Partout: A Distributed Engine for Efficient RDF Processing

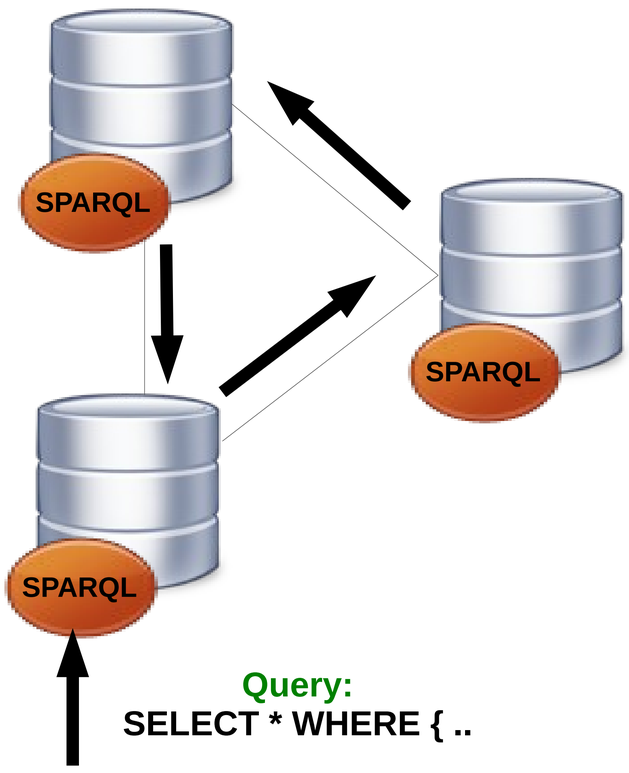
Luis Galárraga
Télécom ParisTech

Katja Hose
Aalborg University

Ralf Schenkel
University of Passau

Approaches for RDF Processing are either slow or not scalable

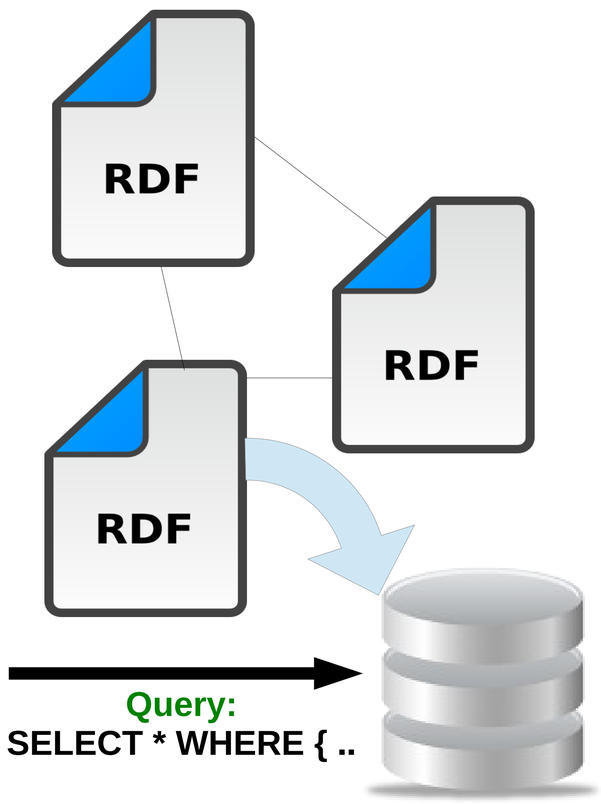
Query-time data retrieval



Data is up-to-date

No guarantees on response time

Data warehousing



Good response time

Not scalable

Solution: Distributed RDF processing on warehouse

Query load aware partitioning

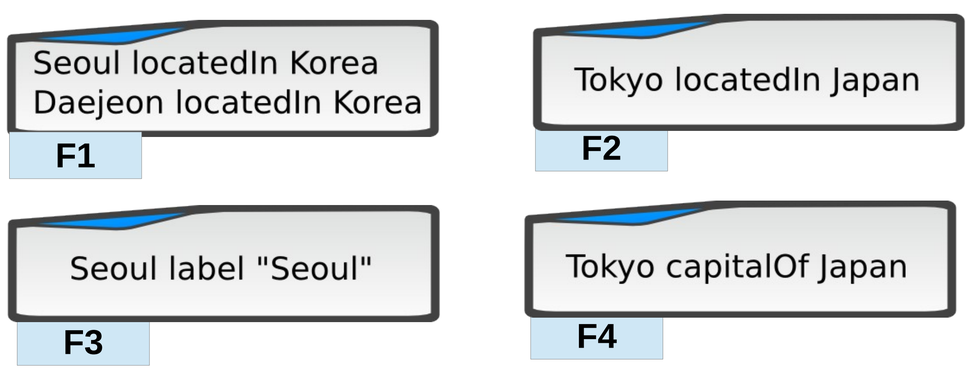
```
select ?name where {
  ?city locatedIn Korea
  ?city label ?name
}
```

1. Extract boolean predicates from **query load.**

Predicate = locatedIn
Object = Korea
Predicate = label
Predicate = capitalOf
Object = Japan

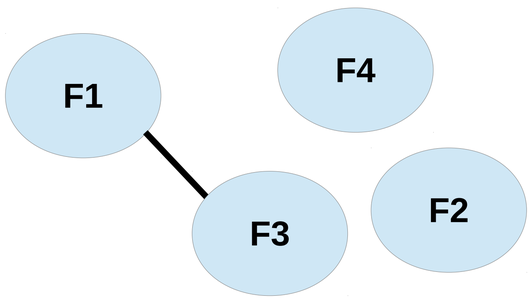
2. Apply **optimal horizontal partitioning**.

F1: Predicate = 'locatedIn' ^ Object = 'Korea'
F2: Predicate = 'locatedIn' ^ Object ≠ 'Korea'
F3: Predicate = 'label'
F4: Predicate = 'capitalOf'



Fragments allocation

4. **Fragment query graph** encodes fragment dependencies according to the query load.



5. Sort fragments in decreasing order by load and each time assign a fragment to the **most beneficial host** to guarantee: (1) **local execution for queries** and (2) **load balancing**.

F1 (load = 2 * 1 = 2)
F3 (load = 1 * 1 = 1)
F2 (load = 1 * 0 = 0)
F4 (load = 1 * 0 = 0)

Host 2

F4
F2

Load(F) = #-of-triples(F) * #-of-subqueries-using(F)

Benefit(Fragment, Host) ∝ Number of fragments already assigned to host connected with the new fragment according to fragment query graph.

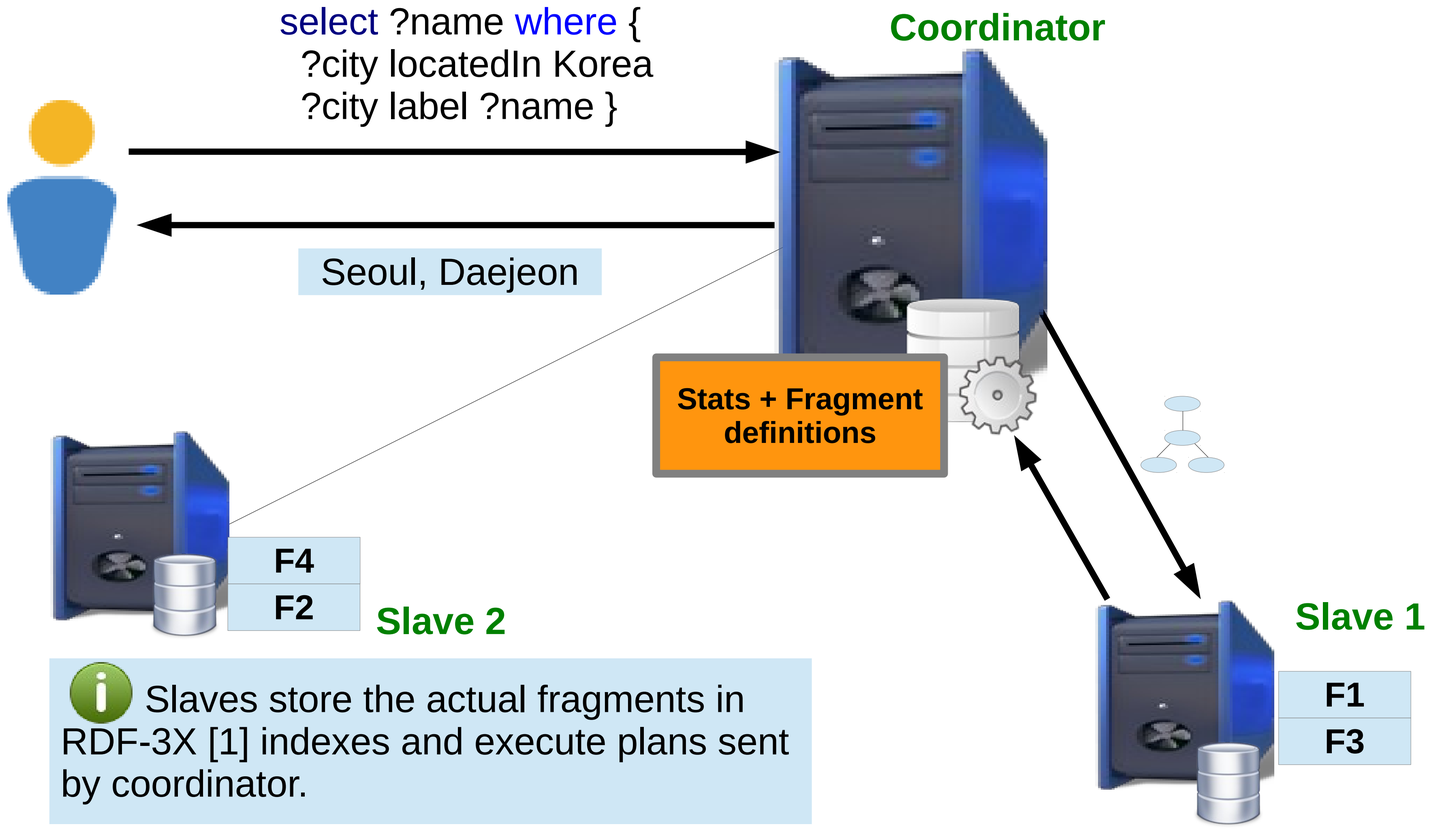
The inverse of the host's current load.

Host 1

F1
F3

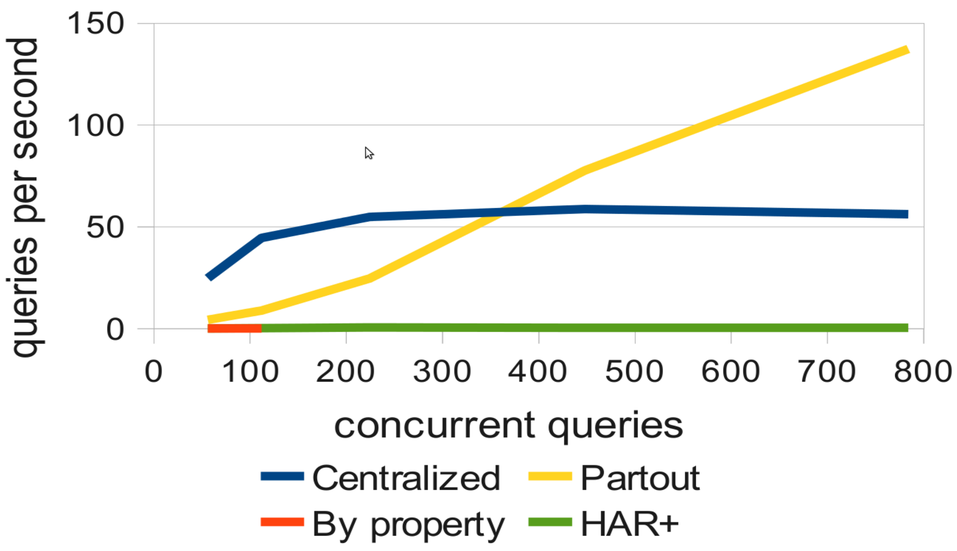
Query execution

Coordinator performs query planning, sends plan to slaves and gathers results.



Slaves store the actual fragments in RDF-3X [1] indexes and execute plans sent by coordinator.

Results on BTC 2008 dataset



Throughput compared against a centralized RDF index, a naive partitioning by property and the HAR+ approach, described in [2].

[1] T. Neumann and G. Weikum. The RDF-3X engine forscalable management of RDF data.VLDB J, 2010
[2] J. Huang, D. J. Abadi, and K. Ren. Scalable SPARQL. Querying of Large RDF Graphs. PVLDB, 2011