- Requirements of Federation Infrastructure
- Federation Challenges
- Statistics Management
- Query Optimization and Execution
- Federation Insfrastructure for Linked Open Data
- Federator
  - Data Source Selection
  - Join Strategies
  - Data Catalog
  - Data Statistics
  - Item Counts
  - Full text indexing
  - Schema level indexing
  - Structural indexing
  - Index Size Reduction
  - Obtaining/Maintaining Data Statistics
  - Index Localization
- Query Optimization
- Data Source Mapping
- Query Execution Plan
- Optimization Fundamentals
- Optimization Strategies
- Query Plan Generation
- Improvement for federation
  - Streaming Results
  - Views
  - Performance Evaluation

A Linked Open Data infrastructure can have different characteristics:

- central or distributed data storage
- central or distributed data indexing
- independent or cooperative data sources

Central Repository:

- RDF data put into a single RDF store
- Advantage:
  - Query evaluation can be implemented efficiently due to optimized index structures
  - Original data sources are not involved in query evaluation

- Disadvantage:
  - If data at data sources changes frequently, data can become obsolete at central storage

Federation:

- Queries sent to a mediator which then executes query on behalf of user on the data sources
- Mediator maintains global index with statistics
  - Used for sending queries to data sources and query optimizatron

Peer to Peer data management:

- Data and index data maintained in a distributed fashion
- no central mediator

0.1 Requirements of Federation Infrastructure

- Declarative language
  - concise formulation of complex queries
- Data catalogue
  - map query expressions to Linked Open Data sources
  - mappings are also needed between vocabularies in order to extend queries with similar terms
- Query Optimizer
  - optimize the query execution:
    - minimize processing cost and the communication cost involved
- Data protocol
  - how queries and results are exchanged between data sources
  - e.g. SPARQL
- Provenance information
- Data changes at Data Sources: Data changes on Data Sources affect statistics and data entities which influences quality of the query optimization
- Scalability: implies two main challenges:
  - Efficient statistics management
  - effective query optimization and execution

	Central Repository	Data source federation
Data Location	local copies	at data sources
Meta data	data statistics	data source metadata
Query processing	local	local+remote yes
Require updates	yes (data)	yes (index)
Complete results	yes	yes
up-to-date results	maybe	yes

1 Federation Challenges

1.1 Statistics Management

- Accuracy vs. index size
- Updating statistics

1.2 Query Optimization and Execution

- execution order of query operators significantly influences the overall query evaluation cost
- Minimizing communication cost:
  - number of contacted data sources influences the performance of the query execution due to the communication overheads
  - Optimizing execution localization

2 Federation Insfrastructure for Linked Open Data

2.1 Federator

Responsible for: - maintaining meta data about known data sources - managing the whole query evaluation process

Actual query processing:

- query parsing

- query adaptation
- query mapping
  - selecting data sources
- query optimization
- query execution

2.1.1 Data Source Selection

- very likely that a single data source may only be able to return results for parts of the query
- respective query fragments have to be send to different data sources and the individual results have to be merged

2.1.2 Join Strategies

- Remote Join
- Mediator Join
- Semi-Join
- Bind join
- Filter chain

2.1.3 Data Catalog

- stores two different kinds of mappings
  1. relations between RDF terms like similarity with owl:sameAs and rdfs:seeAlso
     - can be used to adapt a query to different schemata or simple broaden search space
  2. Associates RDF terms or complex graph structures with data sources
     - Data catalog may be combined with data source statistics for ranking data sources and not querying irrelevant ones

2.1.4 Data Statistics

- can be used by query optimizer to estimate size of results
  - cost of joins
  - amount of data that needs to be transmitted
- trade-off between accuracy of the statistics and the required space for storing them

2.1.4.1 Item Counts

- counting data items
  - number of triples
  - number of individual instances of subject, predicate, and object
  - combinations of subject, predicate, and object

2.1.4.2 Full text indexing

- whole RDF graph in indexed
- Literals are suitable for indexing
- Prefix-tree can be build for URIs

2.1.4.3 Schema level indexing

- restrict the index to the data schema
  - types of instances and relations

2.1.4.4 Structural indexing

- restrict index to information about structure of RDF graph
- path structures or graph structures

2.1.4.5 Index Size Reduction Using: 1. Histogram 2. QTrees 3. R-Trees 4. Bloom Filters

2.1.4.6 Obtaining/Maintaining Data Statistics

- Data Dump Analysis
  - extract statistics from data dump
  - if new version of file, analysis redone
  - when data is changes, no announcement is made
- Source Description
  - Description about the data
    - VoID
      - provides some statics
      - not enough expressive
    - Service Description in DARQ

- Source Inspection
  - Send SPARQL Queries (like ASK Queries)
  - aggregates can be to find data statistics -Result-based Refinement
  - extract metadata from result from data source
  - lack of statistics in the beginning

2.1.4.7 Index Localization

1. Data Source Index: can be obtained as a VoiD file for a particular data source
2. Virtual data source index maintained at the federator
3. Federation Index: centralized index maintained at the data source about information of all known data sources
4. Distributed federation index
   - federation index may be partitioned among all data sources

3 Query Optimization

- objective:
  - find a query execution plan
  - which minimizes the processing cost of the query
  - and the communication cost for transmitting query and results between mediator and Linked Data sources

3.1 Data Source Mapping

- Data source selection

3.2 Query Execution Plan

- query execution plan is an executable query plan where logical operators are replaced by physical operators, e.g. join may be implemented as:
  - nested loop join
  - sort-merge join
  - hash join
  - semi-join
  - bind join
- Two main structure of Query Execution Plan:

  - Left deep trees
    * pipelined execution
    * starting with leftmost leaf node
    * operators are evaluated one after another
    * results are passed as input to the parent node until the root node is reached
  - Bushy trees
    * allow for parallel execution as sub trees can be evaluated concurrently

3.3 Optimization Fundamentals

- Objective: find a query execution plan with minimal cost in terms of processing cost and communication cost
- query execution time is the main costmeasure
- query execution time = processing cost + communication cost
- optimization strategies rely on the same two basic measures for estimating the cost of a query exe- cution plan:
  - cardinality: estimated number of elements in a result set which are returned for a query expression
  - selectivity:
    * defines the estimated fraction of elements which match a query expression
    * Range: 0 or 1
    * 0: most selective
    * 1: least selective

RDF Graph Cardinality = Number of triples in the graph

RDF Term Selectivity = Number of triples in the Graph where which contains the term / RDF Graph Cardinality

Triple Pattern Selectivity = Product of the selectivity of all terms in the triple

Triple Pattern Cardinality = Triple Pattern Selectivity * RDF Graph Cardinality

3.4 Optimization Strategies

- different optimization strategies
- a trade-off between finding the optimal query plan and finding a query plan quickly
- Optimization Strategies:
  - Static Optimization: Generates one plan and stick

  - Dynamic Optimization: may change query plan during execution due to updated statistics

3.5 Query Plan Generation

- Generate query plan
- evaluate plan using a cost model

3.6 Improvement for federation

3.6.1 Streaming Results

- execution chain of operators can be become a bottleneck if have to wait for results
- SPARQL Standard requires that all information regarding query be present before query optimization can be performed
- Partial result cannot yet be performed using SPARQL

3.6.2 Views

- allows for data abstraction
- simplify querying of complex relations
- views to abstract data schema or data sources

3.6.3 Performance Evaluation

- evaluation framework are required to evaluate federated systems