Querying Key-Value Stores Under Simple Semantic Constraints: Rewriting and Parallelization

Olivier Rodriguez  
Université de Montpellier  
olivier.rodriguez@etu.umontpellier.fr

Corentin Colomier  
Université de Montpellier  
corentin.colomier@etu.umontpellier.fr

Cécile Rivière  
Université de Montpellier  
cecile.riviere@etu.umontpellier.fr

Reza Akbarinia  
INRIA Sophia Antipolis  
reza.akbarinia@inria.fr

Federico Ulliana  
Université de Montpellier  
federico.ulliana@lirmm.fr

Accessing data under semantic constraints

Popular paradigm, considered for example in ontology-based data integration. Usually setting is: DL or existential rules + relational or RDF data + conjunctive queries.

Demo contribution

A system for accessing Key-Value stores that
1. accounts for semantic constraints over keys
2. supports MongoDB queries akin to tree-pattern queries (without joins)
3. exploits query reformulation techniques for data access
4. parallelizes the computing of large rewriting sets over multiple threads

Query rewriting

Rewrite the edges of the tree-like query:

$(\forall$-rule) $k_1 \rightarrow k_2$ replace $k_2$ with $k_1$

$(\exists$-rule) $k_1 \rightarrow \exists k_2$ replace $k_2$ with $k_1,$ only on "existential leaves".

$3$-rules must be applied only on existential leaves to ensure soundness.

The tree view of JSON records

General form of a key constraint:

$(\forall$-rule) $k_1 \rightarrow k_2$ (inclusion between keys)

$(\exists$-rule) $k_1 \rightarrow \exists k_2$ (mandatory key)

$\sigma_1$ phone $\rightarrow$ contact  Every phone is a contact

$\sigma_2$ mail $\rightarrow$ contact  Every mail is a contact

$\sigma_3$ prof $\rightarrow$ director  If there is a professor then there is a director

$\sigma_4$ prof $\rightarrow$ faculty  A professor is a faculty member

Key constraints

Parallelization

Goal: parallelize the generation of the whole rewriting set.

Rewrite $Q: \{\text{faculty} : \{ \text{contact} : \{ \text{exists} : \text{true} \} \} \}$ with the rules $\sigma_1, \sigma_2, \sigma_4$

1) Enumerate the edges of the query and build an unranked decision tree where

- the level $j$ corresponds to the rewriting of the edge $j$ of the query ($1 \leq j \leq |Q|$)
- the degree of a node $d_j$ corresponds to the possible rewritings of the edge $j$

2) A leaf-code $\{ c_1, \ldots, c_n \} \in \{0,1, \ldots, |\Sigma|\}^{\mathbb{P}(\Sigma)}$ given $(d_1, \ldots, d_n)$ is the integer

$p_{c_1, \ldots, c_n} = c_1 \cdot d_1 + c_2 \cdot d_2 + \ldots + c_n \cdot d_n$ with $B_1 = 1$ and $B_{j+1} = d_{j+1} \cdot B_{j+1}$

3) Assign to each of $n$ rewritings an interval of queries to be generated $[N, \ldots, N+\lambda]$ with $\lambda = |\text{leaves} (\text{Rew}(Q, \Sigma))/n|$.