A Semantic Web middleware for Virtual Data Integration on the Web

Poster Abstract

Andreas Langegger, Wolfram Wöß, and Martin Blöchl

Institute of Applied Knowledge Processing
Johannes Kepler University Linz
Altenberger Straße 69, 4040 Linz, Austria
{ al | martin.bloechl | wolfram.woess } @jku.at

1 Motivation

One of the most promising fields for the application of Semantic Web technology is the integration of data from distributed sources. Although data integration is a mature field of research and many popular database systems have some support for legacy data and query federation, concepts introduced in the Semantic Web research community provide lots of new opportunities. A commonly used procedure currently is the consolidation of RDF data into a central RDF store. However, for many applications this simple merge of data is inapplicable, especially when data sources are very large or change frequently. In this contribution a system is presented which is capable of virtually integrating distributed, heterogenous data based on global SPARQL queries. While this *Semantic Web Integrator and Query Engine* (SemWIQ) was primarily designed for data sharing and scientific collaboration, it is regarded as a base technology useful for many other Semantic Web applications. At its core it uses a query federator which accepts SPARQL queries adhering to a virtual data set and dispatches sub-queries to all relevant known end-points. Those end-points usually wrap data stored in local information systems to RDF and provide SPARQL endpoints which are registered at the SemWIQ mediator host. The mediator is capable of executing join and union operations across different remote endpoints. The optimization of such distributed operations is a key factor concerning the performance of the overall system. In this poster session we would like to complement the work described in our conference paper and present on-going research regarding the optimization of distributed SPARQL queries.

2 System Overview

The architecture of SemWIQ basically applies the mediator-wrapper approach: there is a single central mediator service and multiple RDF wrappers which are locally attached to remote data sources. For relational database systems, which are the most popular information systems, the D2R-Server[1] has been used so far. The D2R approach is also an adequate basis for other wrappers, e.g. CSV/Excel files in local file systems, etc.

[1] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/, March '08

Setting up SemWIQ requires: (1) setting up a mediator host, (2) setting up wrappers, (3) specification of mappings from local concepts (i.e. database tables and attributes) to RDF Schema and OWL vocabularies for each data source, (4) eventually creation and publication of new vocabularies, and finally (5) registration of data sources at the mediator.

Concept-based data integration means, that all information in the global virtual graph is typed, i.e. entities are instances of some class. Based on this information, the mediator selects relevant data sources when the user executes a query. The query execution process is straight forward: parse, federate, optimize, and execute the final plan. A global query plan basically contains locally dispatched sub-plans which are connected by binary operators such as joins and unions.

3 Query Optimization and Future Work

First plan optimizations have been implemented by applying pre-defined static rules based on heuristics and plan equivalences. Rules may fire multiple times and they may also cause further rules to fire and transform the plan. The optimizer uses a RETE-based rule engine which provides good performance. For instance, the push-down of filters beyond binary operators can be described by a single static rule. Another example is the push-down of unary operators into local sub-plans, which always results in a better performance. For example, when filtering triples, it makes a significant difference whether to transfer all solutions to the mediator first and then apply the filter, or execute the filter as early as possible locally at the data source.

To further improve the optimizer it will become necessary to implement a cost model and use an IDP-based approach (*Iterative Dynamic Programming* is a poplar problem solving algorithm often used by database optimizers). Each wrapper will have to provide a special statistics graph which can be monitored by the mediator.

Future work may also include a visual browser for known vocabularies. Currently, creating a query requires some knowledge about the data which is available in the virtual graph. Additionally, new wrappers will be required to allow other information systems to be integrated.

Acknowledgements

This work is part of the *Grid-enabled Semantic Data Integration Middelware* (G-SDAM), which is supported by the *Austrian Grid Project*, funded by the Austrian BMBWK (*Federal Ministry for Education, Science and Culture*), contract GZ BMWF-10.220/0002-II/10/2007.