Linked Data Indexing Methods

Martin Svoboda, Irena Mlýnková
Charles University in Prague

Problem

Despite the research effort in recent years, several questions in the **area of Linked Data indexing** and querying remain open, not only since the amount of Linked Data globally available significantly increases each year. This poster attempts to introduce **advantages and disadvantages of the existing approaches** and outline several issues related to our ongoing research effort, the **proposal of an efficient querying framework** over Linked Data. In particular, our goal is to focus on **large amounts of distributed and highly dynamic data**.

Issues

System Architecture

The fundamental question of each querying system is the **mutual relationship between physical storages, index structures and querying capabilities**. Local approaches may enable efficient query processing, while online evaluation can assure up-to-date results. Anyway, we cannot ignore the data distribution aspect.

Physical Storage

Although native approaches for querying RDF data could generally represent more efficient solutions, relational databases benefit from decades of experience and research results. We can even find indexing solutions that do not need any physical database layer and all required data are stored directly in the index.

Querying Language

Querying can be based on **full text searching or graph patterns matching**. In order to evaluate queries effectively, we have to use a variety of optimizations. The problem is that we often need to rely on heuristics or imprecise statistics.

Dimensions

- **System scope**: *local*, *distributed* and *global* approaches
- **Updates possibility**: *static* or *dynamic* index structures
- **Index content**: indexing pure *data* or *statistics* about them
- **Data items**: *triples*, *quads* with context or other *sources*
- **Querying layer**: *syntactic*, *structural* or *semantic* querying
- **Query models**: *full text* querying or graph *patterns* matching
- **Index items**: *keywords*, *triples*, *quads*, *trees*, *paths* or *areas*
- **Access patterns**: *universal* or *dedicated* approaches

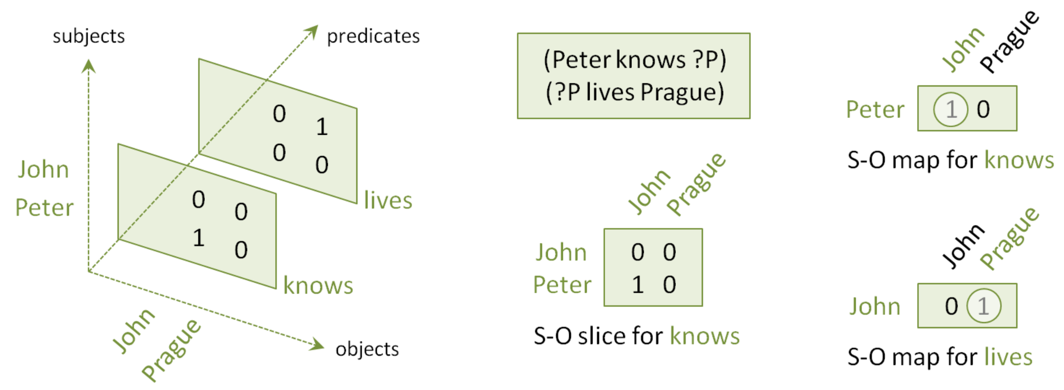
Approaches

RDF-3X Engine [8]

The core of the stream processor RDF-X is based on 6 B^+-tree indices for all *SPO*, *SOP*, *OSP*, *OPS*, *PSO* and *POS* access patterns. Additionally, the authors also use indices with statistics (*S*, *P*, *O*, *SP*, *PS*, *PO*, *OP*, *SO* and *OS* projections) and selectivity histograms and statistics for pre-computed path or star patterns.

BitMat Index [2]

The index model of BitMat approach is based on a matrix with three dimensions for *S*, *P* and *O* values (terms are translated to identifiers, which are used as matrix indices). Each cell contains a bit value equal to 1 if and only if the given triple is stored in the database, otherwise value 0. The index is organized as an ordinary file with all *SO*, *OS*, *PO* and *PS* slices stored using a bit run compression over individual slice rows.

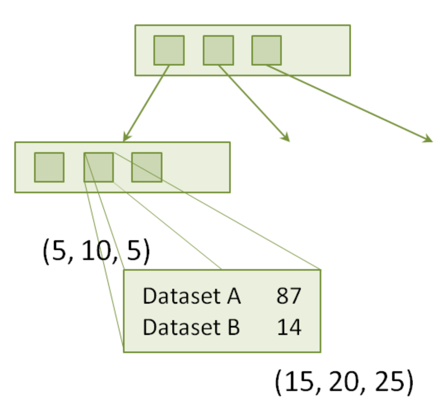


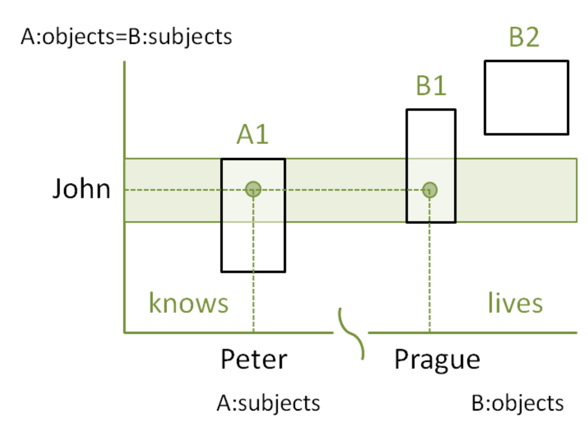
Structure Index [12]

The parameterised index approach is based on bisimilarity relations, putting in a relation such two vertices of the data graph that share the same outgoing and ingoing edges (reflecting only predicates). Vertices from the same equivalence class have the same characteristics and, therefore, prompted queries can first be evaluated over these classes to prune required data.

Data Summaries [6]

The purpose of a data summary index is to enable the source selection over distributed data sources. Data triples are modelled as points in a 3-dimensional space (*S*, *P*, and *O* coordinates are derived by hash functions). The index structure is a QTree based on standard R-Trees. Internal nodes act as minimal bounding boxes for nested nodes, leaf nodes contain statistics about data sources, not data triples themselves.





Comparison

Approach name	Scope	Data	Queries	Index
Local querying approaches				
Harth 2005 Quad index [5]	Local	Quads	Graphs YARS QL	Quads Text
Abadi 2007 Partitioning [1]	Local	Triples	Graphs SQL	Paths
Neumann 2008 RDF-3X [8]	Local	Triples	Graphs SPARQL	Triples
Weiss 2008 Hexastore [14]	Local	Triples	Graphs SPARQL	Triples
Atre 2010 BitMat index [2]	Local	Triples	Graphs SPARQL	Triples
Liu 2005 Path index [7]	Local	Triples	Paths SPARQL	Paths
Udrea 2007 GRIN index [13]	Local	Triples	Graphs SPARQL	Circles
Tran 2010 Structural index [12]	Local	Triples	Graphs SPARQL	Graphs
Distributed querying approaches				
Stuckenschmidt 2004 Repositories [11]	Dist.	Sources	Trees SeRQL	Paths
Quilitz 2008 DARQ [10]	Dist.	Sources	Graphs DARQ	Services
Harth 2010 Summaries [6]	Dist.	Sources	Graphs SPARQL	Boxes
Global searching approaches				
Ding 2004 Swoogle [3]	Global	Files	Full text	Text
Harth 2007 SWSE [4]	Global	Quads	Full text	Text
Oren 2008 Sindice [9]	Global	Files	Full text	Text

Observations

String Compression

The common idea is to transform URIs and literals into unique integer identifiers, store original values in special translation maps, and use these identifiers in the index instead.

Data Pruning

Query evaluation can be supported by data filtering selections or join ordering. Generally, we want to avoid processing of irrelevant data whenever possible.

Challenges

Distribution

Finding an appropriate **compromise between processing local or distributed data** forms one of the most important questions. Maintaining local copies of data may benefit from convenient conditions for efficient query evaluation; however, we are not always able or allowed to gather the data under our control.

Scalability

Even though existing approaches work with large sets of data, experiments performed using various sets of data, queries and prototype implementations of discussed solutions imply that we are still not able to sufficiently flatten performance of such approaches and the **explosion of the Web of Data size**.

Dynamicity

Data on the Web of Data significantly tends to aging. We especially need not only to handle simple **data modifications**, but also deal with **broken links** and attempt to anticipate or correct them. Unfortunately, the problem is that **index structures are often static** and do not allow any further modifications like inserts, updates or deletes.

Quality

The increasing number of globally available data on the Web also causes issues of **data quality, provenance and trust**. Especially in the context of global search engines we need to propose accurate metrics for determining relevance of particular query results. For this purpose we can utilize **knowledge and relationships from social networks**.

References

[1] Abadi, D.J., Marcus, A., Madden, S.R., Hollenbach, K.: **Scalable Semantic Web Data Management Using Vertical Partitioning**. In: Proc. of the 33rd Int. Conf. on Very Large Data Bases. pp. 411-422. VLDB Endowment (2007)

[2] Atre, M., Chaoji, V., Zaki, M.J., Hendler, J.A.: **Matrix "Bit" loaded: A Scalable Lightweight Join Query Processor for RDF Data**. In: Proc. of the 19th Int. Conf. on World Wide Web. pp. 41-50. ACM, New York, NY, USA (2010)

[3] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: **Swoogle: A Search and Metadata Engine for the Semantic Web**. In: Proceedings of the 13th ACM Int. Conference on Information and Knowledge Management. pp. 652-659. CIKM '04, ACM, New York, NY, USA (2004)

[4] Harth, A., Hogan, A., Delbru, R., Umbrich, J., O'Riain, S., Decker, S.: **SWSE: Answers Before Links**. In: Proc. of the Semantic Web Challenge 2007 co-located with ISWC + ASWC 2007. vol. 295, pp. 136-144. CEUR-WS.org (2007)

[5] Harth, A., Decker, S.: **Optimized Index Structures for Querying RDF from the Web**. In: Third Latin American Web Congress, 2005. LA-WEB 2005. IEEE (2005)

[6] Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: **Data Summaries for On-demand Queries over Linked Data**. In: Proc. of the 19th Int. Conf. on World Wide Web. pp. 411-420. ACM, NY, USA (2010)

[7] Liu, B., Hu, B.: **Path Queries Based RDF Index**. In: Proceedings of the First International Conference on Semantics, Knowledge and Grid. pp. 91-93. IEEE Computer Society, Los Alamitos, CA, USA (2005)

[8] Neumann, T., Weikum, G.: **RDF-3X: A RISC-style Engine for RDF**. Proc. VLDB Endow. 1, 647-659 (August 2008)

[9] Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: **Sindice.com: A Document-oriented Lookup Index for Open Linked Data**. Int. J. of Metadata, Semantics and Ontologies 3(1), 37-52 (2008)

[10] Quilitz, B., Leser, U.: **Querying Distributed RDF Data Sources with SPARQL**. In: The Semantic Web: Research and Applications. LNCS, vol. 5021, pp. 524-538. Springer Berlin / Heidelberg (2008)

[11] Stuckenschmidt, H., Vdovjak, R., Houben, G.J., Broekstra, J.: **Index Structures and Algorithms for Querying Distributed RDF Repositories**. In: Proc. of the 13th Int. Conf. on World Wide Web. pp. 631-639. ACM, USA (2004)

[12] Tran, T., Ladwig, G.: **Structure Index for RDF Data**. In: Workshop on Semantic Data Management (SemData@VLDB) 2010 (2010)

[13] Udrea, O., Pugliese, A., Subrahmanian, V.S.: **GRIN: A Graph Based RDF Index**. In: Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2. pp. 1465-1470. AAAI Press (2007)

[14] Weiss, C., Karras, P., Bernstein, A.: **Hexastore: Sextuple Indexing for Semantic Web Data Management**. Proc. VLDB Endow. 1, 1008-1019 (August 2008)

ISSLOD 2011, Leipzig, Germany, 12.—16. 9. 2011
svoboda@ksi.mff.cuni.cz
This work was supported by the Charles University Grant Agency grant 4105/2011