WORQ: Workload-Driven RDF Query Processing

Amgad Madkour (Purdue)
Ahmed Aly (Google)
Walid G. Aref (Purdue)

Department of Computer Science
Purdue University

Introduction
RDF Data Is Everywhere

- RDF is an **integral** component in many systems:
  - Semantic Search, Smart Governments (Data.gov), Medical Systems
- (**Linked**) RDF data contains very rich relations:
  - Data.gov – 5 billion triples
  - Linked Cancer Genome Atlas – 7.36 billion triples
  - US Census Data – 1 billion triples
- **Cloud-based systems** are ideal for RDF data management (e.g., Storage, Query Processing)

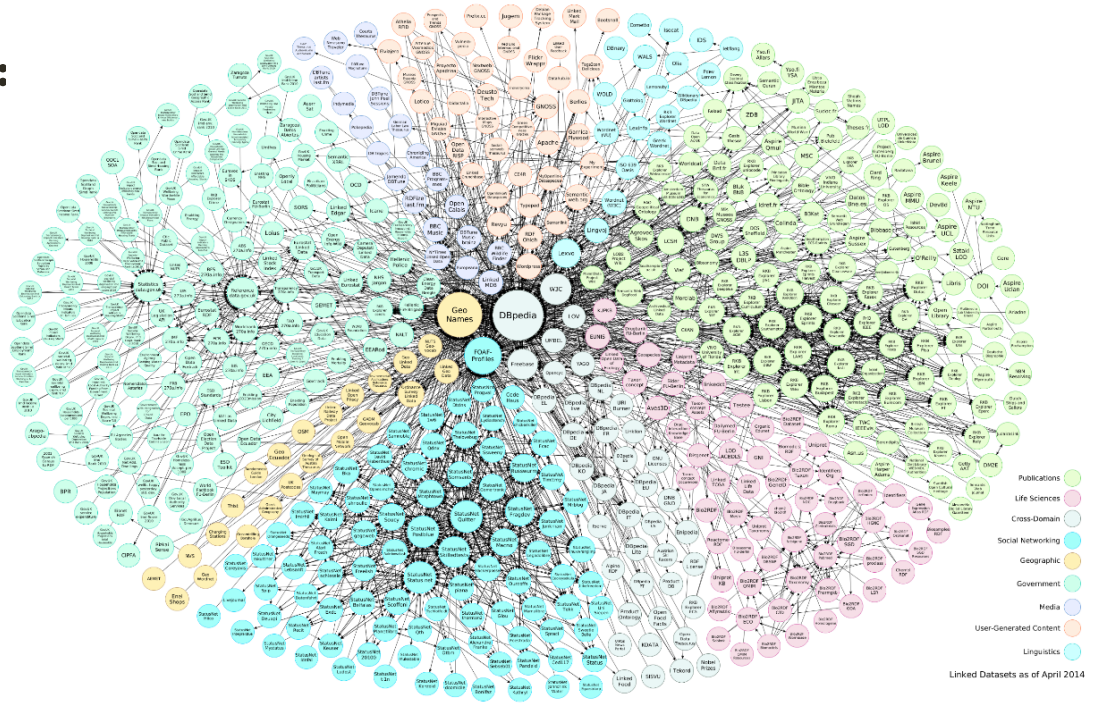
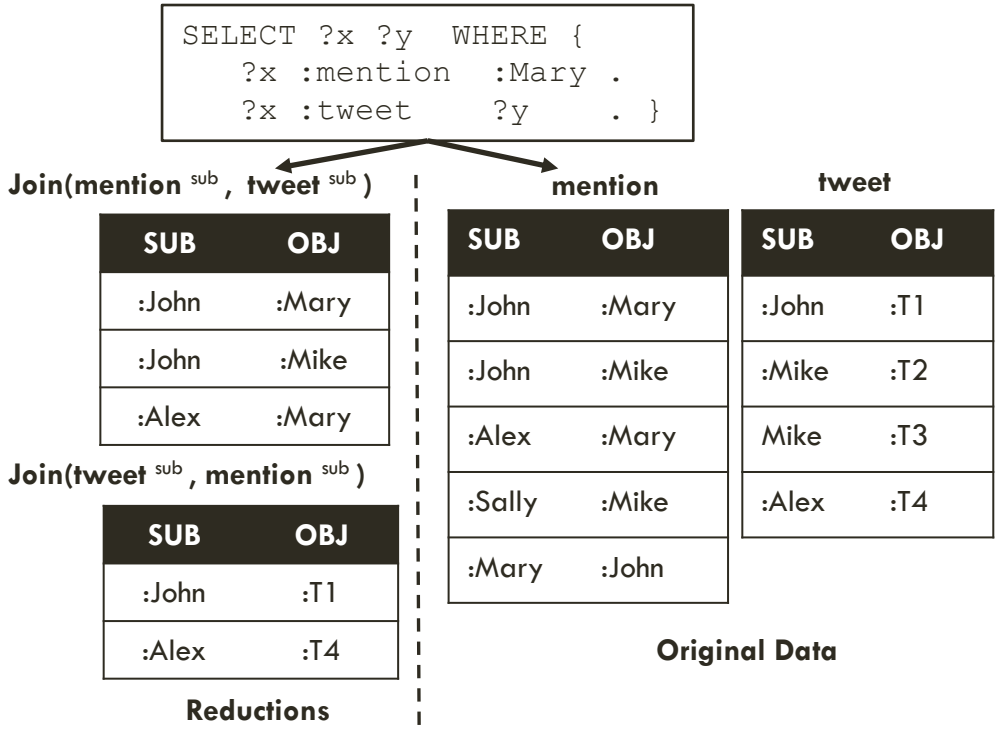


Figure: Linked RDF Data Cloud containing thousands of datasets

Introduction
Processing RDF Queries

- Network **shuffling** overhead **degrades** query **performance** in a distributed environment
- **Intermediate results** represent the data that satisfies the binary join and contributes to the final result of the query
- Reducing the network shuffling relies on how the data is **partitioned** across the nodes and the intermediate results **size**



Problem Statement

- **Data partitioning** incurs a preprocessing overhead as it needs to be performed over the whole data
- **Intermediate results** may contain redundant data triples that do not match all the query joins
- **Caching** the unique query results incurs significant memory storage overhead

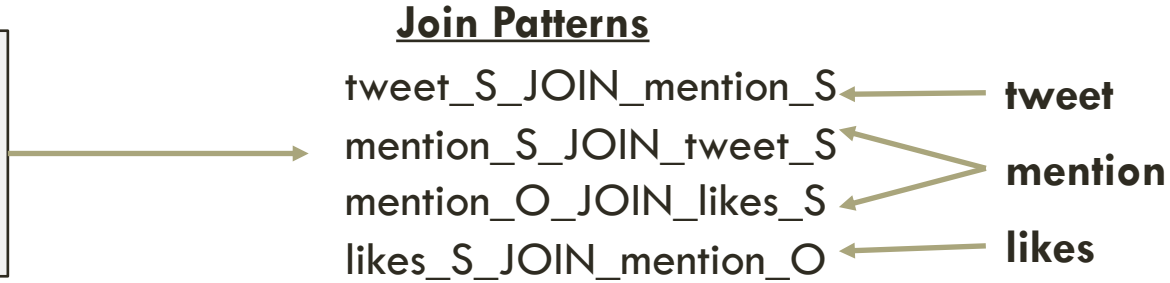
Proposal

- We present **online** method for **computing reductions** of RDF data using **Bloom filters**
- We present **workload-driven partitioning** of RDF triples that can join together in order to minimize the network shuffling overhead
- We show that **caching** the RDF join **reductions** can boost the **query performance** while keeping the cache size minimal
- We study an efficient technique for answering RDF queries with **unbound properties** using Bloom filters

Online Reduction of RDF Data
Join Patterns

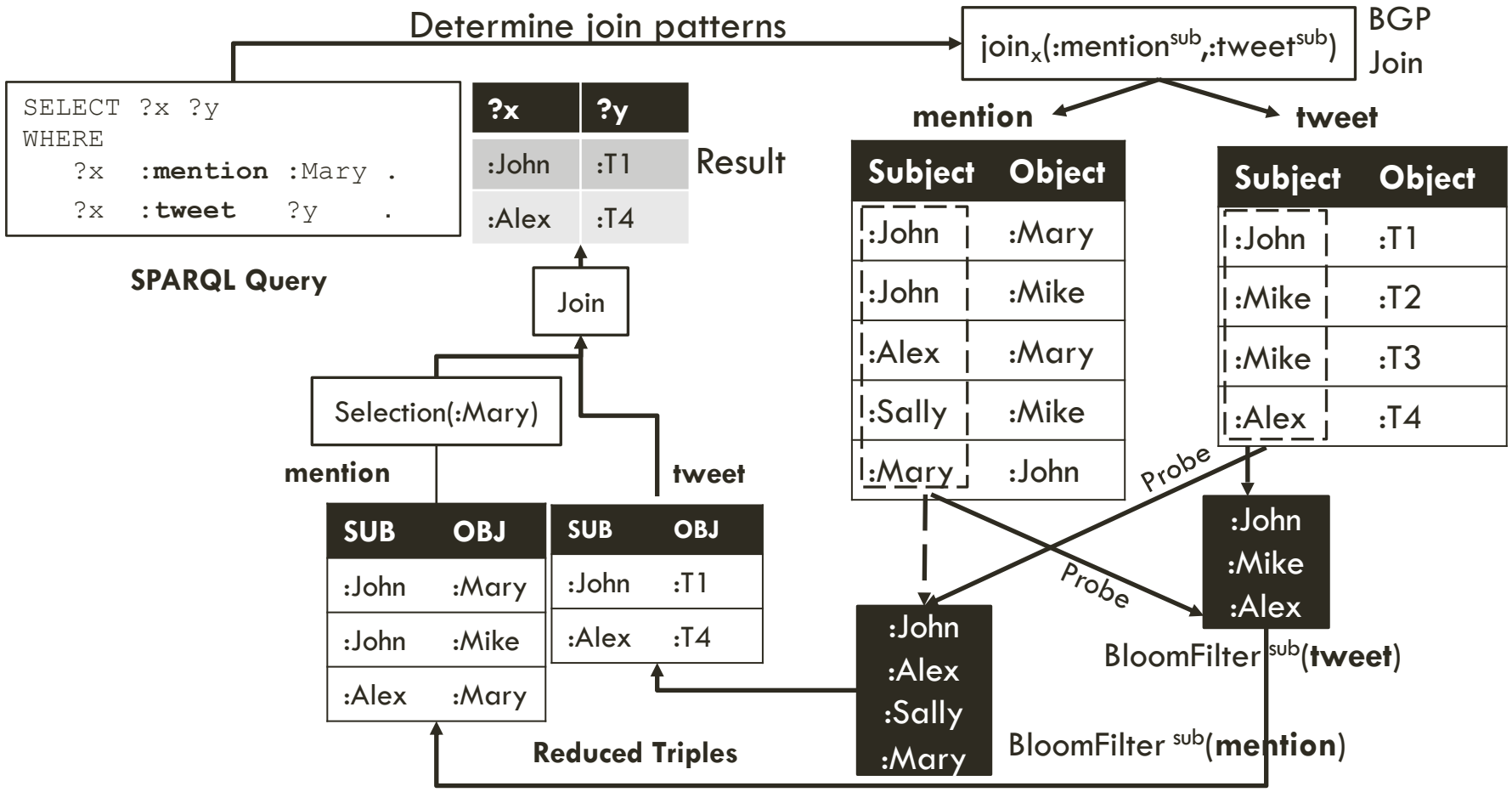
- SPARQL queries consist of **Basic Graph Patterns (BGP)**
- Every BGP consists of a set of **triples**
- **Join patterns** represent **correlations** between triples in a SPARQL Basic Graph Pattern (BGP)

```
SELECT ?x ?y ?w
WHERE
?x        :tweet        :T1
?x        :mention      ?y
?y        :likes        ?w
```



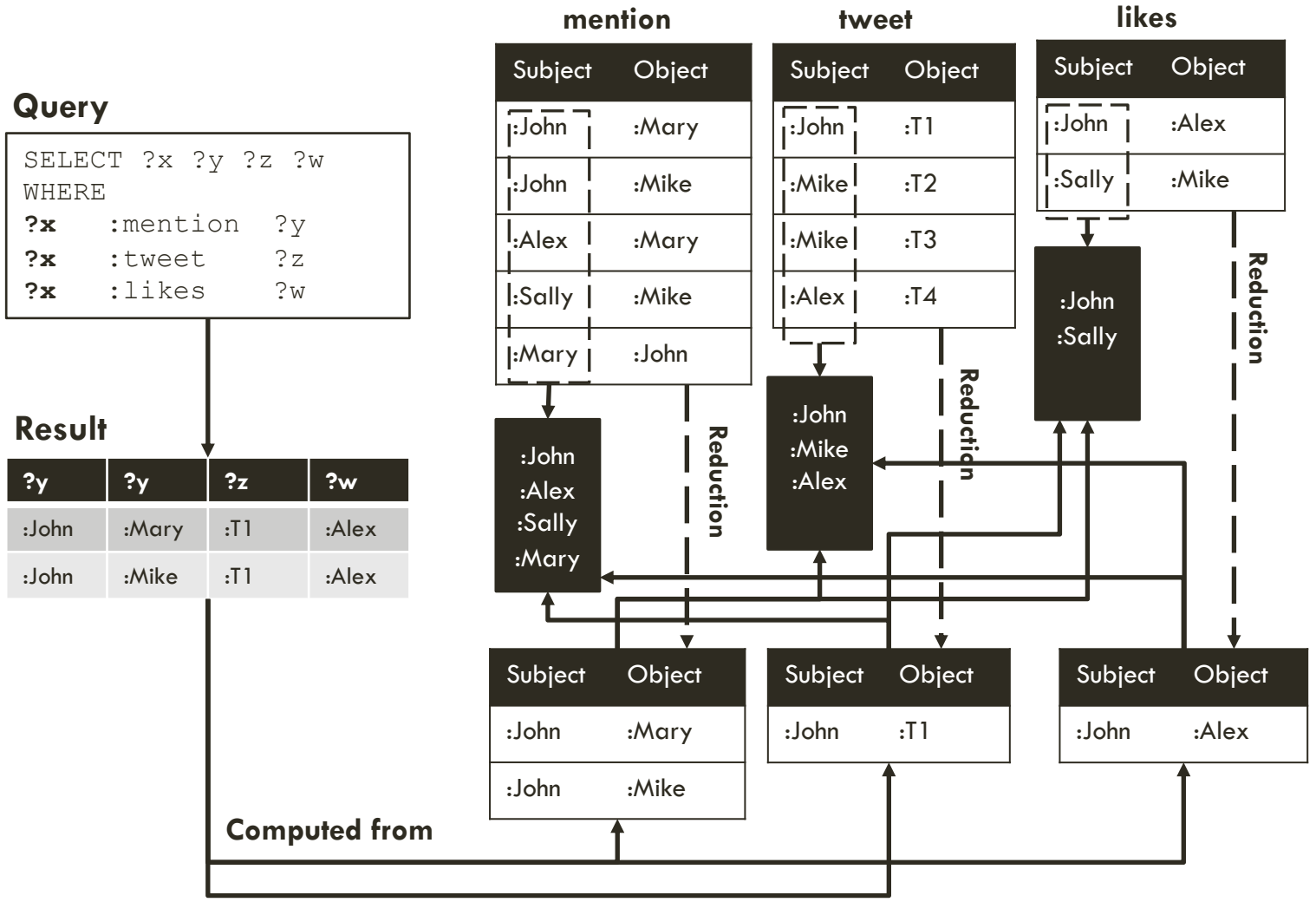
Online Reduction of RDF Data

Bloom Join

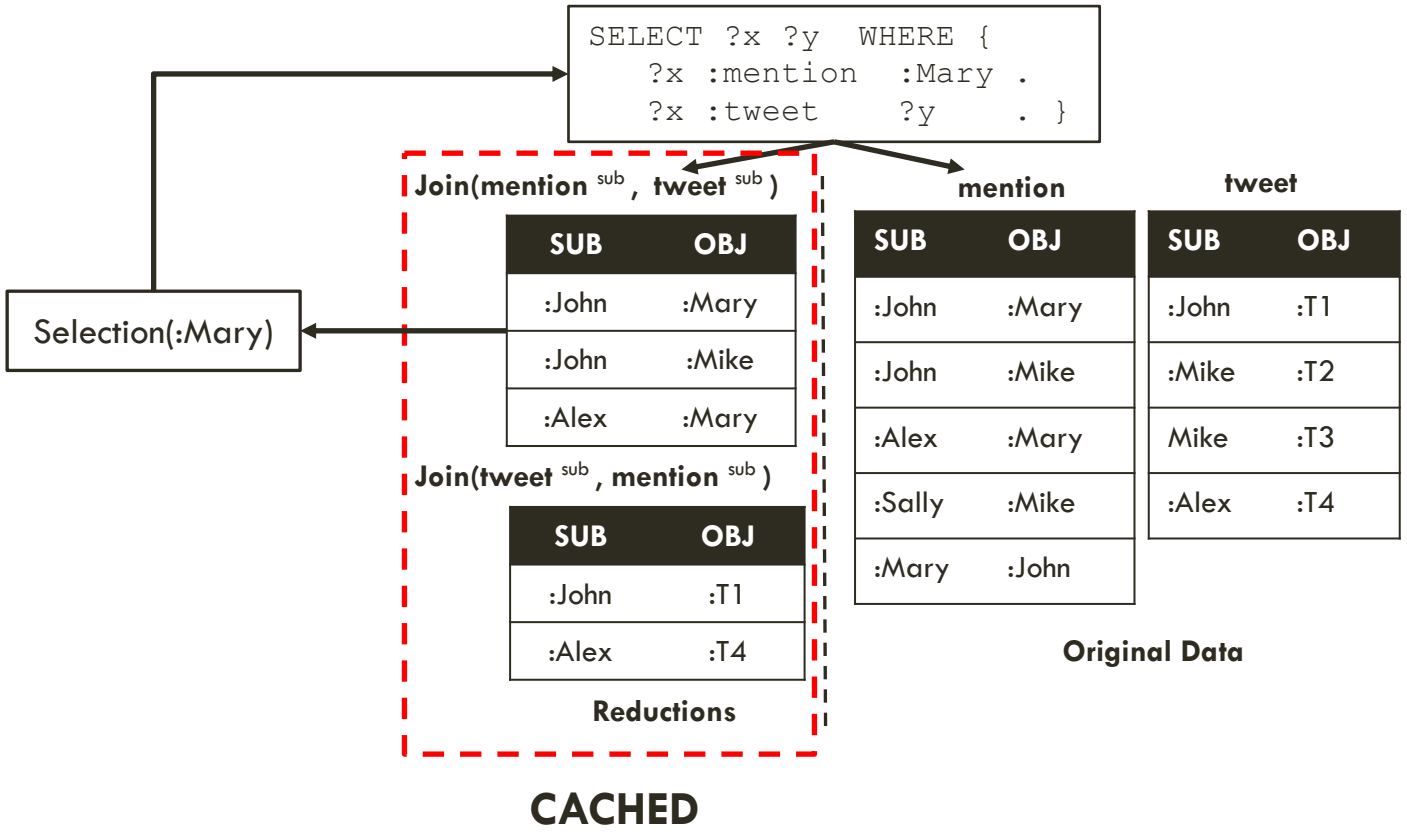


Online Reduction of RDF Data

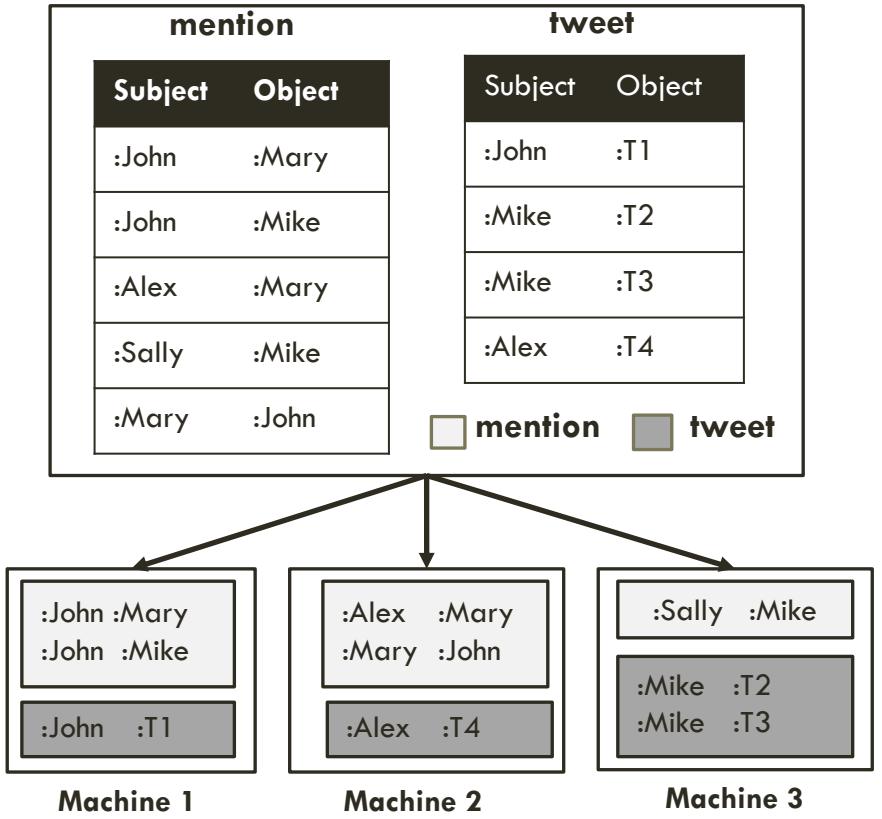
N-ary Join



Online Reduction of RDF Data
Caching



Workload-Driven Partitioning
Overview



Workload-Driven Partitioning
Proposal

```
SELECT ?x ?y ?w
WHERE
?x          :tweet          :T1
?x          :mention        ?y
?y          :likes          ?w
```

Possible Reductions

Reductions	Reduction ID
Reduction(tweet^{sub}, mention^{sub})	R1
Reduction(mention^{sub}, tweet^{sub})	R2
Reduction(likes^{sub}, mention^{obj})	R3

Reductions

Reduction 1 (R1)

Subject	Object
:John	:T1
:Alex	:T4

Reduction 2 (R2)

Subject	Object
:John	:Mary
:John	:Mike
:Alex	:Mary

Reduction 3 (R3)

Subject	Object
:John	:Alex

Partitioning

Machine 1

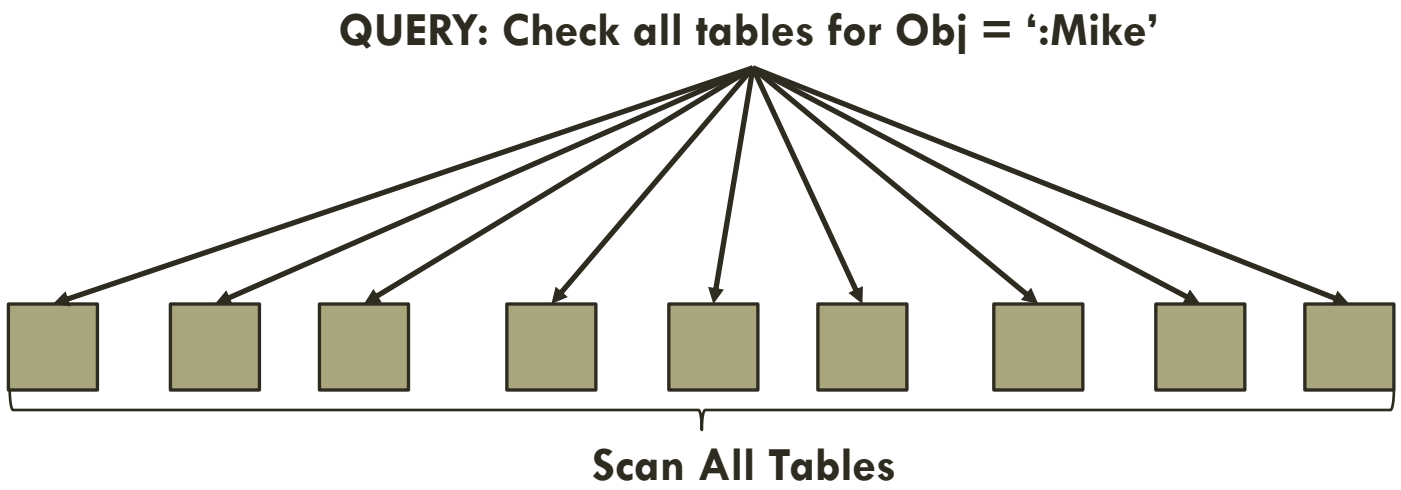
:John	:T1	R1
:John	:Alex	R3
:John	:Mary	R2
:John	:Mike	

Machine 2

:Alex	:T4	R1
:Alex	:Mary	R2

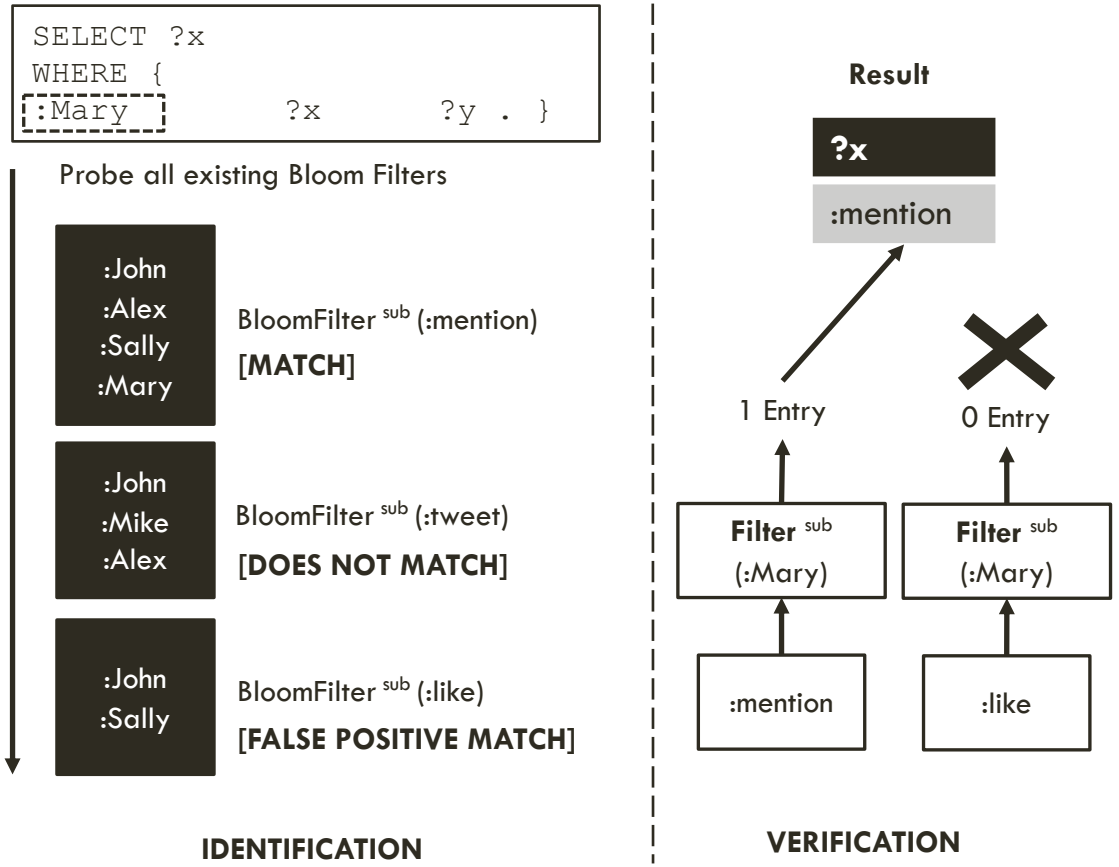
Queries with Unbound Properties
Overview

```
SELECT ?x ?z
WHERE
?x        ?z        :Mike
```



Queries with Unbound Properties

Proposal



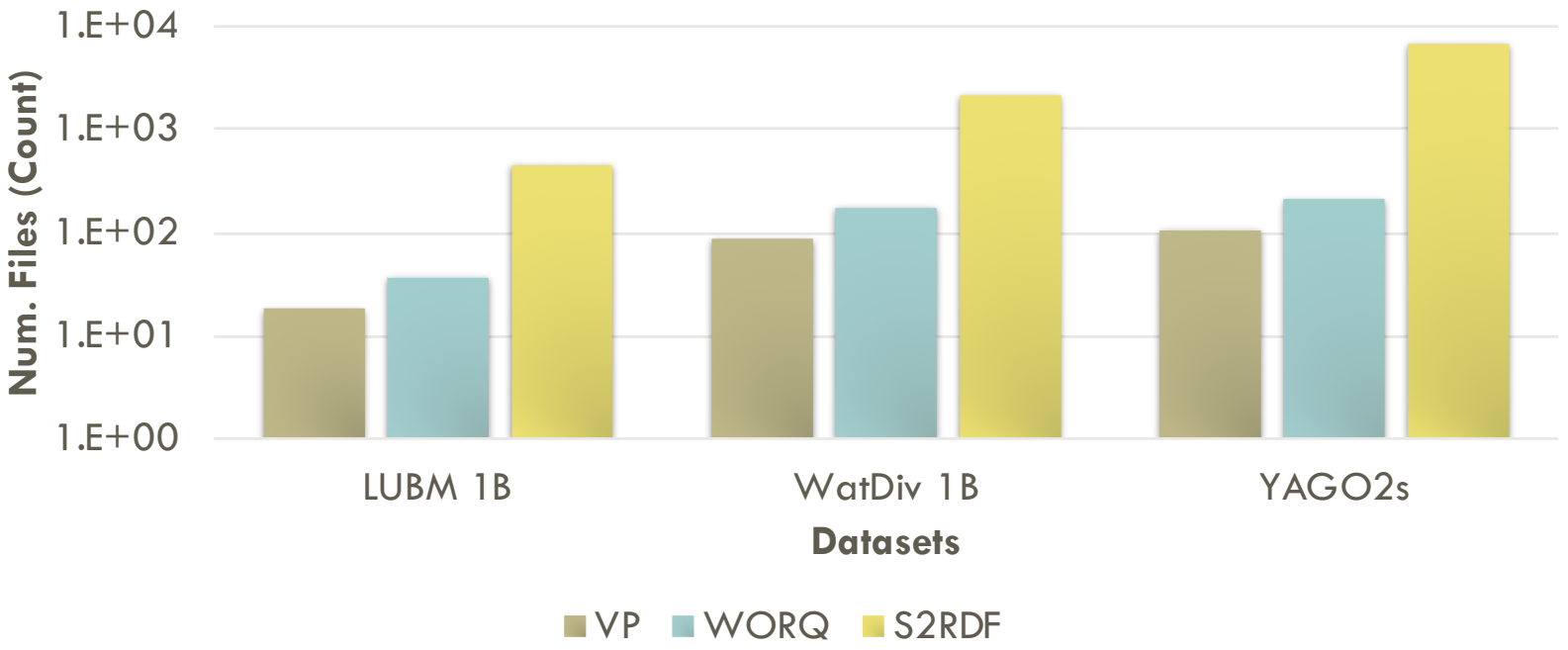
Experimental Setup

- **Systems**
  - WORQ: Implemented inside Knowledge Cubes (KC)
  - S2RDF: State of the art Spark-based RDF engine
- **Benchmarks**
  - **WatDiv**
    - **Dataset:** 1 Billion Triple, **Query Workload:** 5K queries
      - **Patterns:** Covers 100 diverse SPARQL patterns, each containing 50 variations
    - **Unbound Property Queries:** 500 queries
  - **LUBM**
    - **Dataset:** 1 Billion Triple, **Query Workload:** 1K queries
      - **Patterns:** Covers 20 diverse SPARQL patterns
  - **YAGO**
    - Dataset: 245 million triples

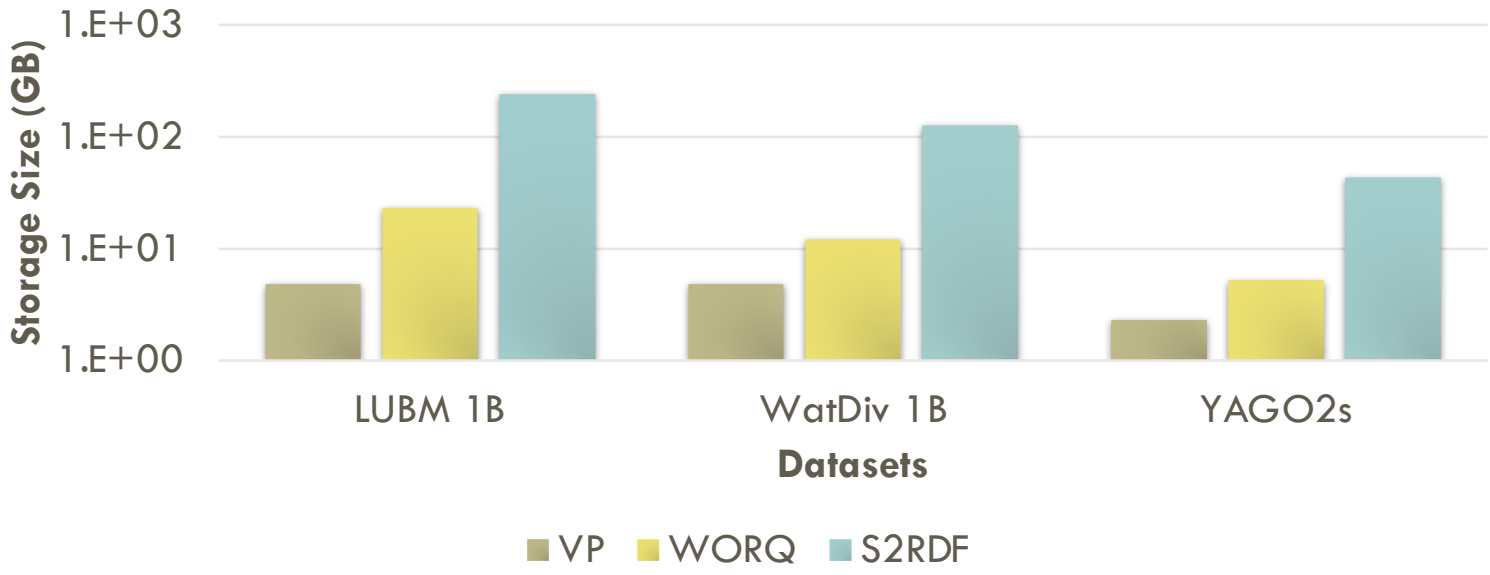
KNOWLEDGE CUBES

GitHub Homepage
https://github.com/amgadmadkour/knowledgecubes

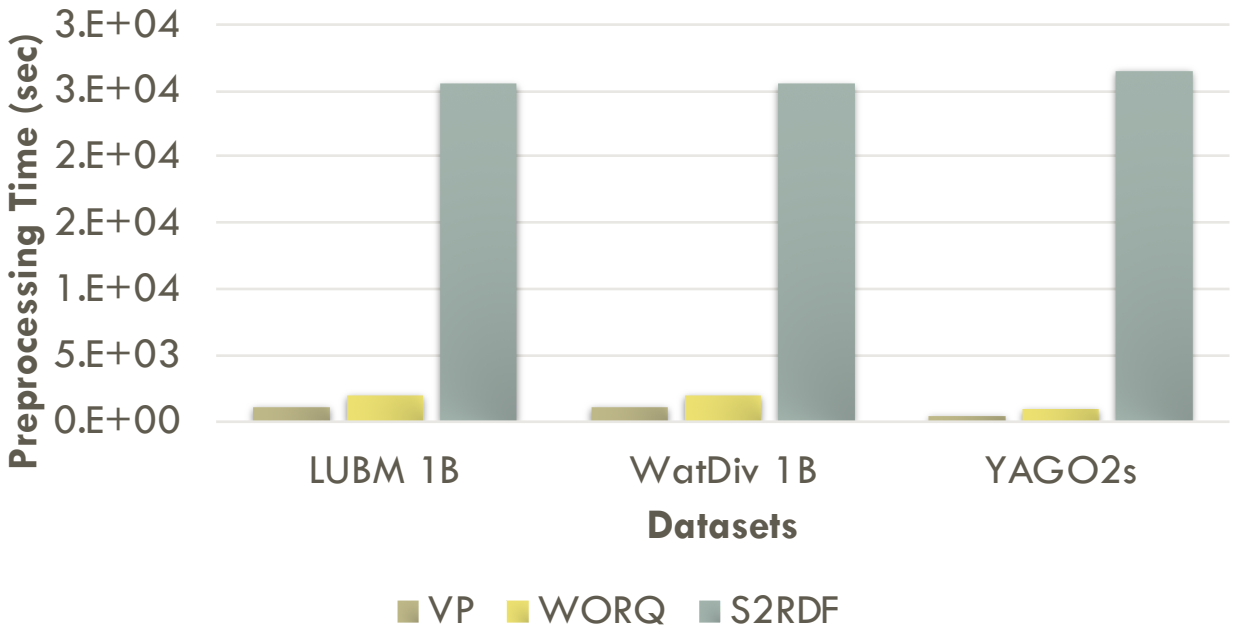
Number of Files



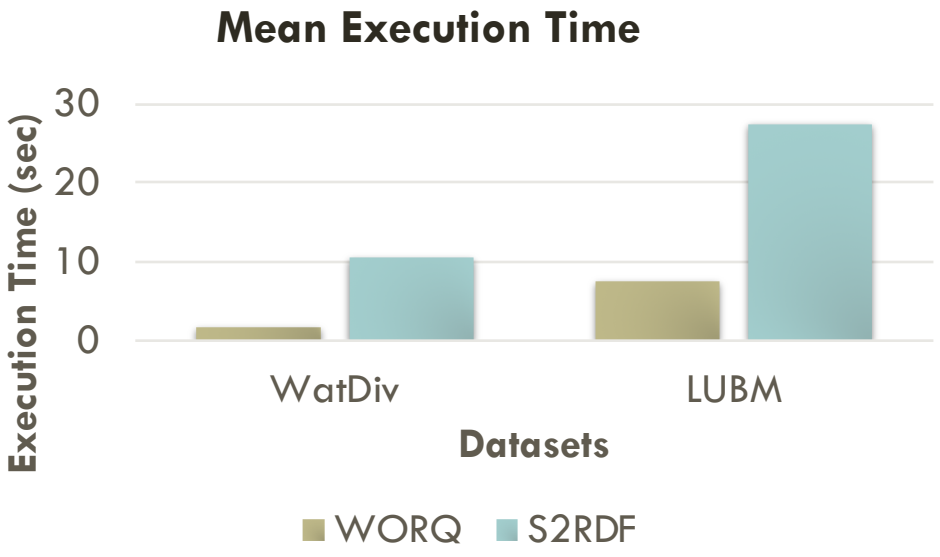
Data Size on HDFS

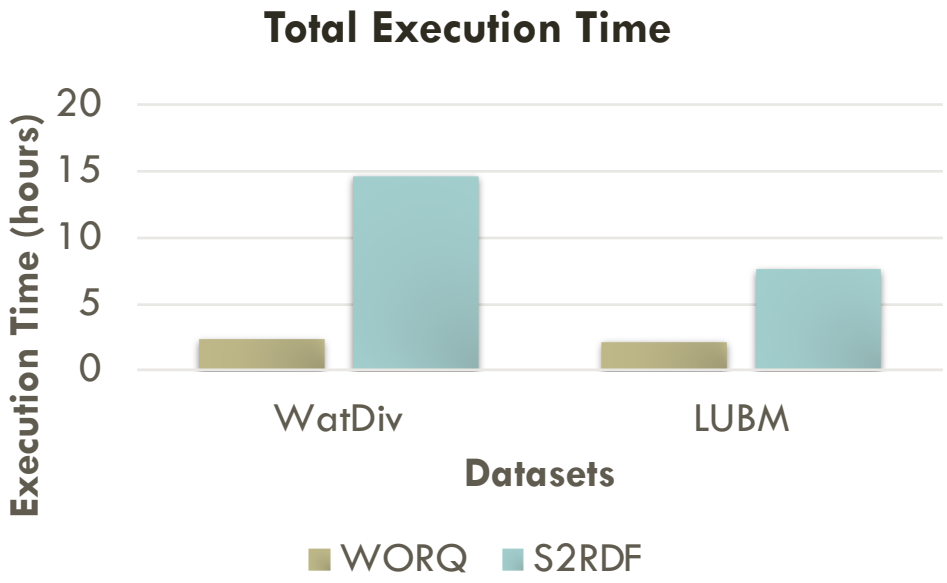


Preprocessing Time



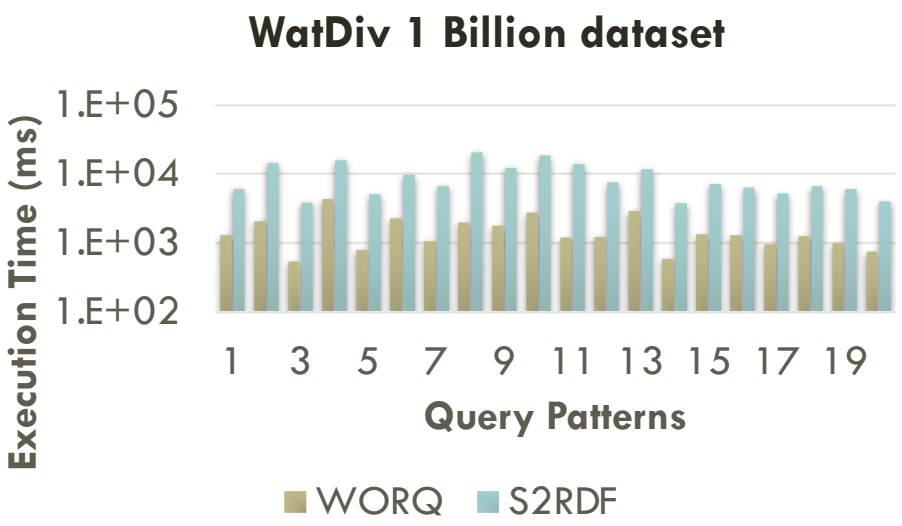
Query Execution Performance
Workload Generators

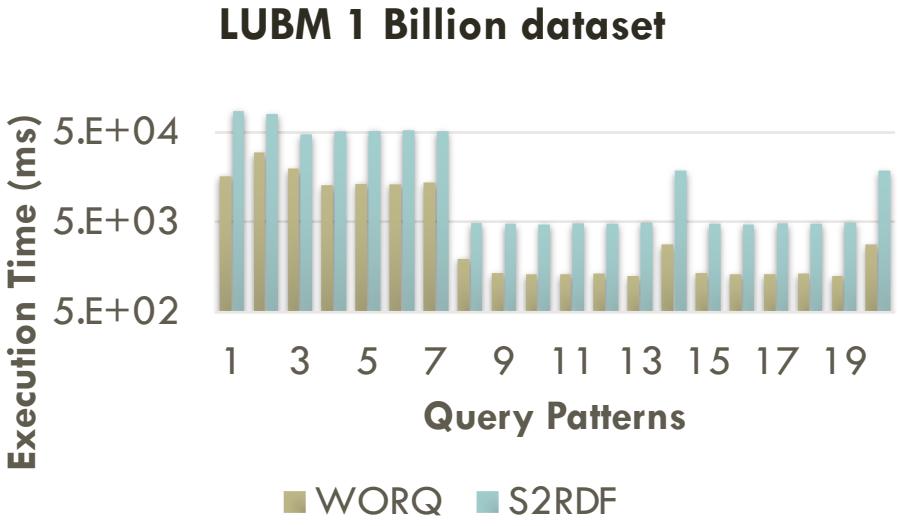




5000 queries over **WatDiv** (1 Billion triples) and
1000 queries over **LUBM** (1 Billion triples)

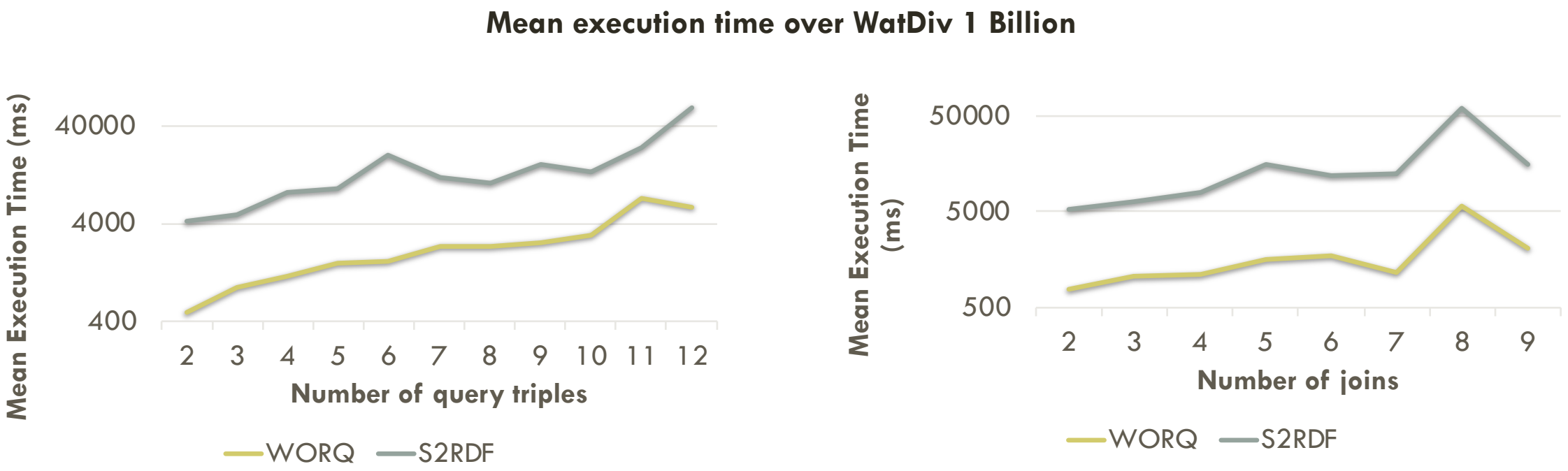
Query Execution Performance
Query Patterns



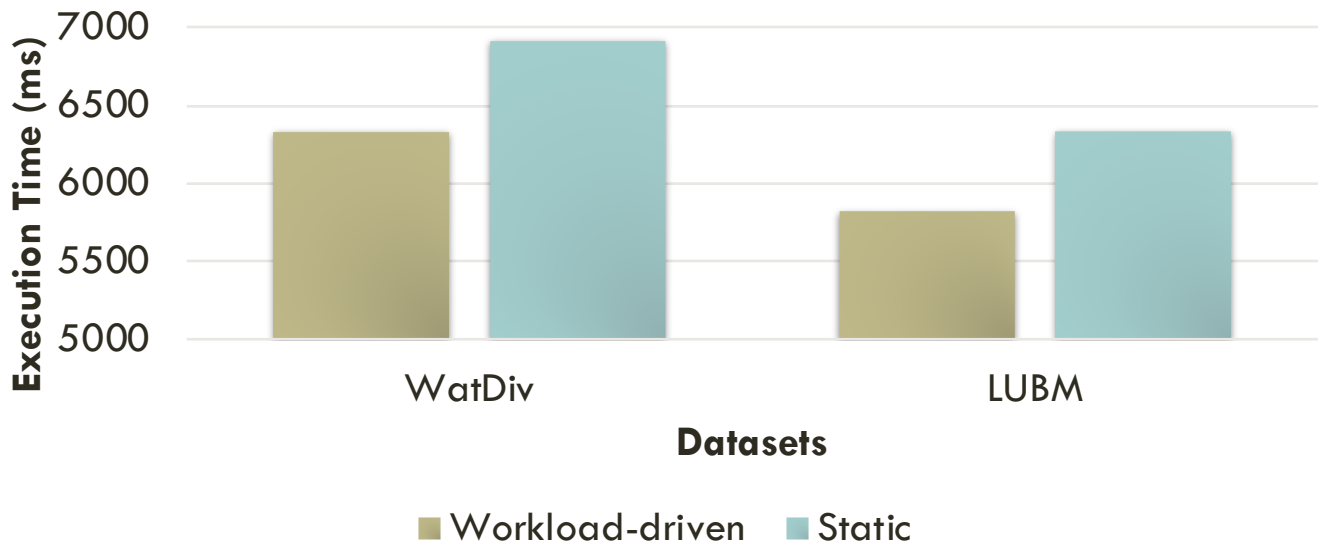


Query Execution Performance

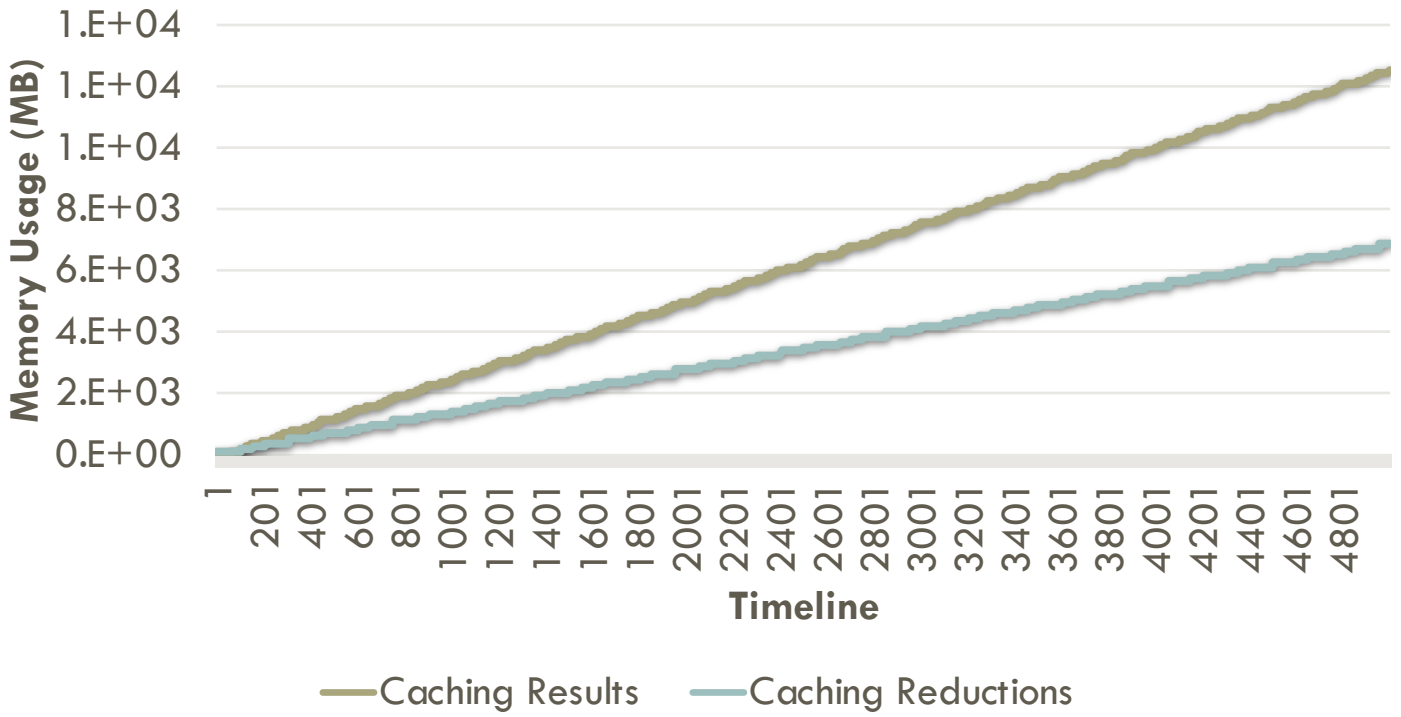
Query Patterns



Query Execution Performance
Workload-Driven Partitioning



Query Execution Performance
Caching



Performance of Unbound-Property Queries

System	BSO-Mean	BSO-Sum	BS-Mean	BS-Sum	BO-Mean	BO-Sum
WORQ	1.25 ms	10.49 min	4.18 ms	34.84 min	3.52 ms	29.34 min
RDF-Table	5.3 ms	44.44 min	3.80 ms	31.67 min	4.35 ms	36.26 min

(BSO) Bound Subject and Object
(BS) Bound Subject
(BO) Bound Object

Conclusion

- WORQ is an **online** method for **computing reductions** of RDF data using **Bloom filters**
- WORQ is a method for **workload-driven partitioning** that minimizes the network shuffling overhead
- WORQ demonstrates how **caching reductions** can boost the **query performance**
- WORQ helps answer RDF queries with **unbound properties** efficiently

Thank You !