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 warehousing

Solution: Distributed RDF processing on warehouse

Query load aware partitioning

1. Extract boolean predicates from query load.
   - Predicate = locatedIn
   - Object = Korea
   - Predicate = label
   - Object = Japan

2. Apply optimal horizontal partitioning.
   - F1: Predicate = 'locatedIn' ^ Object = 'Korea'
   - F2: Predicate = 'locatedIn' ^ Object ≠ 'Korea'
   - F3: Predicate = 'label'
   - F4: Predicate = 'capitalOf'

3. 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
   - (2) load balancing.

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

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

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