Scalable algorithms for cloud-based Semantic Web data management

Ioana Manolescu and François Goasdoué
OAK team (Inria), BD / IASI teams (LRI)

May 15, 2012

Data exchange on the Web has started through simple HTML pages, which information extraction programs mined for information. Structured document formats like XML greatly facilitated data sharing, in particular enabling applications such as RSS topic-based subscriptions. More recently, RDF, the W3C's format for Semantic Web data is being considered as an interesting format for disseminating (publishing) rich structured data. RDF is structured in statements describing *properties* of *resources*, statements also called *triples* and denoted (s, p, o) where o is the value of the property p of the resource s. RDF can be used to encode simple atomic attributes, relationships between entities (such as s_1 *worksFor* s_2), schema information (e.g., *privateCompany* is a subclass of *company*) and typing (s_2 has type *privateCompany*). The RDF model is very tolerant to heterogeneity, since there is no notion of required properties, nor of allowed properties (in an exclusive sense). Furthermore, RDF handles naturally multiple, partially overlapping sources of information, based on the W3C's RDF Schema and XML Namespaces specifications. Merging different RDF databases allows both combining their data and chaining their schema statements to enable rich reasoning [2].

For all these reasons, RDF is currently favored for data sharing, as witnessed by the Linked Open Data initiative (also known as the LOD cloud, see `http://linkeddata.org`), and many other practical applications such as information sharing in social networks, or content management for news organizations. The database community has recently taken interest in RDF data management, notably a team from MIT [1] and one from Max-Planck Institüt, which has developed RDF-3X [11], the most scalable platform currently known. Thanks to the judicious use of database algorithms and optimizations, such systems considerably outperform commonly used RDF tools such as Jena [10] etc. These works, however, considered RDF as a "three-attribute large table", and thus ignored the impact of RDF schema, due to the reasoning it enables on the base data; moreover, the amount of data they can handle was limited to the capacity of a single machine. Our own previous work has considered efficient storage for RDF [6] and efficient mechanisms for pushing RDF querying *and reasoning* within relational database systems [5].

An interesting option to overcome the size challenge is to rely on massively distributed architectures and in particular on *cloud* platforms, as has been done in recent works [4, 8, 9, 12]. However, many challenges remain: [8] ignores schema and reasoning, many choices in [9] are based on heuristics and have not been throughly discussed, while [4, 12] have yet to demonstrate their scalability.

The proposed PhD research work consists of investigating *efficient algorithms for expressive and efficient management of RDF data in a cloud context*. We aim at a rich subset of the language, extending the fragment considered in [1, 8, 11] with the ability to model, query and update schemas, as well as the rich reasoning this entails, and with blank nodes, that are a form of incomplete information specific to RDF. On this language, the purpose of the PhD is to investigate *parallel and distributed algorithms for querying and updating RDF*. Parallelism and distribution are required in order to cope with very large volumes of RDF data. We will consider a parallel infrastructure such as offered by Hadoop [7], and extend over works such as [12, 9] by optimizing for a trade-off between on one hand, the costs associated to *querying* RDF in such a distributed platform, and on the other hand the costs for *reasoning* over data and knowledge distributed across the store. Two sub-problems which can be tackled first are:

1. Adaptive partitioned and redundant storage: a large volumes of small RDF triples needs to be split across several machines. The partitions thus obtained may be redundant, i.e., some data may be

stored multiple times, for more efficiency. Of particular interest is the placement of schema triples with respect to data triples on which they may allow reasoning: communication across nodes in a distributed platform may be expensive, thus computations entailed by the management of RDF data must be performed within a single node whenever possible. Current partitioning schemes [8] handle distribution at the granularity of individual properties, while we envision *query molecules*, or small, commonly-occuring query fragments, as an interesting unit of distribution. Furthermore, as the RDF data and schema change, and/or as the querying patterns of this data vary, the data layout may change with the storage fragments determined by our query molecules growing, shrinking or moving across the partitions.

2. Efficient evaluation of conjunctive queries with reasoning: considering (to start with) the conjunctive core of the SPARQL RDF query language, in order to compute complete query answers, one also needs to take into account implicit data (due to the presence of a schema). This can be done either by saturating the database (that is, adding explicitly all the implicit triples next to the explicit ones), or by reformulating a conjunctive query into a union of conjunctive queries, such that evaluating this union over the explicit data only, yields the complete query results. We have started to investigate trade-offs between these methods in a centralized setting in [5]. However, a distributed setting significantly challenges these algorithms; for saturation, recursive query answering techniques based on Map/Reduce (e.g., [3]) are a starting point, whereas reformulation will require distributed query optimization techniques tailored for the specific shapes of queries resulting from reformulation.

Work on cloud-based data management has started in Oak within our participation to the KIC EIT ICT Labs activities "Europa" (on cloud-based data management) and "DataBridges" (on data integration techniques for digital cities, in particular we worked on RDF processing). We are now preparing renewal proposals of these activities for 2013. An Inria engineer (ADT DistribWeb) has arrived in the team in October 2011 and will be helping us in our cloud-based platform development.

References

[1] Daniel J. Abadi, Adam Marcus, Samuel R. Madden, and Kate Hollenbach. Scalable semantic web data management using vertical partitioning. In *VLDB*, 2007.

[2] Serge Abiteboul, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset, and Pierre Senellart. *Web Data Management*. Cambridge University Press, 2012.

[3] Foto N. Afrati, Vinayak R. Borkar, Michael J. Carey, Neoklis Polyzotis, and Jeffrey D. Ullman. Map-reduce extensions and recursive queries. In *EDBT*, pages 1–8, 2011.

[4] Francesca Bugiotti, François Goasdoué, Zoi Kaoudi, and Ioana Manolescu. RDF Data Management in the Amazon Cloud. In *Workshop on Data analytics in the Cloud (DanaC 2012)*, 2012.

[5] François Goasdoué, Ioana Manolescu, and Alexandra Roatis. Getting more rdf support from relational databases (poster paper). In *WWW Conference*, 2012.

[6] François Goasdoué, Konstantinos Karanasos, Julien Leblay, and Ioana Manolescu. View Selection in Semantic Web Databases. *Proceedings of the VLDB Endowment (PVLDB)*, 5(2), October 2011.

[7] Apache Hadoop. Available at `hadoop.apache.org`.

[8] Jiewen Huang, Daniel J. Abadi, and Kun Ren. Scalable SPARQL querying of large RDF graphs. *PVLDB*, 4(11):1123–1134, 2011.

[9] M. Husain, J. McGlothlin, M.M. Masud, L. Khan, and B.M. Thuraisingham. Heuristics-based query processing for large RDF graphs using cloud computing. *IEEE Transactions on Knowledge and Data Engineering*, 23(9):1312 –1327, sept. 2011.

[10] Apache Jena. Available at `jena.apache.org`.

[11] Thomas Neumann and Gerhard Weikum. The RDF-3X engine for scalable management of RDF data. *VLDB J.*, 19(1):91–113, 2010.

[12] Nikolaos Papailiou, Ioannis Konstantinou, Dimitrios Tsoumakos, and Nectarios Koziris. H2RDF: Adaptive Query Processing on RDF Data in the Cloud (demo). In *WWW Conference*, 2012.