NSWI144 – Linked Data – Lecture 7 – 29th November 2011

Querying and Indexing

Martin Svoboda

Faculty of Mathematics and Physics

Charles University in Prague

Querying Systems

- Issues
  - Physical storage
  - Index structures
  - Query processor
- Problems
  - Data scalability, distribution and dynamicity

Querying Systems

- Architecture
  - Local
    - Efficient processing
    - Independent data
    - Storage requirements
  - Distributed
    - Runtime requests
    - Up-to-date data
    - Network throughput

Querying Systems

- Approach categories
  - Querying systems
    - Local or distributed data
    - Structural queries
    - Complete results
  - Searching engines
    - Global data cloud
    - Full text queries
    - Imprecise results

Storage Solutions

- Storages for RDF triples
  - Relational databases
    - Decades of research
    - Implemented systems
  - Native approaches
    - Novel approaches
    - Appropriate logical model

Relational Databases

- Triple table
  - Schema
    - Table(Subject, Predicate, Object)

Subject	Predicate	Object
S1	P1	O1
S1	P1	O2
S2	P2	O2

  - Indices
    - Standard relational indices over selected columns

Relational Databases

- Triple table
  - Advantages
    - Without NULL values
    - Multi-value properties
    - Symmetric access patterns
  - Disadvantages
    - Less selective queries
    - Problematic self joins

Relational Databases

- Property tables
  - Idea
    - Combining all/some properties for a given subject
      - Disjoint or potentially overlapping sets of properties
  - Schema
    - $Table_1(Subject, P_1, P_2, \ldots)$
    - …

Subject	P1	P2
S1	O1	*NULL*
S2	*NULL*	O2

Relational Databases

- Property tables
  - Advantages
    - Fewer join operations
    - Follows relational model
  - Disadvantages
    - Clustering is not trivial
    - Multi-value properties
    - Potentially very sparse
    - Asymmetric access patterns

Relational Databases

- Binary tables (vertical partitioning)
  - Idea
    - Separate table for each predicate
  - Schema
    - P_1(Subject, Object)
    - …

P1	
Subject	**Object**
S1	O1
S1	O2

P2	
Subject	**Object**
S2	O2

Relational Databases

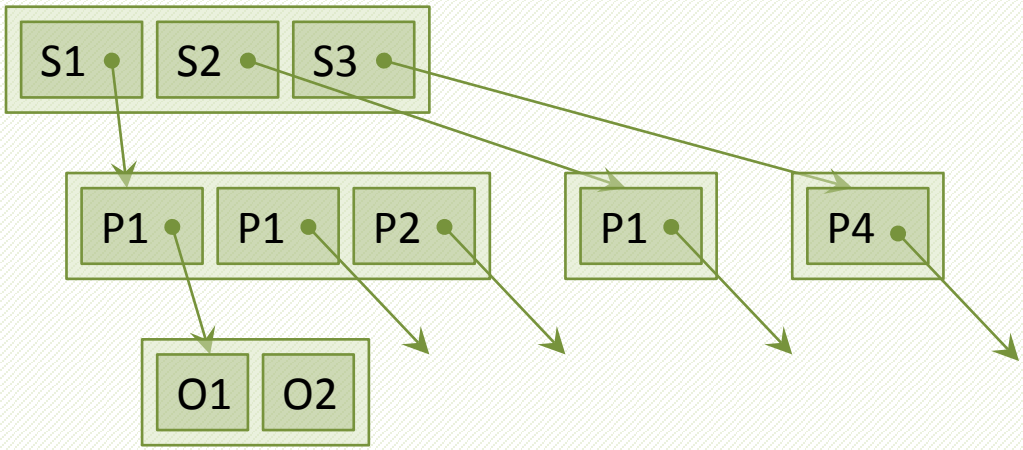
- Binary tables
  - Advantages
    - Multi-value properties
    - Efficient column storages
  - Disadvantages
    - Unbound query predicates
    - Large number of tables
    - Asymmetric access patterns

Sextuple Indexing

- Paper
  - Cathrin Weiss et al.: **Hexastore: Sextuple Indexing for Semantic Web Data Management**
- Index
  - Local database of RDF triples
  - Based on ordered nested lists
  - Supports all access patterns

Sextuple Indexing

- Index model
  - SPO, SOP, OSP, OPS, PSO, POS nested lists
  - Duplicated lists are shared
    - For example: P lists for corresponding SO and OS



BitMat Index

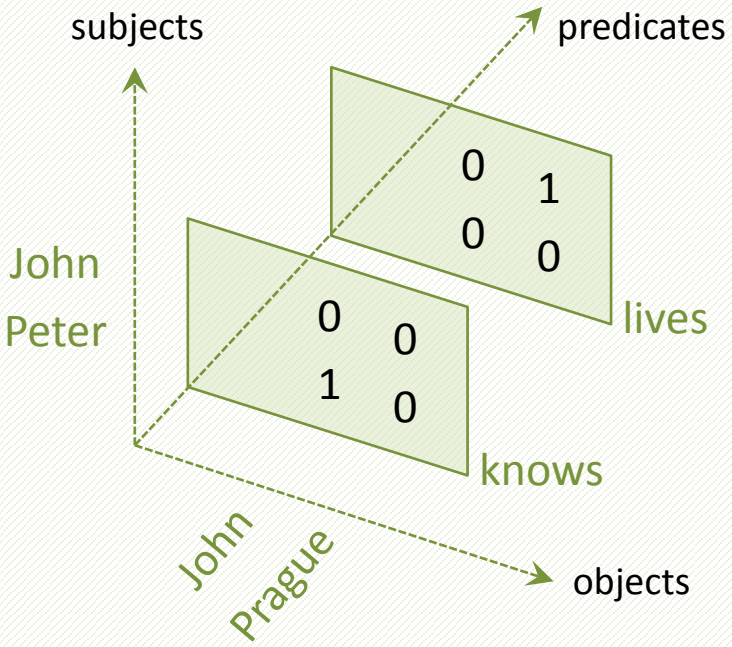
- Paper
  - Medha Atre et al.: **Matrix "Bit"loaded: A Scalable Lightweight Join Query Processor for RDF Data**
- Problem
  - Local database with RDF triples
  - Conjunctive SPARQL queries

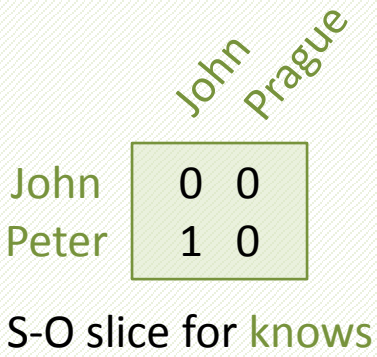
BitMat Index

- Queries
  - Conjunctive queries
    - A few patterns with joining variables
  - Query selectivity
    - Low selectivity of both patterns and joining
- Objectives
  - Low memory requirements
    - Without intermediate result materialization

BitMat Index

- Index model





S-O slice for knows

BitMat Index

- Index model
  - Terms
    - Active domains of all subjects, predicates and objects
    - Assignment of unique integer identifiers
  - Index
    - 3-dimensional matrix with bit values 0 or 1
    - Dimensions for subjects, predicates and objects
  - Slices
    - SO and OS for each predicate
    - PO for each subject and PS for each object

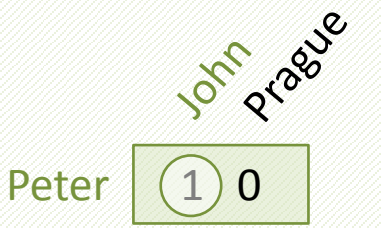
BitMat Index

- Index implementation
  - Ordinary file
  - Compression
    - Slices are stored per individual rows
    - Rows are compressed using bit runs
  - Query evaluation
    - All operations over compressed bit runs

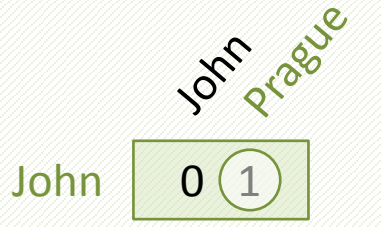
BitMat Index

- Query processing
  - Initialization
    - Loading required index components
  - Optimizations
    - Pruning loaded index components
  - Joining
    - Stream joining of individual patterns
    - Based on nested loops algorithm
      - Idea from relational databases
      - Starting with the most selective pattern

(Peter knows ?P)
(?P lives Prague)



S-O map for knows



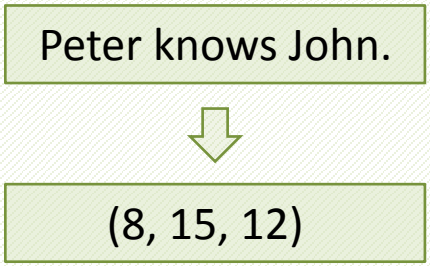
S-O map for lives

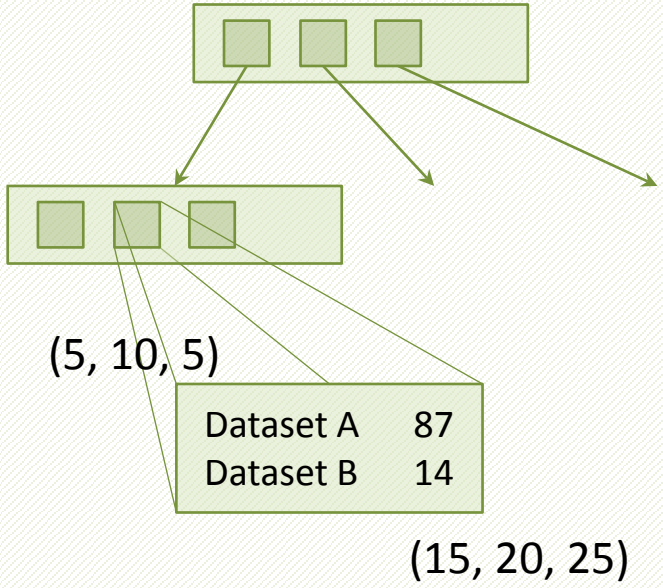
Data Summaries

- Paper
  - Andreas Harth et al.: **Data Summaries for On-Demand Queries over Linked Data**
- Problem
  - High number of distributed datasets
  - Conjunctive SPARQL queries
  - Source selection algorithm

Data Summaries

- Index model
  - Numeric space
    - 3-dimensional numeric space
    - Transforming triples to points
    - Based on hash functions





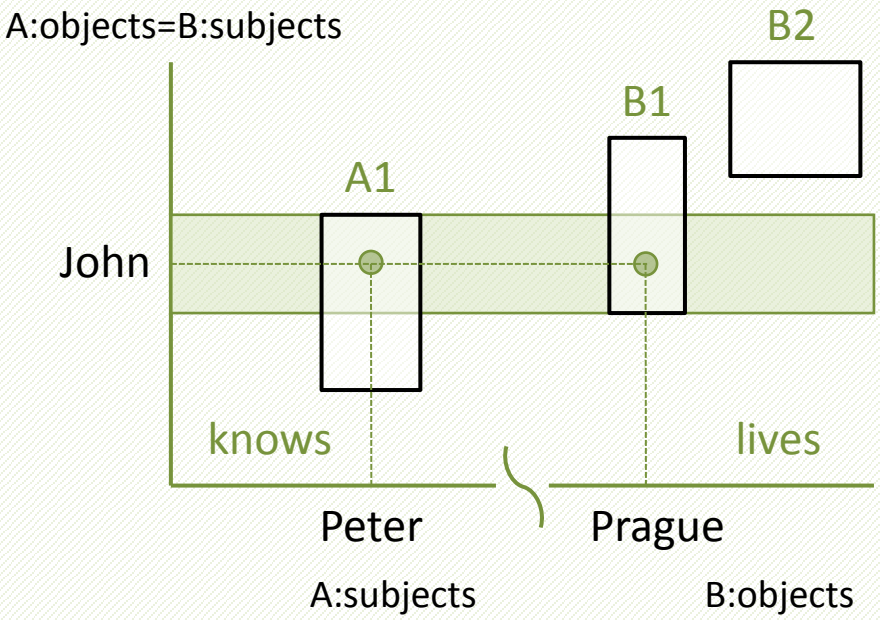
Data Summaries

- Q-tree
  - Based on R-trees and histograms
  - Internal nodes
    - Set of bounding boxes
      - These boxes may generally overlap themselves
  - Leaf nodes
    - Buckets with summaries
      - For each dataset a number of its corresponding triples
  - Features
    - Fixed (parameterized) size

Data Summaries

- Selection algorithm
  - Query transformation
    - Intervals for variables
  - Source selection
    - Individual patterns
      - Index traversal
      - Sets of buckets
    - Inductive joins
      - Required overlapping
  - Query processing

(Peter knows ?P)
(?P lives Prague)



Search Engines

- Swoogle
  - Paper
    - Li Ding et al.: **Swoogle: A Search and Metadata Engine for the Semantic Web**
  - Idea
    - Search engine for semantic documents
      - Data documents / ontologies and schemata
    - Provided functionality
      - Document metadata
    - Based on IR techniques

Search Engines

- SWSE
  - Paper
    - Andreas Harth et al.: **SWSE: Answers Before Links**
  - Idea
    - Search engine for RDF triples with context
    - Provided functionality
      - Keyword matching
      - Concept filtering

Search Engines

- Sindice
  - Paper
    - Eyal Oren et al.: **Sindice.com: A Document-oriented Lookup Index for Open Linked Data**
  - Idea
    - Search engine for semantic documents
    - Provided functionality
      - Keyword matching
      - Inverse functional properties

Common Observations

- String compression
  - Repeating string values
    - URIs and literals
  - Unique integer identifiers
    - Efficient processing
    - Space requirements
  - Translation maps
    - Both directions
    - Based on B-trees

Common Observations

- Data pruning
  - Idea
    - Query optimization
    - Relevant data
  - Methods
    - Filtering selections
    - Join ordering
  - Problem
    - Partial knowledge

Open Problems

- Data distribution
  - Motivation
    - Datasets are distributed
    - Appropriate compromise
  - Problems
    - Network drawbacks
    - Space requirements
    - Independent datasets

Open Problems

- Data scalability
  - Motivation
    - Web of Data size explosion
      - September 2011:
      - 295 datasets, 31 billion triples, 504 million links
  - Problems
    - Scalable storages and indices
    - Efficient query evaluation
    - Quality, provenance and trust

Open Problems

- Data dynamicity
  - Motivation
    - Data tend to ageing
  - Problems
    - Continuous updates
    - Dynamic structures

Conclusion

- Querying systems
  - Local / distributed approaches
- Storage solutions
  - Relational databases / native approaches
- Indexing methods
  - Existing techniques
  - Common practices
  - Open problems