Related Work: Distributed SPARQL

DARQ

Quilitz, B. & Leser, U. (2008), 'Querying Distributed RDF Data Sources with SPARQL', The Semantic Web: Research and Applications , 524--538 .

Functionality

- Distributed SPARQL, sources are SPARQL endpoints
- Introduces service descriptions: capabilities of endpoints: i.e. what kind of triple patterns can be answered.
- Descriptions include statistical info: # of triples, selectivity of triple pattern
- Description using own vocabulary in RDF
- About the sub-queries (portions of queries that can be answered by a source), they can have multiple triple patterns on them, as the algorithm of sub-query generation describes this step.

Optimisations

- Logical:
  - Use rules for rewriting, merging patterns and replacing variables by constants from filter expressions
  - Move value constraints into sub-queries if possible.
- Physical: find best plan among all possible. Compare using a cost model, the goal is to reduce the amount of transferred data and transmissions.
  - Query result size estimation using the provided statistics
  - Transfer cost of joins calculated

Limitations

- No mapping/translation rules between the endpoints (no reconciliation)
- Vocabularies of endpoints may be different
- Need to explicitly provide service description for sources, how are statistics updated?
- Identify the SPARQL queries that can be handled
- Only bound predicates considered

Architecture

- Mediator based
- Sources are SPARQL endpoints (wrapped or not)

Status

- Research prototype, not continued.

SemWIQ

Langegger, A.; Wöß, W. & Blöchl, M. (2008), A Semantic Web Middleware for Virtual Data Integration on the Web, in '5th European Semantic Web Conference (ESWC2008)' , pp. 493-507

Functionality

- Distributed SPARQL, sources are SPARQL endpoints, can be Relational, CSV wrapped sources as well

- Catalog stores descriptions and statistics of sources
- statistics updated periodically by monitoring service
- Description using own vocabulary in RDF
- Finds sources that store instances of a determined type. Breaks the 'canonical' plan into sub-plans that will be executed by the sources.
- So the D2R system transforms the SPARQL sub-plan to SQL internally and executes it in case of wrapped relational sources.

Optimisations

- Push down of filter expressions
- Push down of optional graph patterns
- Push down of local joins where possible
- join and union reordering. All optimisations are 'being implemented' at the time of the writing of the paper, so no more details given.

Limitations

- Mappings through D2R, need to know language for wrapping relational sources
- Need to explicitly provide service description for sources, how are statistics updated?
- Identify the SPARQL queries that can be handled, some construct not supported, only SELECT.
- Requires all subject variable to be typed

Architecture

- Mediator-wrapper architecture
- Sources are SPARQL endpoints (wrapped or not)

Status

- Research prototype.

Networked Graphs

Schenk, S. & Staab, S. (2008), Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web, in 'WWW '08: Proceeding of the 17th international conference on World Wide Web' , ACM, New York, NY, USA , pp. 585--594

Jan Zemánek, Simon Schenk: Optimizing SPARQL Queries over Disparate RDF Data Sources through Distributed Semi-Joins. International Semantic Web Conference (Posters & Demos) 2008

Functionality

- Extend Named Graphs with SPARQL-based mechanism
- Views NG=(n,G,[G1,....Gn],m), G is graph, m is a mapping from list of networked graps to a view
- A view can be used to reference a graph in another remore RDF data source
- Syntactic extension to RDF for expressing views in RDF Graphs, essentially
- Sesame SAIL implementation capable of doing Networked graphs
- User explicitly says which queries are executed in which source (using the views)

Optimisations

- Distributed semi-joins used when dealing with remote nodes.

Limitations

- Requires users to build the queries explicitly stating where to fetch the data
- RDFstore based
- No statistics
- No mappings, reconciliation

Architecture

- Adding views to RDF stores, able to access a remote node
- Expanded Sesame SPARQL algebra, triple patterns evaluated remotely in remote endpoint.

Status

- Research prototype, implemented as extension to Sesame

SPARQL rewriting for Data Integration over Linked Data

Correndo, G.; Salvadores, M.; Millard, I.; Glaser, H. & Shadbolt, N. (2010), SPARQL query rewriting for implementing data integration over linked data, in 'EDBT '10: Proceedings of the 2010 EDBT/ICDT Workshops' , ACM, New York, NY, USA , pp. 1--11

Functionality

- Defined SPARQL query rewriting using rules
- Rules encoded in RDF, horn clauses
- Entity alignment defninition:
  - EA=(LHS,RHS,FD)
  - Left hand side, right hand side and functional dependencies
- Graph pattern rewriting, exploiting the rules:
  - (Triple(x;rdf:type;O1:WhiteWine) --> Triple(x;rdf:type; O2:Wine) and Triple(x;O2:hasColor;"White"))

Optimisations

- none, only basic translation done.

Limitations

- Only a translation or query transformation rule language
- Not considering aspects of DQP, optimisation, manageability of sources, statistics, other underlying data sources,
- No implementation of distributed SPARQL system

Architecture

Status

- Research demo

SPARQL queries over Web of Linked Data

Olaf Hartig, Christian Bizer, Johann-Christoph Freytag. Executing SPARQL Queries over the Web of Linked Data. ISWC09

Functionality

- Traversal of RDF links to discover relevant data
- Data sources discovered at query run time
  - Dereferencing URI's of the partial results
  - Using navigational nature of linked data
  - Need a starting point (data source)

Optimisations

- URI prefetching. Early dereferencing of URIs as soon as they are available.
- Non blocking iterators, avoid long waiting times for dereferencing URIs. Use of asynchronous pipelines that work in parallel. Enables the possibility of postponing the processing of certain solution items.

Limitations

- Only a basic graph patterns BGP (more complex on future works)
- Starting point needed, no complete results
- Infinite link discovery, unforeseen large data sets
- URI dereferencing may take long time

Status

- Recent research, Research demo
- Prototype SWCILib

Distributed SPARQL with OGSA-DQP

Carlos Buil Aranda, Óscar Corcho, Amy Krause: Robust Service-Based Semantic Querying to Distributed Heterogeneous Databases. DEXA Workshops 2009:74-78

Functionality

- Transform SPARQL queries to relational representation, OGSA-DQP is all relational.
- Extending OGSA-DQP for the RDF resource

Optimisations

- Applying distribution/optimisations: push projection, join reordering, table implosion, some of these optimisations already provided by the implementation of OGSA-DQP

Limitations

- Still work in progress, early stages

Status

- Recent research, publication to be submitted
- Some previous related work has been published by the author