Workshop on Semantic Data Management A Summary Report

Reto Krummenacher
STI, University of Innsbruck, Austria
reto.krummenacher@sti2.at

Karl Aberer
LSIR, Ecole Politechnique Federale de Lausanne, Switzerland
karl.aberer@epfl.ch

Atanas Kiryakov
Ontotext AD, Bulgaria
naso@ontotext.com

Rajaraman Kanagasabai
Institute for Infocomm Research, Singapore
kanagasa@i2r.a-star.edu.sg

ABSTRACT

The International Workshop on Semantic Data Management (SemData) was held in Singapore co-located with the VLDB Conference 2010, with the goal of serving as a platform for the discussion and investigation of various aspects related to semantic databases and data management in the large. The workshop was a full-day event featuring two research sessions, one industry session and a panel discussion, and attracted over 25 attendees. This report summarizes the key topics presented, interesting ideas discussed and the new perspectives identified during the workshop.[1]

1. INTRODUCTION

The Semdata Workshop started off with a presentation of the organizing initiative (www.semdata.org) by one of its co-founders. The SemData initiative represents a series of events and activities that aim at facilitating the development and adaption of semantic data management concepts, standards, tools, benchmarks and best practices; or, in the words of the presenter, SemData offers a platform for researchers and industry to work on "making *teenage experiences* (great things and errors) more consistent and a trend in data management". The introduction to the workshop pointed out the particularities of RDF data management and the RDF paradigm, and recalled the open challenges and research questions in database technologies for semantic data: performance and scalability improvements, benchmarking, distribution (data partitioning, replication, and federation), and interoperability and integration with traditional database systems. Indeed, many database solutions for semantic data still run behind comparable non-semantic technologies. Although, it is indispensable that semantic repositories reach near performance parity with some of the best RDBMS solutions, they must not have to omit the advantages of higher query expressivity compared to basic key-value stores, or higher schema flexibility compared to the relational model. To launch the workshop on a high note, the introduction also accentuated the fast increase in popularity of semantic repositories in various vertical sectors, which clearly showcases the relevance and timeliness of the initiative and the workshop.

[1]The workshop was supported by the EC-FP7 IP projects LarKC (www.larkc.eu) and SOA4All (www.soa4all.eu).

2. RESEARCH PAPERS

The workshop featured four peer-reviewed research papers on SPARQL processing and query optimization, on RDF main memory storage and on structured indexing of RDF data.

In "SPARQL Query Answering on a Shared-nothing Architecture" a technique was proposed to do distributed and join-less RDF query answering based on query pattern-driven indexing. To this end, the authors proposed an extension to SPARQL that allows for specifying query patterns. These patterns were used to build query-specific indexes using MapReduce, which are later queried using a NoSQL store; i.e., the system only indexes the data that is actually needed. The evaluation indicated that, for a predefined query pattern, the proposed system offered very high query throughput and fast response times.

The paper "Optimizing SPARQL queries over the Web of Linked Data" addressed the challenge of optimizing SPARQL queries over the Web of Data, and proposed a two-phase approach. First, the queries

were analyzed before execution, and classes of data were discovered that do not contribute towards the provisioning of answers; such data is then prevented from being fetched. In a second step, the query execution is modeled as a context graph to prune further nodes. An implementation of this approach showed first benefits and at least theoretical performance improvements.

The contribution "SpiderStore: Exploiting Main Memory for Efficient RDF Graph Representation and Fast Querying" presented a novel storage concept that is capable of efficiently managing large RDF data sets, and that provides powerful and fast SPARQL processing facilities. The solution was to leverage the natural network-structure of RDF by using fast and random access to main memory; the graph is represented by a set of addresses (vertices) and bi-directional pointers (edges) in memory. The abandonment of additional mappings or meta-information, as used in most available repositories, led to a significant performance gain compared to other RDF stores, when it comes to querying and random access patterns.

The paper "Structure Index for RDF Data" elaborated on a novel data partitioning strategy, which leverages the structure of the underlying RDF data. The index was represented in form of a parameterized structure index called PIG that summarizes the structure of general graph-based data like RDF; the structure index was a graph too that simulates the schema. The authors managed to show, with a first benchmark against state-of-the-art techniques, that their structure-based approach for partitioning and query processing exhibited 7-8 times faster performance.

3. INDUSTRY POSITION PAPERS

As stated in the introduction, the application of RDF repositories is gaining momentum in various business settings that reach largely beyond the early adaptor sectors such as life sciences and eHealth. In order to provide a more concrete view onto the requirements of companies, businesses and public bodies, the workshop organizers invited selected industry representatives to talk about their concerns, customers and products.

Orri Erling, program manager and lead developer of Virtuoso, first reflected on the market possibilities of RDF and RDF databasing. Besides the aforementioned pharmaceutical and biomedical industry, telecoms, for example, have a natural use for the advantageous integration and interchange capabilities of the schema-last RDF model; telecoms are a patchwork of business compositions. The talk was continued with short presentations of various cutting edge developments in the scope of Virtuoso. Column-wise compressed storage subsystems allows for more than fourfold improvement in space efficiency and comparable query times between relational and RDF forms. A second part of the talk looked at how to synchronize RDF data sets across distance and at how to keep RDF extractions of local relational data up to date. The presentation concluded with an outlook at open topics: benchmarking, parallelism and reuse of intermediate results, online data integration, and finally improvements to the consumption layer (search and visualization).

The second speaker was Atanas Kiryakov, executive director of Ontotext AD. The presentation started off with a general discussion of how to interpret and conduct benchmarking of (RDF) databases. In particular when dealing with RDF data, it is relevant to understand whether a store relies on materialization or query time inference. While the former improves query-time responsiveness, the latter is much better in terms of load time. Hence, benchmarking must cover the loading and querying of data; i.e., the entire life-cycle. The talk continued with an introduction to OWLIM, its newest features (smooth invalidation, full-text search, consistency checking, replication cluster) and its application to the BBC World Cup 2010 Web site. BBC had decided to move from a relational database to RDF not only because of the available ontological models, but mainly due to the fact that their old solution become unmanageable and too slow. The BBC application was run on a replication cluster for supporting the query load of over one million requests a day. Again, distribution in RDF databasing is not a question of volume yet, but solely for speeding up data management and query answering through concurrency. As of today, OWLIM could host almost the entire Web of Data (25 billion statements as of September 2010) on a EUR10000.- server.

The third invited speaker was Jans Aasman, president and CEO of Franz Inc, who presented the most recent developments (stored procedures in for example Lisp or javascript) and applications of the AllegroGraph RDF graph database. Very recently Amdocs, a vendor that provides customer care, billing and order management systems, was releasing AIDA (Amdocs Intelligent Decision Automation) that runs on top of AllegroGraph. AIDA combines space-based architectures, semantic technology and a Bayesian belief network to bring together predictive analytics with customer experience management. The interest of Amdocs in the semantic technology solutions

of AllegroGraph are grounded in the fact that they required the processing and transformation of enormous amounts of data in real-time – which they could not support with available relational solutions. Moreover, semantics provides them with the flexibility and agility demanded by the fact that AIDA does not rely on predefined schemas. Other recent applications of AllegroGraph are TwitLogic, a semantic data aggregator, and Gruff. TwitLogic extracts entities from Twitter and converts those to an RDF stream, which is consumed by an AllegroGraph instance in order to execute time and location-based queries. Gruff is a free tool for visual query building and graph-based triple store browsing.

4. DISCUSSIONS

The workshop ended with an interactive afternoon full of lightning talks, panel and open discussion. We conclude this summary paper with a listing of the main questions addressed and some of the thoughts that were expressed.

How to get RDF widely used? RDF-vendors should join forces and put together a high-quality set of interoperable tools for the consumption of RDF (exploration, query building, search, visualization), simple coding procedures for the inclusion of RDF in Web sites, and last but not least some convincing benchmark scenarios to ease the selection of database features and engines. In this context, it becomes also clear that there need to be arguments for showing that the complexity decrease for users is much more impacting than the performance loss when moving to RDF. But how is such a benchmark done, one that shows the RDF integration capability? Indeed, time-to-solution is the relevant competitive characteristic of RDF, but it is very difficult to show. A starting point would be the quantification of requirements, and the identification of differentiating applications; e.g., data diversity at Web scale, integration of in-house data, schema-last.

What are RDF databases good for? Following up on the previous item, the conceptually strongest point of RDF is the graph nature of the structure and the voluntary schema requirement; schema can be added later. In particular when dealing with annotations and metadata, it is very natural to not have a schema pre-defined. This is one of the main distinctions between the relational and the RDF data model. Similarly, there is a weakening in requirements in terms of typing when moving from object-oriented models to RDF: OO is about strong typing, RDF is about weak typing. The benefits of RDF databases are hence the same as for others: it is about optimizing the management functionality and query processing for RDF. An RDF database is thus argued to be simply a tweak towards RDF, and there is per se no such thing as an RDF database. Applications require particular queries and functionality such as RDF graphs, reasoning and others; and hence a database responding to these needs becomes an RDF database.

Who cares about formal semantics and reasoning? Most applications that build on RDF are doing simple reasoning, mostly through backward chaining. When dealing with SQL most people use views, and such a concepts is not provided by SPARQL. The views functionality in the RDF context comes from reasoning. For many applications reasoning does not need to be very powerful, but then, what is enough or not? In the Web context reasoning is governed by RDFS and OWL, but should it be OWL? The specification of OWL2 with its new types of application specific profiles goes in the right direction. Indeed, profiles are very useful when it comes to optimizing the trade-off between expressiveness and complexity. Effectively, some standardization in terms of inference and reasoning would help to implement engines, as particular types of inferencing still requires particular, even ad hoc, implementations. Important to note as well, for many cases reasoning goes beyond the "standard languages RDFS and OWL; e.g., combining probabilistic reasoning and RDF is something that would certainly attract new applications. Similarly, there is still very little research on combining structured queries and full-text search.

Distribution and parallelisation To start clear, data is distributed – there is nothing to do about it. Not having a schema however is a challenging characteristics, when it comes to distributed data management, not even speaking about reasoning yet. Although distribution is in principle much easier with RDF, as data integration is easier with RDF, data partitioning is not all that obvious. The use of distributed memory, as stated previously, is not yet a must, but dataset size will grow and call for (expensive) distributed repositories, and the requirement for such solutions will arise, sooner or later. The principle driver for distribution is thus still not size but parallelization for acquiring more processing power – although more resources do not imply more scale. The applications will thus have to evaluate the approach: singleton versus distributed database, universal storage verses universal access. The current move, on a general basis, is rather from distribution to parallelized centralization.