1 <p1> **?v2** .
3. **?v2** <p3> ?v3 .
4. **?v2** <p4> ?v4 .



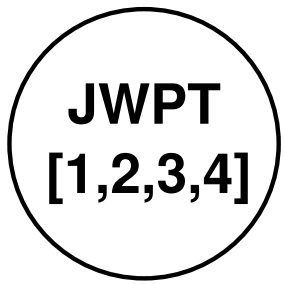
Joined Wide Property Table



Joined Wide Property Table								
p3-i	**p2-i**	**p1-i**	**p0-i**	**r**	**p0**	**p1**	**p2**	**p3**
NULL	NULL	NULL	NULL	r0	r2	NULL	NULL	NULL
NULL	NULL	NULL	NULL	r1	NULL	r2	NULL	NULL
NULL	NULL	r1	r0	r2	NULL	NULL	r3	r4
NULL	r2	NULL	NULL	r3	NULL	NULL	NULL	NULL
r2	NULL	NULL	NULL	r4	NULL	NULL	NULL	NULL

Joined Wide Property Table

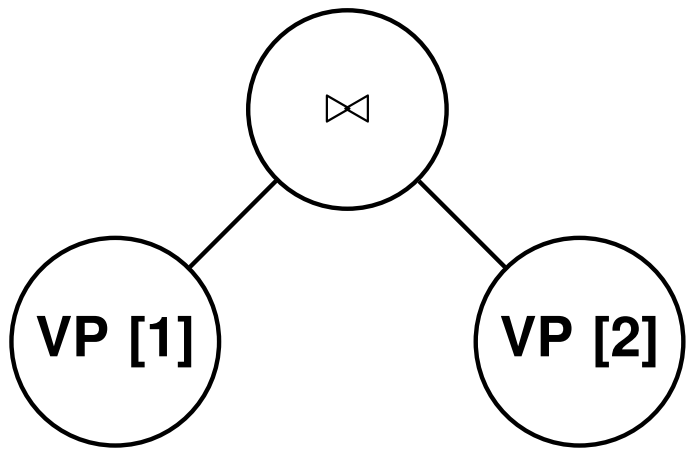
1. ?v0 <p0> **?v2** .
2. ?v1 <p1> **?v2** .
3. **?v2** <p3> ?v3 .
4. **?v2** <p4> ?v4 .



Outline

S2RDF

1. ?v0 <p0> ?v1 .
2. ?v1 <p1> ?v2



S2RDF

vp [1]	
v0	**v1**
a_0	**r**
a_1	b_1
...	...
a_n	b_n

$\bowtie$

vp [2]	
v1	**v2**
r	d_0
c_1	d_1
...	...
c_n	d_n

$\Longrightarrow$

Join Result		
v0	**v1**	**v2**
a_0	r	d_0

S2RDF

vp_p0	
s	**o**
a_0	**r**
a_1	b_1
...	...
a_n	b_n

vp_p1	
s	**o**
r	d_0
c_1	d_1
...	...
c_n	d_n

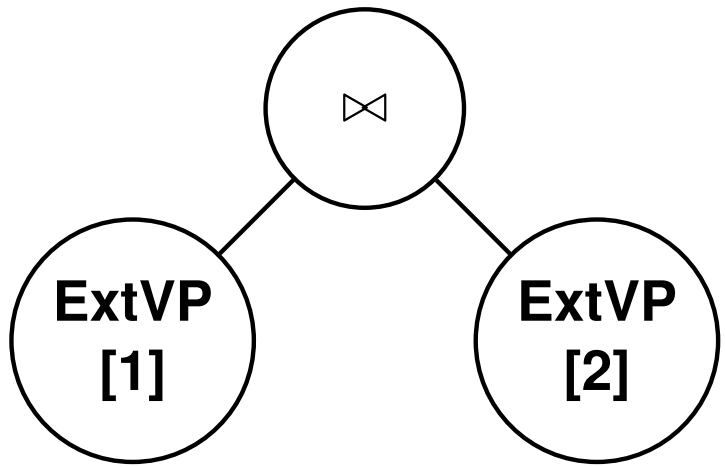
$\Longrightarrow$

extvp_OS_p0|p1	
s	**o**
a_0	r

extvp_SO_p1|p0	
s	**o**
r	d_0

S2RDF

1. ?v0 <p0> ?v1 .
2. ?v1 <p1> ?v2



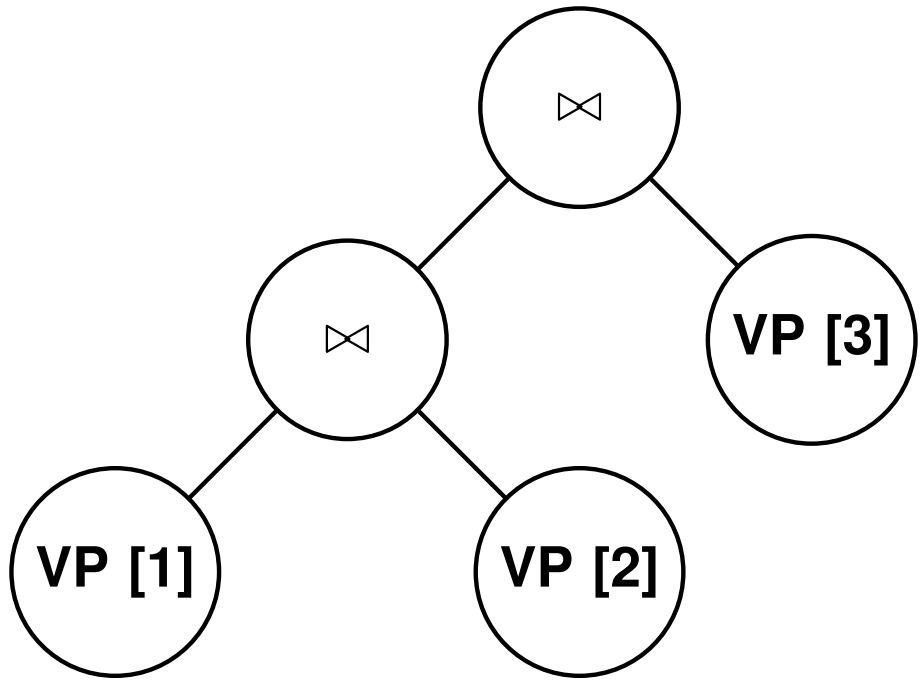
Loading time

100 million triples:

- **PRoST**: less than 1h
- **S2RDF**: more than 16h

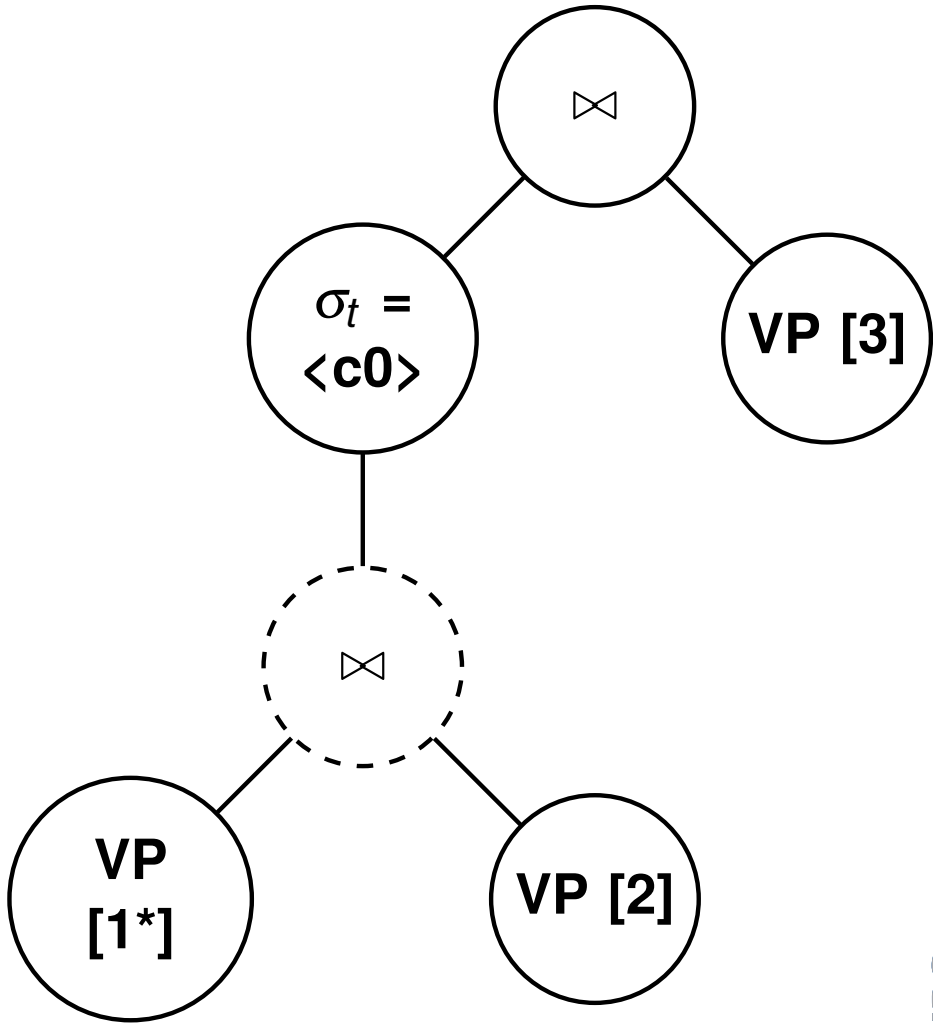
Dynamic ExtVP

1. <c0> <p0> ?v1 .
2. ?v1 <p1> ?v2 .
3. ?v2 <p2> ?v3 .



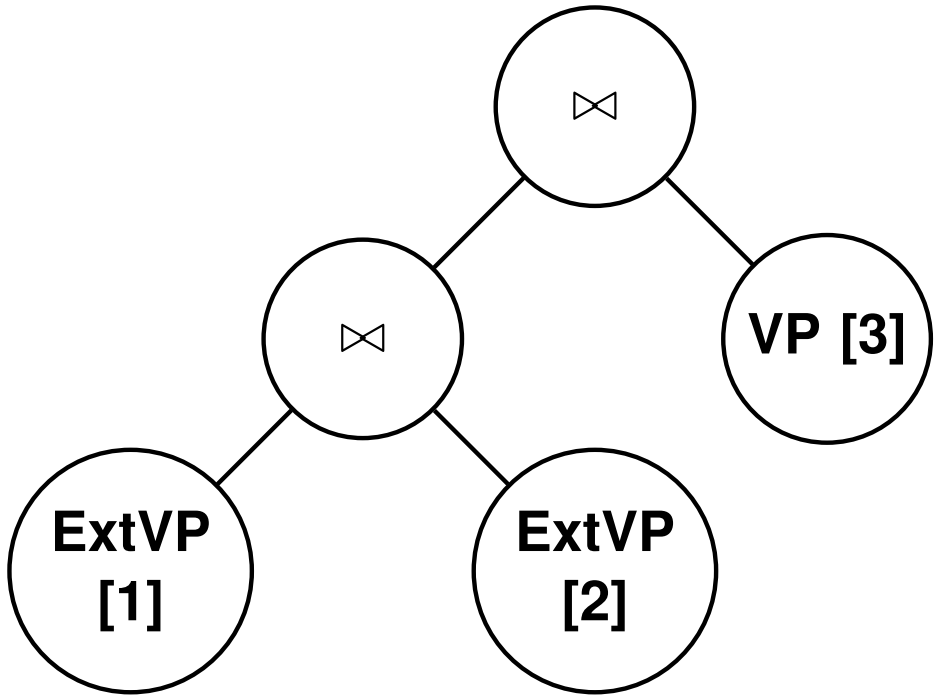
Dynamic ExtVP

1. <c0> <p0> ?v1 .
2. ?v1 <p1> ?v2 .
3. ?v2 <p2> ?v3 .



Dynamic ExtVP

1. <c1> <p0> ?v1 .
2. ?v1 <p1> ?v2 .
3. ?v2 <p2> ?v3 .



Outline

Cluster

10 machines:

- 6 Core Intel Xeon E5-2420
- 32GB RAM
- 4TB HD
- Spark 2.2.0

Dataset

WatDiv synthetic dataset:

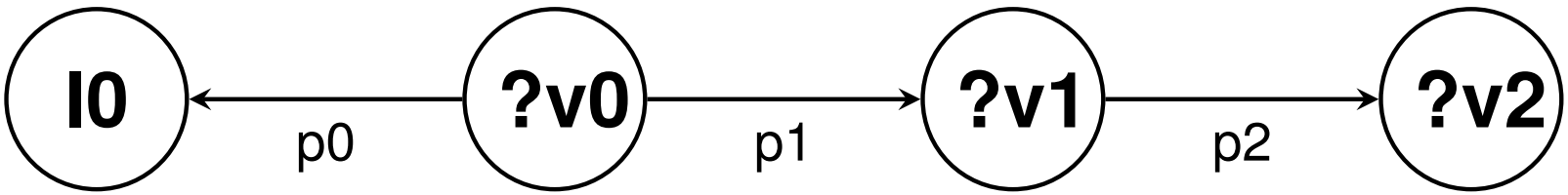
- 100 million triples
- 5 million subjects
- 80 distinct predicates
- 9 million objects

Queryset

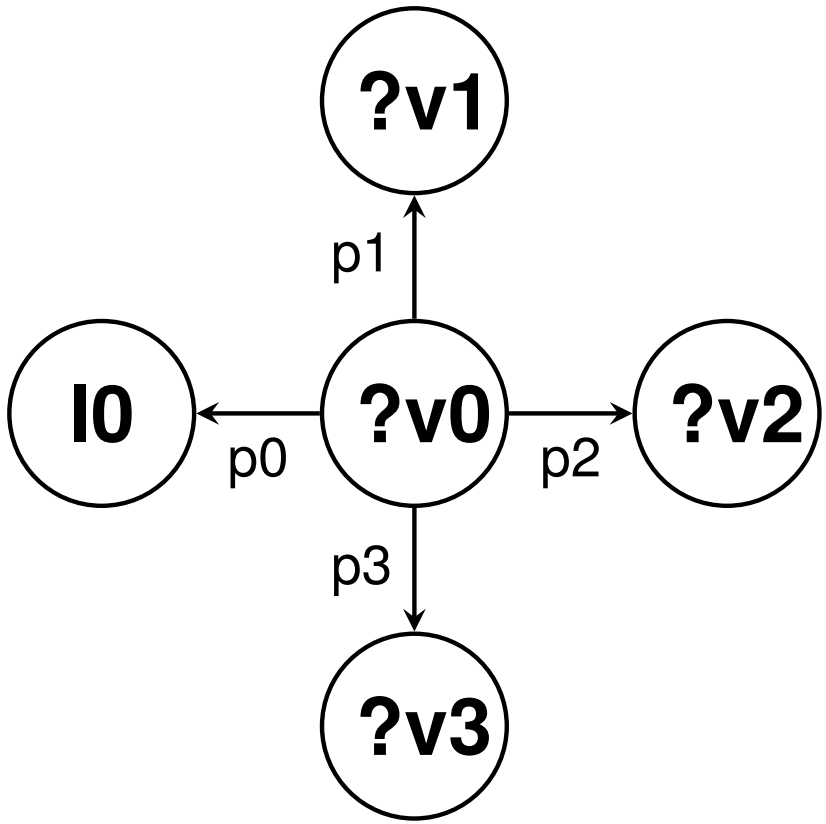
88 queries:

- 25 linear-shaped queries
- 35 star-shaped queries
- 25 snow-shaped queries
- 3 complex-shaped queries

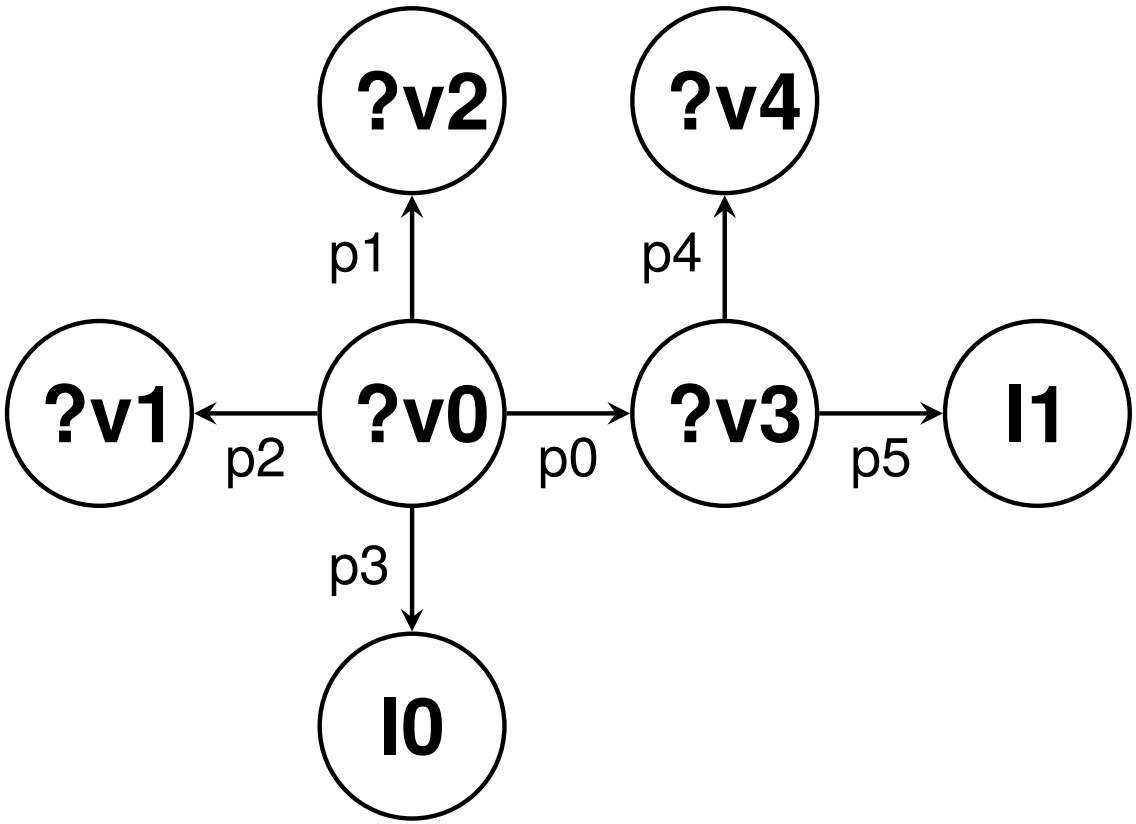
Linear-shaped query



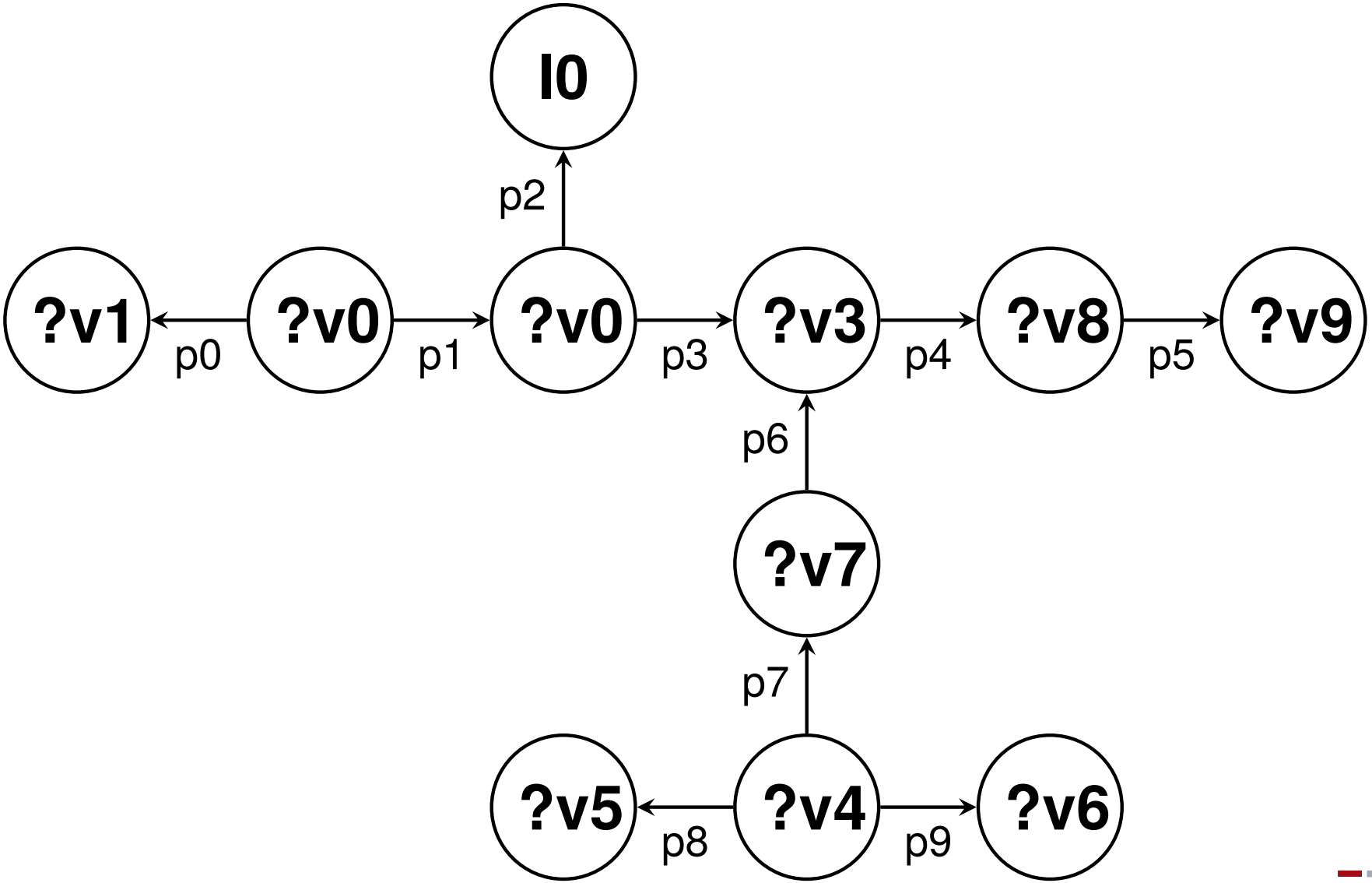
Star-shaped query



Snow-shaped query



Complex-shaped query

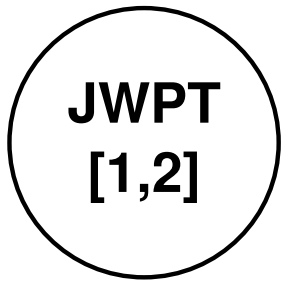


JWPT Evaluation

- VP + (I)WPT **(baseline)**
- VP + JWPT
- VP + (I/J)WPT

VP + JWPT

1. ?v0 <p0> ?v1 .
2. ?v0 <p1> ?v2 .



Joined Wide Property Table								
p3-i	**p2-i**	**p1-i**	**p0-i**	**r**	**p0**	**p1**	**p2**	**p3**
NULL	NULL	NULL	NULL	r0	r2	NULL	NULL	NULL
NULL	NULL	NULL	NULL	r1	NULL	r2	NULL	NULL
NULL	NULL	r1	r0	r2	NULL	NULL	r3	r4
NULL	r2	NULL	NULL	r3	NULL	NULL	NULL	NULL
r2	NULL	NULL	NULL	r4	NULL	NULL	NULL	NULL

VP + (I/J)JWPT

1. ?v0 <p0> ?v1 .
2. ?v0 <p1> ?v2 .

Wide Property Table				
s	**p0**	**p1**	**p2**	**p3**
r0	r2	NULL	NULL	NULL
r1	NULL	r2	NULL	NULL
r2	NULL	NULL	r3	r4

VP + JWPT

Less join operations than the baseline:

- linear-shaped: 1.5s faster
- star-shaped: 200ms - 1.7s faster
- snow-shaped: 5s - 8s faster

Same number of join operations as the baseline:

- 200ms -1.1s slower

VP + (I/J)WPT

Less join operations than the baseline:

- linear-shaped: 1.5s faster
- star-shaped: 200ms - 1.7s faster
- snow-shaped: 5s - 8s faster

Same number of join operations as the baseline:

- no statistically significant difference

Discussion

VP + JWPT:

- Less join operations => faster processing
- More tuples => slower processing

VP + (I/J)WPT:

- At least as fast as the baseline

Dynamic ExtVP Evaluation

- VP **(baseline)**
- Dynamic ExtVP
- Dynamic ExtVP (table creation only)

Dynamic ExtVP Evaluation

Dynamic ExtVP:

- 2s - 15s faster

Dynamic ExtVP (table creation only):

- Two ExtVP tables created: no statistically significant difference
- More tables created: 1.5s to 5.9s slower

Discussion

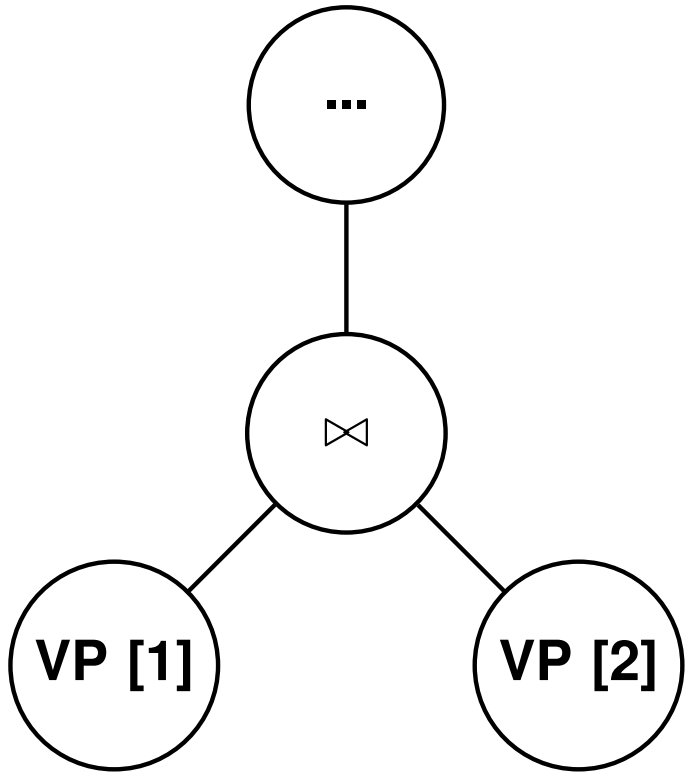
- On average, faster processing of queries
- Slower processing, when many ExtVP tables are created in a single execution
- Tables only need to be created once
- Faster loading time than S2RDF
- Slower processing than PRoST with VP +(I)WPT

Dynamic ExtVP Evaluation

- VP + (I)WPT **(baseline)**
- Dynamic ExtVP + (I)WPT
- Dynamic ExtVP (table creation only) + (I)WPT

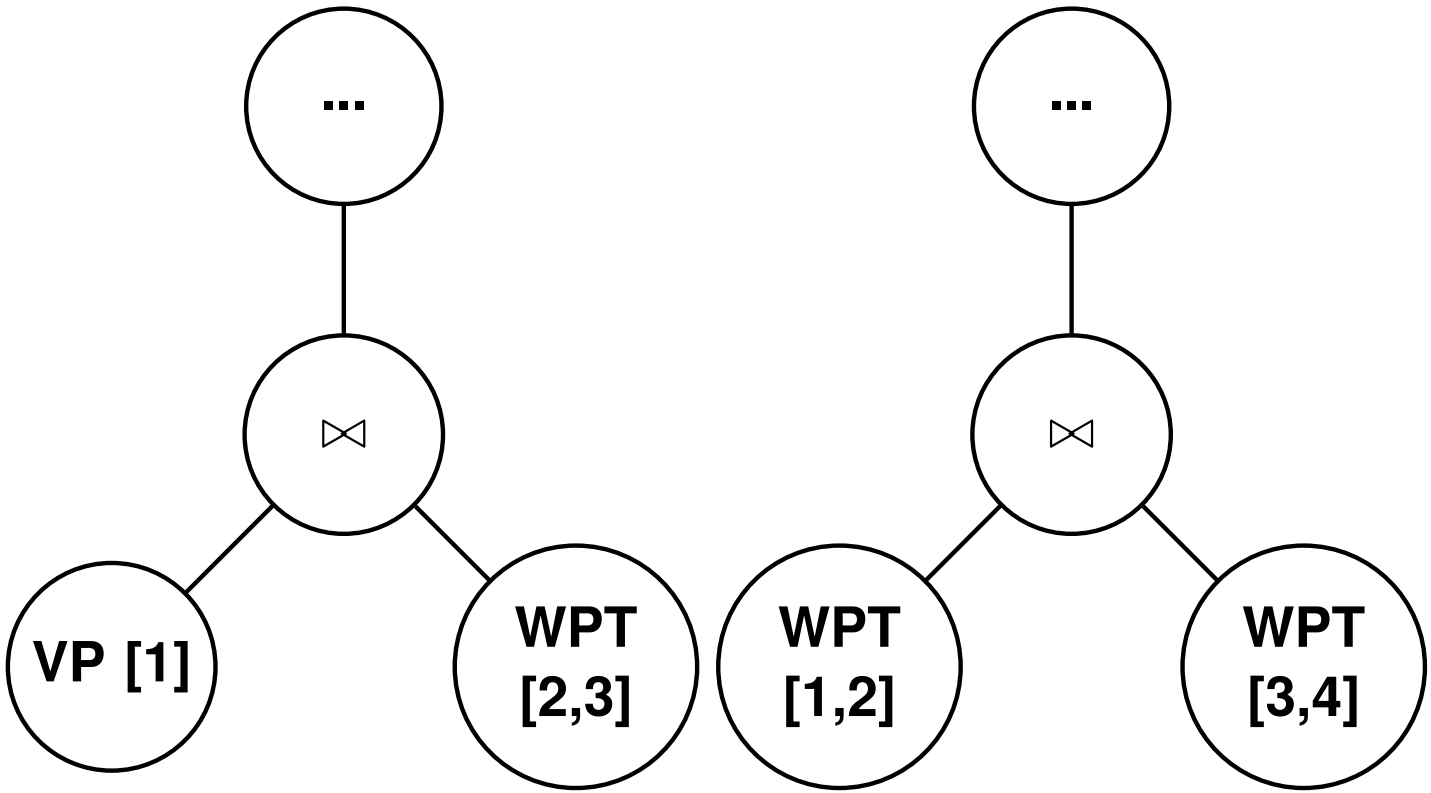
No statistically significant execution time difference.

Why?



=> Two ExtVP tables created

Why?



Outline

Dynamic ExtVP

- Reduced tables are faster
- No extra loading cost
- Creation time of tables is not very high

Future: persist reduced tables from join operations between arbitrary node types

JWPT

- Faster when it reduces the number of required join operations
- Not always efficient by itself

Future: reduce the number of tuples in the JWPT

JWPT

Joined Wide Property Table								
p3-i	**p2-i**	**p1-i**	**p0-i**	**r**	**p0**	**p1**	**p2**	**p3**
NULL	NULL	NULL	NULL	r0	r2	NULL	NULL	NULL
NULL	NULL	NULL	NULL	r1	NULL	r2	NULL	NULL
NULL	NULL	r1	r0	r2	NULL	NULL	r3	r4
NULL	r2	NULL	NULL	r3	NULL	NULL	NULL	NULL
r2	NULL	NULL	NULL	r4	NULL	NULL	NULL	NULL

Joined Wide Property Table								
p3-i	**p2-i**	**p1-i**	**p0-i**	**r**	**p0**	**p1**	**p2**	**p3**
NULL	NULL	r1	r0	r2	NULL	NULL	r3	r4

Outline

Acknowledgments

Thank you.

Guilherme Schievelbein
schieveg@tf.uni-freiburg.de