SAINT-DB: A structural indexing based triple store *

George H.L. Fletcher, Jan Hidders, Yongming Luo, François Picalausa, Stijn Vansummeren, and Paul De Bra

1Eindhoven University of Technology, The Netherlands
2Université Libre de Bruxelles, Belgium
3Delft University of Technology, The Netherlands

We present the SAINT-DB native RDF database system [2], which is designed to investigate the possibility of bridging structural indexes with the state-of-the-art value based triple stores. In SAINT-DB, RDF graphs are represented as an edge-labeled graph where triples are represented as nodes and where edges like $t_1 \xrightarrow{s} t_2$ indicate that the subject of triple $t_1$ equals the object of $t_2$. Over these graphs we can establish a precise coupling between the expressive power of SPARQL fragments and graph simulations over these graphs [1]. The graph simulations can then be used to determine if triples should be clustered together in the structural index. The structural index itself formally consists of a partition of the set of RDF triples, together with a set of labeled edges over the partition blocks. Here, an edge like $B_1 \xrightarrow{E} B_2$ between partition blocks $B_1$ and $B_2$ indicates that the subject of some triple in $B_1$ equals the object of some triple in $B_2$.

SAINT-DB extends the open source RDF-3X code-base to accommodate quads instead of triples, i.e., all storage and statistical structures are extended from triples to quads of the form $(S, P, O, C)$, using the fourth position $C$ of each quad to hold the integer identifier of the partition block to which the input RDF triple $(S, P, O)$ belongs in the structural index. We also materialize, in a reserved partition block 0, the edges $B_1 \xrightarrow{E} B_2$ of the structural index as a set of quads of the form $(B_1, E, B_2, 0)$ where $B_1$ and $B_2$ are partition block identifiers.

Several sophisticated query processing strategies have been investigated over the quad-based representation of structural indexes and the underlying RDF data set. Major optimizations include using statistics to decide when to first perform query evaluation on the index graph, to restrict the search space, or to by-pass this stage altogether and go directly to the underlying triple graph. We have performed extensive experimental studies of these strategies, on both synthetic and real world RDF graphs. On queries over highly structured data graphs, SAINT-DB has exhibited up to a 15-fold speedup in disk-access costs, over RDF-3X. In situations advantageous to value-based approaches (i.e., highly selective queries over relatively unstructured graphs), SAINT-DB query evaluation costs were shown to be competitive with RDF-3X. These initial investigations indicate that it is indeed advantageous and worthwhile to further study the use of structural indexing in RDF storage and query processing.

References


*The research of the prospective presenter Yongming Luo is supported by NWO SeeQR project (project number 612.001.005, Free Competition of the NWO Division for Physical Sciences). For more information please visit http://www.win.tue.nl/~yluo/seequer/
†{g.h.l.fletcher, y.luo, P.M.E.D.Bra}@tue.nl
‡{fpicalau, stijn.vansummeren}@ulb.ac.be
§{a.j.h.hidders}@tudelft.nl
SAINT-DB: A structural indexing based triple store

1. Motivation

State-of-the-art RDF data management systems use value-based indexes (e.g., B-Tree) as their storage backends, without considering many of the structural aspects of RDF datasets.

2. Main idea

What if we introduce structural information of RDF graph into triple stores?

- Structural indexes have been successfully applied in XML data management.
- We have established similar structural equivalence criteria for practical fragments of SPARQL [1].

3. Design

Store quads instead of triples in the back-end, to materialize the structural index.

4. Results [2]

Experiment results show that it is indeed advantageous and worthwhile to further study the use of structural indexing in RDF storage and query processing.

References
