Accepted Manuscript

On the size of intermediate results in the federated processing of SPARQL BGPs

Jonas Halvorsen, Audun Stolpe

PII: S1570-8268(18)30027-1
DOI: https://doi.org/10.1016/j.websem.2018.06.001
Reference: WEBSEM 461

To appear in: *Web Semantics: Science, Services and Agents on the World Wide Web*

Received date : 28 March 2018
Accepted date : 18 June 2018

Please cite this article as: J. Halvorsen, A. Stolpe, On the size of intermediate results in the federated processing of SPARQL BGPs, *Web Semantics: Science, Services and Agents on the World Wide Web* (2018), https://doi.org/10.1016/j.websem.2018.06.001

On the size of intermediate results in the federated processing of SPARQL BGPs

Jonas Halvorsen[a,b,*], Audun Stolpe[a]

[a]*Norwegian Defence Research Establishment (FFI), Postboks 25, 2027 Kjeller, Norway*
[b]*Department of Informatics, University of Oslo, Norway*

Abstract

This paper is a foundational study in the semantics of federated query answering of SPARQL BGPs. Its specific concern is to explore how the size of intermediate results can be reduced without, from a logical point of view, altering the content of the final answer. The intended application is to reduce communication costs and local memory consumption in querying dynamic network topologies and highly distributed, share-nothing or sharded architectures. We define row-reducing and column-reducing operations that, if a SPARQL resultset is viewed as a table, reduces the number of rows and columns respectively. These operations are deliberately designed so that they do not anticipate the unfolding of the evaluation process, which is to say that they do not presuppose knowledge about the structure or content of data sources, or equivalently, that they do not require data to be exchange in order to make intermediate results smaller. In other words, the operations that are studied are based solely on the shape of evaluations trees and the distribution of variables within them. The paper culminates with a study of different compositions of the aforementioned reduction operators. We establish mathematically that our row- and column operators can be combined to form a single reduction operator that can be applied repeatedly without altering the semantics of the final result of the query answering process.

Keywords: Federated query processing, intermediate results, mimimization, blank nodes, sparql

1. Introduction

Federated SPARQL processing concerns the task of answering a a global query using the combined information from distinct sources. It involves breaking up a global query into a set of jointly exhaustive subqueries each of which is directed to a particular SPARQL endpoint before the results are returned to the federated query processor and combined into a correct answer to the initial global query, if one is to be had. That the exploitation and dissemination of Semantic Web data requires powerful federation is something of a truism, given the Web wide scope of names in RDF and the whole Linked Data philosophy.

This paper formalizes and investigates various optimizations that can be used to lighten the overall dataflow that this process consumes. More specifically, it is concerned with the question of how to reduce the size of intermediate result without compromising the semantics of the final answe of said global query.

Although the problem of keeping intermediate results small is of interest to both local and federated query execution, it is particularly pressing in the distributed case where the triples participating in a join may be stored on different servers. Such *cross-site* joins require network communication during join evaluation, that is, data has to be exchanged between servers in order to evaluate the join in question. Needless to say, this will claim bandwidth and CPU time proportionate to the amount of data that is exchanged, and as pointed out in [1] may easily grow with the overall data size to exceed the capacity of individual servers. Hence if the size of intermediate results is allowed to grow unconstrained, then in addition to any capacity issues with bandwidth and/or remote servers, it is likely that memory overflow problems will propagate back to the local thread of execution. Therefore, how *little* data one can send and keep in memory without sacrificing the precision and completeness of the final query answer should be a worthwhile question to address.

We approach this question by studying combinations of reduction operators, as we shall call them, in different order. Some of these operators are best regarded as part of the folklore, although we believe we offer at least one new one as well. However, the main contribution of the present paper is an integrated formal account of these operators that allows them to be studied in combination in a mathematically principled manner.

There are two kinds of reduction operators: operators that remove redundant rows and operators that remove redundant columns. We are after the conservative cores, so to speak, of intermediate results, by which we mean the smallest amount of data that needs to be retained to prevent information loss in the final query answer.

The first hurdle here is to clarify what it means for a federated SPARQL processor to lose information. Our take on this is to say that a federated SPARQL processor should return the same answer *set* (to make life a bit easier, we adopt the set semantics rather than the multitset semantics for SPARQL) as the one that would be returned were the query to be executed

*Corresponding author
Email addresses: jonas.halvorsen@ffi.no (Jonas Halvorsen), audun.stolpe@ffi.no (Audun Stolpe)