A Cost-based Optimizer for SPARQL Queries

Edna Ruckhaus and María Esther Vidal and Eduardo Ruiz and Javier Sierra

Universidad Simón Bolívar, Caracas, Venezuela
{ruckhaus,mvidal,eruiz,jsierra}@ldc.usb.ve

1 Introduction

The cost of answering a query against an ontology is affected by at least three elements: the size of the ontology, the strategy followed to combine the data, and the order or plan in which data is processed. In the context of the Semantic Web, very large ontologies have been defined; therefore, techniques to identify efficient evaluation strategies are needed.

We propose cost-based optimization techniques for SPARQL queries. In our approach, ontologies are modeled as a deductive database. The extensional database is comprised of meta-level predicates that represent the information explicitly modeled by the ontology; for each RDFS built-in vocabulary term, we define a meta-level predicate (e.g., $subClassOf$). The intensional database corresponds to the deductive rules that implement the semantics of the vocabulary terms (e.g., the transitive properties of the $subClassOf$ term). Currently, we have developed the following techniques:

1. A hybrid cost model to estimate the cardinality and evaluation cost of the predicates that represent the ontology's extensional and intensional facts [1]. Extensional fact estimates are computed using traditional relational database cost models. Conversely, to estimate the cost and cardinality of data that do not exist a priori, which is the case of the intensional facts, sampling techniques are applied.
2. A twofold optimization strategy that combines cost-based optimization and Magic Sets techniques [1]. In the first stage, a dynamic programming-based algorithm is used to identify an ordering of patterns in the query that minimizes its estimated evaluation cost. In the second stage, Magic Sets techniques are used to push down query selections into the ontology representation. This strategy considers SPARQL Basic graph patterns and traverses a search space of left linear execution plans.
3. A randomized optimization strategy based on the Simulated Annealing algorithm. This algorithm explores execution plans of any shape (bushy trees); also, the cost model has been modified to consider in its formulas unbounded values. Thus, Group and Optional patterns are considered.

A similar approach was presented in [2]: cost-based optimization techniques are also implemented, but the proposed strategies may explore a reduced portion of the space of possible plans, and the identified solution could be costly. On the contrary, our techniques do not restrict the shape of the plans considered, and may successfully identify optimal execution plans.

2 Architecture

Our system is comprised of two main components: a query engine and an ontology manager. The query engine is composed of a query parser, a query optimizer and an execution engine. The ontology manager translates the ontologies into the deductive database representation and extracts the statistics that describe the ontologies: cost of inferring intensional facts, cardinality of extensional and intensional facts, and number of different values of each attribute. Once a SPARQL query is received, the patterns in the *WHERE* clause are translated into a conjunctive query, where each pattern corresponds to an extensional or intensional predicate. The conjunctive query is then passed on to the optimizer. The optimizer uses the statistics stored in the catalog to identify an efficient query execution plan. Next, the plan is given to the query engine, which evaluates it against the ontology.

3 Experimental Results

We conducted an experimental study to analyze the behavior of the above described twofold optimization technique. We considered basic pattern queries against synthetic ontologies and real-world ontologies Galen and EHR_RM [1]. We analyzed the predictive capability of the cost model and cost improvements from using the optimization strategy. The correlation for the real-world ontology Galen is 0.53, while for EHR_RM it is 0.43, i.e., the estimated cost and the actual cost are related. Additionally, we studied the benefits of the twofold optimization strategy. For the three ontologies studied we can observe that the cost of the Magic Sets optimal ordering falls in at least the 74th percentile, indicating that the cost of three quarters of all the plans is worse than the optimal ordering cost. We also reported the average ratio of the cost of the Magic Sets optimal ordering to the worst cost. For the synthetic ontologies, the average of the ratio of the optimal cost with respect to the worst-case is 45%; Galen and EHR_RM have averages of 15% and 32%, respectively. In addition, we have observed that the cost of the optimization process itself is negligible with respect to the evaluation cost.

From these results we can conclude that the implemented techniques improve the evaluation execution time of SPARQL queries. Currently, we are conducting an experimental study on Basic, Group and Optional query patterns. Additionally, we are comparing the performance of the optimal queries identified by our strategy on different RDF engines.

References

1. E. Ruckhaus, E. Ruiz, and M. Vidal. OnEQL: An Ontology Efficient Query Language Engine for the Semantic Web. In *Proceedings ALPSWS*, 2007.
2. M. Stoker, A. Seaborne, A. Bernstein, C. Keifer, and D. Reynolds. SPARQL Basic Graph Pattern Optimization Using Selectivity Estimation. In *WWW*, 2008.