Chapter 16
On the Efficiency of Querying and Storing RDF Documents

Maria-Esther Vidal
Universidad Simón Bolívar, Venezuela

Amadís Martínez
Universidad Simón Bolívar &Universidad de Carabobo, Venezuela

Edna Ruckhaus
Universidad Simón Bolívar, Venezuela

Tomas Lampo
University of Maryland, USA

Javier Sierra
Universidad Simón Bolívar, Venezuela

ABSTRACT

In the context of the Semantic Web, different approaches have been defined to represent RDF documents, and the selected representation affects storage and time complexity of the RDF data recovery and query processing tasks. This chapter addresses the problem of efficiently querying and storing RDF documents, and presents an alternative representation of RDF data, Bhyper, which is based on hypergraphs. Additionally, access and optimization techniques to efficiently execute queries with low cost, are defined on top of this hypergraph based representation. The chapter's authors have empirically studied the performance of the Bhyper based techniques, and their experimental results show that the proposed hypergraph based formalization reduces the RDF data access time as well as the space needed to store the Bhyper structures, while the query execution time of state-the-of-art RDF engines can be sped up by up to two orders of magnitude.

DOI: 10.4018/978-1-61350-053-8.ch016

INTRODUCTION

Emerging infrastructures such as the Semantic Web, the Semantic Grid, Service Oriented architectures and the Cloud of Linked Data support on-line access to a wealth of ontologies, data sources and Web services. Ontologies play an important role in these infrastructures, and provide the basis for the definition of concepts and relationships that make the recovery and integration of Web data and resources possible. Particularly, in the context of the Cloud of Linked Data, a large number of diverse datasets have become available, and an exponential growth has occurred during the last years. In October 2007, datasets consisted of over 2 billion RDF triples, which were interlinked by over 2 million RDF links. By May 2009 this had grown to 4.2 billion RDF triples interlinked by around 142 million RDF links. At the time this chapter was written, there were 13,112,409,691 triples in the Cloud of Linked Data; datasets can be on medical publications, airport data, drugs, diseases, and clinical trials, among others.

Furthermore, the number of available Web services has rapidly increased during the last few years. For example, the molecular biology databases collection currently includes 1,078 databases (Galperin, 2008) which is 110 more than the previous year (Galperin, 2007). Tools and services as well as the number of instances published by these resources follow a similar progression (Benson, 2007). In addition, thanks to this wealth, users rely more on various digital tasks such as data retrieval from public data sources or from the Cloud of Linked Data, as well as data analysis with Web tools or services organized in complex workflows. Thus, Web architectures need to be tailored for the provision of efficient storage structures and the processing of large number of resources and instances, in order to scale up to user requests.

In the context of the Semantic Web, several query engines have been developed to access RDF documents efficiently (e.g., AllegroGraph; Harth et al., 2007; Ianni et al., 2009; JENA; JENATDB; Neumann & Weikum, 2008; Wielemaker, 2005). The majority of these approaches have developed techniques to generate evaluation plans, and execution engines where these plans can be executed in a way that the processing time is reduced (e.g., AllegroGraph; Neumann & Weikum, 2008; Lampo et al., 2009; Vidal et al., 2010). Additionally, some of these approaches have implemented structures to efficiently store and access RDF data. Tuple Database or TDB (JENATDB) is a persistent graph storage layer for Jena. TDB works with the Jena SPARQL query engine (ARQ) to support SPARQL together with a number of extensions (e.g., property functions, aggregates, arbitrary length property paths).

YARS2 (Yet Another RDF Store, Version 2) (Harth et al., 2007) is a repository for queries against an indexed federation of RDF documents; three types of in-memory indices are used to scan keywords, perform atomic operations on RDF documents, and speed up combinations of patterns or values. RDF-3X (Neumann & Weikum, 2008) focuses on an index system, and its optimization techniques were developed to explore the space of plans that benefit from these index structures. Hexastore (Weiss et al., 2008) is a main memory indexing technique that uses the triple nature of RDF as an asset. RDF data is also indexed in six possible ways, one for each possible triple pattern permutation. Finally, secondary-memory index-based representations for large RDF datasets are presented in (e.g., Fletcher & Beck, 2009; McGlothlin & Khan, 2009; Weiss & Bernstein, 2009).

All these approaches may reduce the execution time of RDF queries; however, for some queries, the solution identified can be far from optimal. For instance, as we will show in this chapter, some queries can be reordered and grouped into small-sized star-shaped groups, and the execution time can be orders of magnitude less than the execution time of the original query. However, because these approaches are not tailored to identify this type of plans or to use their storage structure properties to