SPARQL Query Optimization Using Selectivity Estimation

Abraham Bernstein, Markus Stocker, and Christoph Kiefer

Department of Informatics, University of Zurich, Switzerland
{bernstein,stocker,kiefer}@ifi.uzh.ch

Abstract. This poster describes three static SPARQL optimization approaches for in-memory RDF graphs: (1) a selectivity estimation index (SEI) for single query triple patterns; (2) a query pattern index (QPI) for joined triple patterns; and (3) a hybrid optimization approach that combines both indexes. Using the Lehigh University Benchmark (LUBM), we show that the hybrid approach outperforms other SPARQL query engines such as ARQ and Sesame for in-memory graphs.

1 Introduction

In recent years, the RDF query language SPARQL has gained increasing popularity.[1] The performance of SPARQL queries have, however, not yet been systematically investigated. Therefore, this poster focuses on static SPARQL query optimization for in-memory RDF graphs. Our three presented optimization approaches are inspired from join re-ordering strategies using selectivity estimation [2, 3]. The early execution of triples with low selectivity heavily reduces the number of intermediate results boosting query performance.

Our first approach, SEI, focuses on selectivity estimation of single triple patterns based on statistics of the queried data. The second approach, called QPI, indexes joined triple patterns based on the ontology schema. Last, the combined SEI-QPI hybrid optimizer results in being the most promising approach.

The proposed poster is divided into two primary parts: first, we explain the rational behind our three approaches, and second, we present some preliminary results using the queries from the Lehigh University Benchmark [1].

2 Theory and Implementation of OptARQ

Our approaches are implemented in a suite called *OptARQ* that bases on Jena ARQ.[2] OptARQ's principle is *to find the query execution plan that minimizes the intermediate result set size of each triple pattern by join re-ordering strategies.* In the following, we present our three approaches to achieve this goal.

SEI – Selectivity Estimation Index. SEI focuses on selectivity estimation of single triple patterns. Inspired by [3], the selectivity $sel(T)$ of a triple pattern T is defined as the fraction of triples matching T. Selectivity estimation serves to

[1] http://www.w3.org/TR/rdf-sparql-query/
[2] http://jena.sourceforge.net/ARQ/

minimize the number of intermediate results: *the smaller the selectivity the less intermediate results it is likely to produce and the earlier it should be executed in the query execution plan.* Selectivities are estimated using statistical information about the queried RDF data. As we treat selectivities as probabilities, the overall selectivity for a triple pattern T can be defined as $sel(T) = sel(S) \times sel(P) \times sel(O)$ where S, P, and O are the subject, predicate, and object of the pattern.[3] We estimate the selectivities of a subject S as $\frac{1}{|R|}$ (R = number of resources) and of a predicate P as $\frac{|T_P|}{|T|}$ where $|T_P|$ is the (exact) number of triples matching predicate P and $|T|$ the total number of triples in the data. SEI estimates the selectivity of an object O using equal-width histograms [2] which model the distribution of valid values per predicate (its range). This range of values is divided into B equal-width classes O_c. The selectivity of an object O given predicate P can then be estimated as $\frac{h_c(P,O_c)}{|T_p|}$ (h_c = height of histogram class).

QPI – Query Pattern Index. QPI focuses on joined triple pattern selectivity estimation as it may well be that two triple patterns are not very selective taken separately, but highly selective in combination. Based on the ontology schema, the QPI features a list of valid joined triple patterns and their exact selectivities. In a nutshell, using domain and range information of the properties in the schema, the QPI pre-processor computes all possible joins/combinations between properties.

Hybrid Approach – Combining SEI and QPI. Finally, SEI and QPI are combined such that QPI runs before SEI to (1) rank the triple patterns within each 2-triple pattern join and to (2) sort the remaining, non-joined patterns to which QPI does not attach any selectivity among each other.

3 Preliminary Results

In the poster, we show a number of detailed figures which show the in-memory performance of plain ARQ, SEI, QPI, Hybrid, and Sesame[4] on the Lehigh University Benchmark [1]. We show that the hybrid optimization approach outperforms all other approaches on most queries (up to a factor of 390'000 times faster). We also show that the SEI-optimization is clearly outperformed by QPI wherever joined patterns are available. QPI, on the other hand, fails when there are less than two joined triple patterns. Thus, it is not surprising that the hybrid technique combining SEI and QPI leads to the best results.

References

1. Y. Guo, Z. Pan, and J. Heflin. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics*, 3(2–3):158–182, 2005.
2. G. Piatetsky-Shapiro and C. Connell. Accurate Estimation of the Number of Tuples Satisfying a Condition. In *SIGMOD*, pages 256–276, 1984.
3. P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access Path Selection in a Relational Database Management System. In *SIGMOD*, pages 23–34, 1979.

[3] We assume independence of the selectivities – which is inaccurate – but beneficial.

[4] `http://www.openrdf.org/`