Semantic Integrator

UNIMAN-UPM Collaboration
21 July 2010

Related Work: SPARQL Optimization

Complexity

- Complete complexity analysis

SPARQL Algebra equivalences

- Set of rewriting rules (6 groups)
- Based on relational (e.g. filter, projection pushing)
- SPARQL specific rules

Semantic Query Optimization

- Additional rewriting rules

See Schmidt et al. (2010)

Related Work: SPARQL Optimization

Join order optimization

- optimization of Basic graph patterns (BGPs)
- cost based optimization
- cost functions depend on cardinalities and selectivities estimation
- approaches
  - *Heuristics: e.g. bound joins more selective than unbound joins*
  - *Dynamic programming*

Related Work: SPARQL Optimization

Cost models & selectivities, e.g.

- $|A_G|$ is cardinality of expression A over graph G
- $sel(A_G) = |A_G|/|G|$ is the selectivity of A_G
- Join expression cardinality:
  - $|A_{Gi}.B_{Gj}| = |A_{Gi}| x |B_{Gj}|$ x $\min(sel(A_{Gi}), sel(B_{Gj}))$
- Cost functions
  - $c_{nested\text{-}loop\text{-}join}(A,B) = |A| \cdot |B| \cdot c_{compare}$

Different estimation methods, cost functions.

See Stocker et al. (2008), Görlitz et al. (2010), Quillitz et al (2008), Stuckenschmidt et al (2004)

Related Work: SPARQL Optimization

Index provision

- Data Source index: managed locally per source. Need to expose/export them.
- Virtual Data source index: Collected by other means and used by the federator
- Federation index: centralised index. Collected for all sources.

See Gorlitz et al (2010).

Related Work: SPARQL Optimization

Obtaining Statistics form Sources

- Source descriptions
  - Provided by sources (e.g. Quillitz et al.), statistical information
- Inspection
  - Exploration of sources, crawling (e.g. SemWIQ, Harth et al.)
- Result-based refinement
  - non intrusive
  - initially poor statistics
  - e.g. Görlitz et al (2010), Harth et al (2010).

Related Work: SPARQL Optimization

Data structures

- Histograms
  - Similar data items in buckets, count number of items
  - See Stocker et al., Neuman et al.
- QTree
  - combination of histogram and R-Tree, see Harth et al (2010)
- Compression techniques, e.g. Neuman et al.

See Görlitz et al. (2010)

Related Work

Learn about:

- SPARQL optimization rewriting rules
- Selectivity estimation for RDF stored data
- Cost-based optimization

Extend:

- Distributed SPARQL (DARQ, Quillitz et al (2008))
  - Most relevant to date.

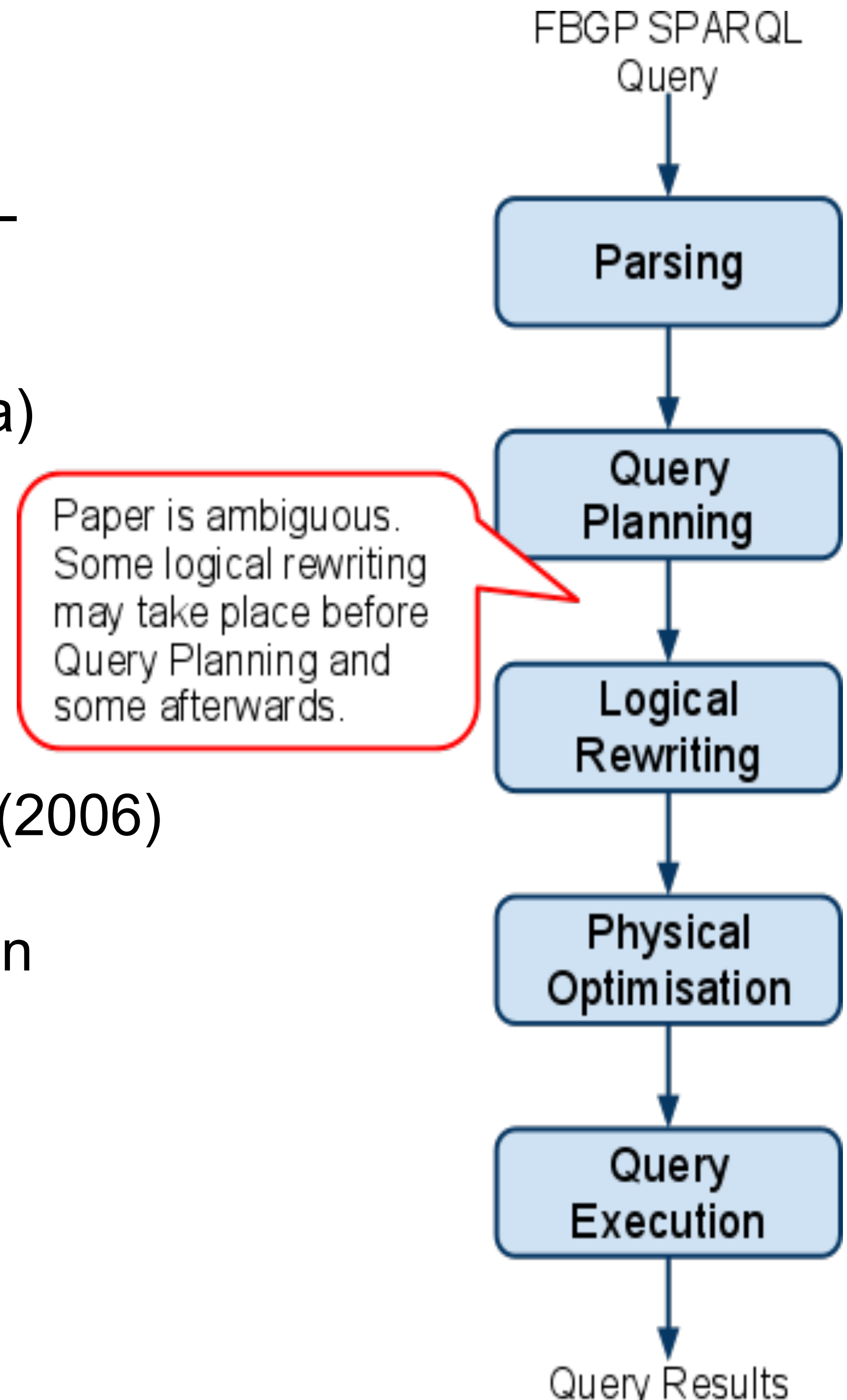
Adapt in a different setting (streams):

- DARQ
  - Estimation of cardinalities/selectivities
  - Revision of cost models
  - See next section

DARQ

Federates a set of distributed SPARQL endpoints into a virtual RDF store

- Service descriptions
  - Predicates (vocabulary/schema)
  - Cardinality estimates
  - Access patterns
- Query optimisation based on
  - Source assemblage
  - Logical rewriting
    - Rewrite rules of Pérez et al (2006)
    - Push filters down
  - Cost-based physical optimisation
    - Join re-ordering
    - Aim: data reduction
    - Costs based on cardinality estimates



DARQ: Service Description

- Data available
  - Set of predicates with subject/object constraints
- Access patterns
  - Required subject/object bindings
- Statistics
  - N_D: Number of triples
  - Optionally, for each predicate:
    - $n_D(p)$: Number of triples with predicate
    - $ssel_D(p)$: Subject selectivity for p
    - $osel_D(p)$: Object selectivity for p

Proposed Improvements

- Use SPARQL1.1 service descriptions and standard vocabularies
- Identify other useful statistics
- Extend for streaming sources

DARQ: Query Planning

- Source Assemblage
  - For each triple pattern, identify the sources which *potentially* contribute that pattern
    - Uses predicate information from service description
- Building Sub-queries
  - For each triple pattern
    - For each potential data source
      - Add triple pattern to source sub-query

Proposed Improvements

- Source assemblage after logical (and physical?) optimisation
- Graph pattern matching (see next slide)

Graph Pattern Source Matching

Idea: Only use a source if it can contribute *mulitple/chain-of* triple patterns.

Process:

- Extract source graph patterns, or include in service descriptions
- Only query a source if it can provide all the required information

Issue:

- Joins are expressed as chain-of triple patterns
- Need to ensure we do not eliminate cross-source joins

DARQ: Logical Rewriting

- Equivalence rules of Pérez et al (2006)
- Split and push filter constraints to sources

Proposed Improvements:

- Update to equivalence rules of Schmidt et al (2010)
- Extend for streaming language constructs (minimal effort)

DARQ: Physical Optimisation

Single triple pattern result size estimation:

$$\cos ts_d((s,p,o),b) = \begin{cases} n_d(p) & \text{if } \neg bound(s,b) \wedge \neg bound(o,b) \\ n_d(p) * osel_d(p) & \text{if } \neg bound(s,b) \wedge bound(o,b) \\ n_d(p) * ssel_d(p) & \text{if } bound(s,b) \wedge \neg bound(o,b) \\ 0.5 & \text{if } bound(s,b) \wedge bound(o,b) \end{cases}$$

Basic graph pattern cost:

$$\cos ts_d(T,b) = \min_{v \in T_{bound}} \left(\cos ts_d(v,b)\right) * \prod_{u \in T_{unbound}} \cos ts_d(u,b)$$

Proposed Improvements:

- Estimate if both subject and object are bound: min?
- Extend for streams

DARQ: Physical Optimisation – Joins

Two join implementations:

- Nested Loop Join: transfer cost estimate

$$C(q_1 \bowtie q_2) = |R(q_1)|c_t + |R(q_2)|c_t + 2c_r$$

- Bind Join: transfer cost estimate

$$C(q_1 \bowtie_B q_2) = |R(q_1)|c_t + |R(q_1)|c_r + |R(q_2')|c_t$$

where c_t and c_r are the transfer cost of one tuple and one query

Result size estimate:

$$|R(q_1 \bowtie q_2)| = |R(q_1)||R(q_2)|sel_{12}$$

where $sel_{12} = 0.5$

Proposed Improvements:

- Improve estimates
- Apply streaming cost models

References

- Olaf Görlitz and Steffen Staab. Federated Data Management and Query Optimization for Linked Open Data. Book chapter to appear. 2010.
- Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K., and Umbrich, J. 2010. Data summaries for on-demand queries over linked data. In WWW '10.
- Neumann, T. and Weikum, G. 2008. RDF-3X: a RISC-style engine for RDF. Proc. VLDB Endow. 1, 1 (Aug. 2008)
- J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. In ISWC 2006.
- Quilitz, B. and Leser, U. Querying distributed RDF data sources with SPARQL. In ESWC 2008.
- M. Schmidt, M. Meier, and G. Lausen. Foundations of SPARQL query optimization. In ICDT2010.
- Stocker, M., Seaborne, A., Bernstein, A., Kiefer, C., and Reynolds, D. SPARQL basic graph pattern optimization using selectivity estimation. In WWW '08.
- Stuckenschmidt, H., Vdovjak, R., Houben, G., and Broekstra, J. Index structures and algorithms for querying distributed RDF repositories. In WWW '04.