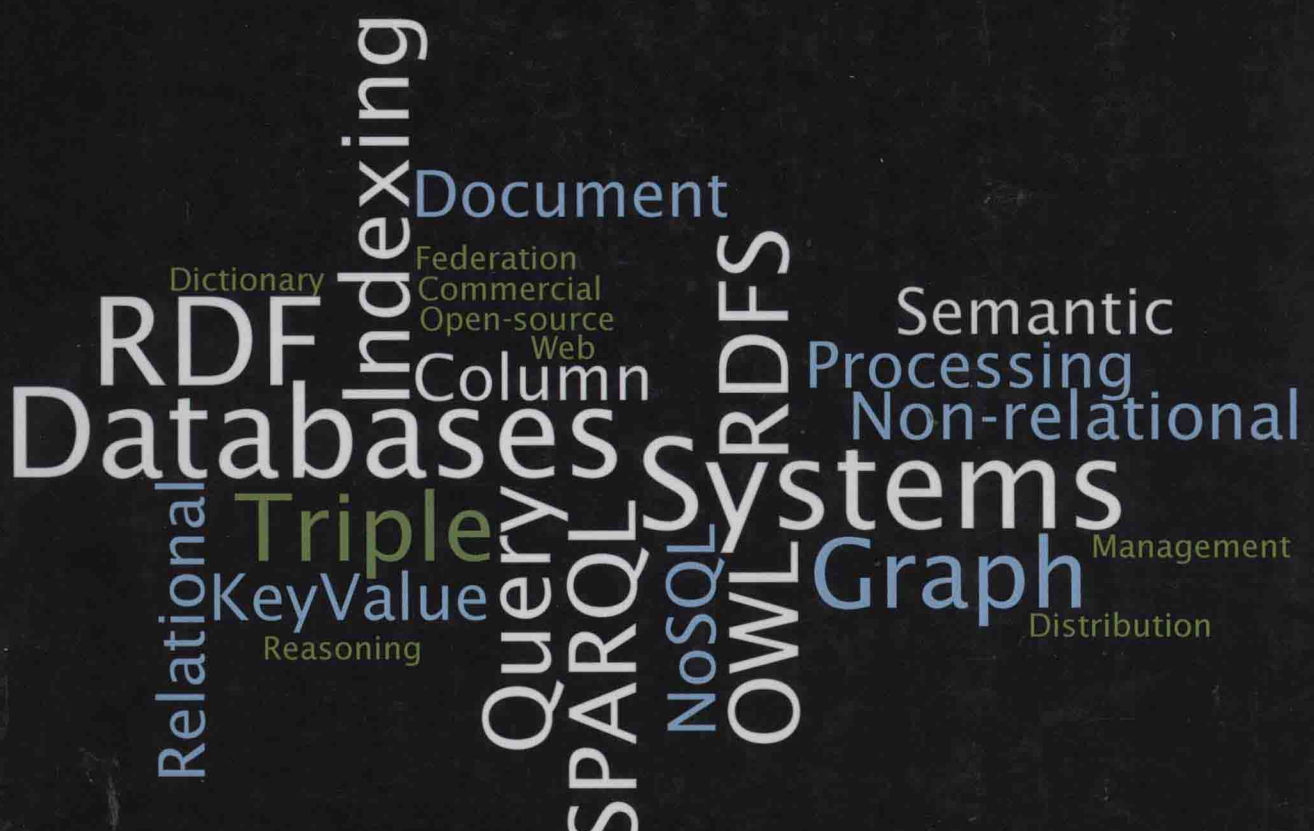
RDF Database Systems

Triples Storage and SPARQL Query Processing

Olivier Curé and Guillaume Blin



MK
MORGAN KAUFMANN

RDF Database Systems
TRIPLES STORAGE AND SPARQL QUERY PROCESSING

Edited by

OLIVIER CURÉ
Associate Professor of Computer Science,
Université Paris Est Marne la Vallée and
Université Pierre et Marie Curie, France

GUILLAUME BLIN
Professor of Computer Science,
Université de Bordeaux, France

ELSEVIER

Amsterdam • Boston • Heidelberg • London
New York • Oxford • Paris • San Diego
San Francisco • Singapore • Sydney • Tokyo
Morgan Kaufmann is an Imprint of Elsevier

MK
MORGAN KAUFMANN

Executive Editor: Steve Elliot
Editorial Project Manager: Kaitlin Herbert
Project Manager: Anusha Sambamoorthy
Designer: Mark Rogers

Morgan Kaufmann is an imprint of Elsevier
225 Wyman Street, Waltham, MA 02451, USA

First edition 2015

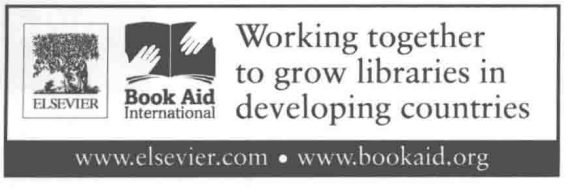
Library of Congress Cataloging-in-Publication Data

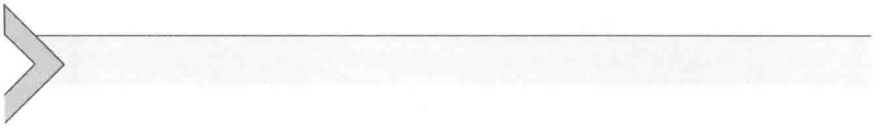
Curé, Olivier.
RDF database systems : triples storage and SPARQL query processing / by Olivier Curé, Guillaume Blin. — First edition.
pages cm
Includes index.
ISBN 978-0-12-799957-9 (paperback)
1. Database management. 2. RDF (Document markup language) 3. Query languages (Computer science) 4. Querying (Computer science) I. Blin, Guillaume. II. Title.
QA76.9.D3C858 2015
005.74 — dc23

2014034632

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN: 978-0-12-799957-9





RDF Database Systems
TRIPLES STORAGE AND SPARQL QUERY PROCESSING

PREFACE

In 1999, the *World Wide Web Consortium* (W3C) published a first recommendation on the *Resource Description Framework* (RDF) language. Since its inception, this technology is considered the cornerstone of the *Semantic Web* and *Web of Data* movements. Its goal is to enable the description of resources that are uniquely identified by *uniform resource identifiers* (URIs), for example, `http://booksite.mkp.com`. Recently, RDF has gained a lot of attention, and as a result an increasing number of data sets are now being represented with this language. With this popularity came the need to efficiently store, query and reason over large amounts of RDF data. This has obviously motivated some research work on the design of adapted and efficient data management systems. These systems, frequently referred to as *RDF stores*, *triple stores*, or *RDF data management systems*, must handle a data model that takes the form of a directed labeled graph where nodes are URIs or literals (e.g., character strings) and edges are URIs. This data model is quite different from the currently popular relational model encountered in a *relational database management system* (RDBMS). But RDF stores have other features that make their design a challenging task. In the following, we present four major features that differentiate RDF stores with other data management systems.

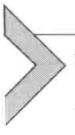
The first feature is related to the data model of RDF, which is composed of triples corresponding to a subject, a property, and an object, where the subject takes an object as a value for a property. A set of triples is characterized by the omnipresence of URIs at the three positions of RDF statements, although over forms can also be used, e.g., literals at the object position. The fact that a given URI can be repeated multiple times in a data set and that URIs are generally long strings of characters raises some memory footprint issues. Therefore, the use of efficient dictionaries—encoding URIs with identifiers that are used for the representation of triples—is crucial in triple stores, while in relational models a suitable database schema minimizes such data redundancy.

The second feature is that a data management system requires a language to query the data it contains. For this purpose, the Semantic Web group at the W3C has published the *SPARQL Protocol and RDF Query Language* (SPARQL) recommendation; one of its main aspects is to support a query language. Although some clauses such as `SELECT` and `WHERE` make it look like the popular *Structured Query Language* (SQL) language of RDBMS, SPARQL is based on the notion of triples and not on relations. Answering a SPARQL query implies some pattern-matching mechanisms between the triples of the query and the data set. This requires research in the field of query processing and optimization that goes beyond the state of the art of the relational model.

The third feature concerns vocabularies associated to RDF data sets. The most predominant ones are *RDF Schema* (RDFS) and *Web Ontology Language* (OWL), both of which enable the ability to reason over RDF data—that is, they enable the discovery of implicit data from explicitly represented ones.

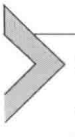
The final feature is that RDF data is concerned with the *Big Data* phenomenon, which induces the distribution of data sets over clusters of machines. The graph nature of the RDF data model makes the distribution a challenging task that has to be understood in an inference-enabled context.

This book aims at presenting the fundamentals on all these features by introducing both the main concepts and techniques, as well as the principal research problems that emerge in the ecosystem of RDF database management systems. The presentations are based on the study of a large number of academic and commercial products. It is important to understand that the fulfillment of the Semantic Web vision largely depends on the emergence of robust, efficient, and feature-complete RDF stores.



WHO SHOULD READ THIS BOOK

The book targets a wide range of readers. We consider that technology practitioners, including developers, project leaders, and database professionals, will find a comprehensive survey of existing commercial and open-source systems that will help them to select a system in a given application implementation context. We also consider that students and teachers could use this book for lectures in Semantic Web and knowledge representation, as well as for advanced database courses. In fact, the idea of providing a textbook for our master's students at the University of Paris Est was one of the motivations for starting this work. Finally, researchers in both the Semantic Web and database fields will find in this book a comprehensive and comparative overview of an active research domain. Indeed, we were in need for such an overview when we started our work on the **roStore** and **WaterFowl** RDF stores.



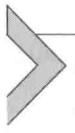
ORGANIZATION OF THE BOOK

This book is organized in two parts composed of short chapters to support an efficient and easy reading. The first part defines the scope of the book but also motivates and introduces the importance of efficiently handling RDF data. To make the book self-contained, it also provides background knowledge on both database management systems and Semantic Web technologies.

With these notions understood, readers can delve into the second part of the book, which investigates an important number of existing systems based on the criteria we have defined: the definition of an RDF dictionary, backend storage of RDF triples, indexation, query processing, reasoning, distribution of workloads, and query federation. We dedicate one chapter per criteria for easier reading of the book. With this structure, for example, a reader can learn about a given aspect of RDF stores in Chapter 4 on RDF dictionaries without searching through the complete book. This approach imposes that the different components of important systems are detailed in different chapters—for

example, the *RDF-3X* system is discussed in Chapters 4–6. At the end of each chapter we provide a summary list of the main notions studied.

Finally, the book concludes and anticipates on future trends in the domain of RDF storage and the Semantic Web.



GUIDELINES FOR USING THIS BOOK

We identify three distinct readings of this book that are mainly motivated by a reader's expertise in either database management or Semantic Web technologies. Of course, readers not confident about their knowledge on any of these domains should read both background knowledge chapters—that is, Chapters 2 and 3.

In the case of a reader with database expertise, Chapter 2 can be skipped, but we warn that several different forms of database management systems are studied. For example, most readers with a database profile may already master all notions considered in the section dedicated to the relational model, but it may not be the case on the sections addressing *NoSQL* and novel extensions of the relational model.

If a reader's profile corresponds to a Semantic Web literate, then reading Chapter 3 may be optional because it spends most of its content presenting RDF, SPARQL, and ontology languages such as RDFS and OWL, as well as associated reasoning services.

The second part of the book is its cornerstone. All readers should be interested in reading its content and learn about the different approaches to address selected dimensions.

IT project managers and developers will acquire knowledge on the main characteristics of existing, commercial, or open-source systems, while researchers will learn about the latest and most influential systems with references to publications.

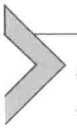
CONVENTIONS USED IN THIS BOOK

This book contains the following typographical conventions:

- **Bold** highlights the first occurrence of a software, system, or company name within a chapter.
- `Code font` is used for queries (e.g., expressed in SPARQL or SQL) and URIs, and can also be found within paragraphs to refer to query or program elements.

SUPPLEMENTAL MATERIALS

Additional materials for this book can be found at http://igm.mlv.fr/~ocure/rdfstores/



ACKNOWLEDGMENTS

We would like to thank our technical reviewer, Dean Allemang, special thanks go to our editor: Andrea Dierna, Kaitlin Herbert, and Anusha Sambamoorthy. We are also grateful to David Celestin Faye for participating in the design of roStore.

A very special thanks go to our families for their support: Virginie, Noé, Jeanne, Axelle, and Charlie.

CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER ONE

Introduction

If you have this book in your hands, we guess you are interested in database management systems in general and more precisely those handling *Resource Description Framework* (RDF) as a data representation model. We believe it's the right time to study such systems because they are getting more and more attention in industry with communities of Web developers and *information technology* (IT) experts who are designing innovative applications, in universities and engineering schools with introductory and advanced courses, and in both academia and industry research to design and implement novel approaches to manage large RDF data sets. We can identify several reasons that are motivating this enthusiasm.

In this introductory chapter, we will concentrate on two really important aspects. An obvious one is the role played by RDF in the emergence of the *Web of Data* and the *Semantic Web*—both are extensions of the original *World Wide Web*. The second one corresponds to the impact of this data model in the *Big Data* phenomenon. Based on the presentation of these motivations, we will introduce the main characteristics of an RDF data management system, and present some of its most interesting features that support the comparison of existing systems.

1.1 BIG DATA

Big Data is much more than a buzzword. It can be considered as a concrete phenomenon that is attracting all kinds of companies facing strategic and decisional issues, as well as scientific fields interested in understanding complex observations that may lead to important discoveries. The *National Institute of Standards and Technologies* (NIST) has proposed a widely adopted definition referring to the data deluge: "a digital data volume, velocity and/or variety that enable novel approaches to frontier questions previously inaccessible or impractical using current or conventional methods; and/or exceed the capacity or capability of current or conventional methods and systems" (NIST Big Data, 2013). Most Big Data definitions integrate this aspect of the three V's: volume, velocity, and variety (which is sometimes extended with a fourth V for veracity).

Volume implies that the sizes of data being produced cannot be stored and/or processed using a single machine, but require a distribution over a cluster of machines. The challenges are, for example, to enable the loading and processing of exabytes (i.e., 10^3 petabytes or 10^6 terabytes) of data while we are currently used to data loads in the range of at most terabytes.

Velocity implies that data may be produced at a throughput that cannot be handled by current methods. Solutions, such as relaxing transaction properties, storing incoming data on several servers, or using novel storage approaches to prevent input/output latencies, are being proposed and can even be combined to address this issue. Nevertheless, they generally come with limitations and drawbacks, which are detailed in Chapter 2.

Variety concerns the format (e.g., **Microsoft**'s *Excel* [XLS], *eXtended Markup Language* [XML], *comma-separated value* [CSV], or RDF) and structure conditions of the data. Three main conditions exist: structured, semi-structured, and unstructured. Structured data implies a strict representation where data is organized in entities, and then similar entities are grouped together and are described with the same set of attributes, such as an identifier, price, brand, or color. This information is stored in an associated schema that provides a type to each attribute—for example, a price is a numerical value. The data organization of a *relational database management system* (RDBMS) is reminiscent of this approach. The notion of semi-structured data (i.e., self-described) also adopts an entity-centered organization but introduces some flexibility. For example, entities of a given group may not have the same set of attributes, attribute order is generally not important in the description of an entity, and an attribute may have different types in different entity groups. Common and popular examples are XML, *JavaScript Object Notation* (JSON), and RDF. Finally, unstructured data is characterized by providing very little information on the type of data it contains and the set of formatting rules it follows. Intuitively, text, image, sound, and video documents belong to this category.

Veracity concerns the accuracy and noise of the data being captured, processed, and stored. For example, considering data acquired, one can ask if it's relevant to a particular domain of interest, if it's accurate enough to support decision making, or if the noise associated to that data can be efficiently removed. Therefore, data quality methods may be required to identify and clean "dirty" data, a task that in the context of the other three V's may be considered one of the greatest challenges of the data deluge. Surprisingly, this dimension is the one that has attracted the least attention from Big Data actors.

The emergence of Big Data is tightly related to the increasing adoption of the Internet and the Web. In fact, the Internet proposes an infrastructure to capture and transport large volumes of data, while the Web, and more precisely its 2.0 version, has brought facilities to produce information from the general public. This happens through interactions with personal blogs (e.g., supported by **WordPress**), wikis (e.g., **Wikipedia**), online social networks (e.g., **Facebook**), and microblogging (e.g., **Twitter**), and also through logs of activities on the most frequently used search engines (e.g., **Google**, **Bing**, or **Yahoo!**) where an important amount of data is produced every day. Among the most stunning recent values, we can highlight that Facebook announced that, by the beginning of 2014, it is recording 600 terabytes of data each day in its 300 petabytes data warehouse and an average of around 6,000 tweets are stored at Twitter per second with a record of 143,199 tweets on August 3, 2013.

The *Internet of Things* (IoT) is another contributor to the Big Data ecosystem, which is just in its infancy but will certainly become a major data provider. This Internet branch is mainly concerned with *machine-to-machine* (M2M) communications that are evolving on a Web environment using Web standards such as *Uniform Resource Identifiers* (URIs), *HyperText Transfer Protocol* (HTTP), and *representational state transfer* (REST). It focuses on the devices and sensors that are present in our daily lives, and can belong to either the industrial sector or the consumer market. These active devices may correspond but are not limited to smartphones, *radio-frequency identification device* (RFID) tagged objects, wireless sensor networks, or ambient devices.

IoT enables the collection of temporospatial information—that is, regrouping temporal as well as spatial aspects. In 2009, considered an early year of IoT, Jeff Jonas in his blog (Jonas, 2009) was already announcing that 600 billion geospatially tagged transactions were generated per day in North America. This ability to produce enormous volumes of data at a high throughput is already a data management challenge that will expand in the coming years. To consider its evolution, a survey conducted by **Cisco** (Cisco, 2011) emphasized that from 1.84 connected devices per person in 2010, we will reach 6.58 in 2020, or approximately 50 billion devices. Almost all of these devices will produce massive amounts of data on a daily basis.

As a market phenomenon, Big Data is not supervised by any consortium or organism. Therefore, there is a total freedom about the format of generated, stored, queried, and manipulated data. Nevertheless, best practices of the major industrial and open-source actors bring forward some popular formats such as XLS, CSV, XML, JSON, and RDF. The main advantages of JSON are its simplicity, flexibility (it's schemaless), and native processing support for most Web applications due to a tight integration with the *JavaScript* programming language. But RDF is not without assets. For example, as a semi-structured data model, RDF data sets can be described with expressive schema languages, such as *RDF Schema* (RDFS) or *Web Ontology Language* (OWL), and can be linked to other documents present on the Web, forming the *Linked Data* movement.

With the emergence of Linked Data, a pattern for hyperlinking machine-readable data sets that extensively uses RDF, URIs, and HTTP, we can consider that more and more data will be directly produced in or transformed into RDF. In 2013, the *linked open data* (LOD), a set of RDF data produced from open data sources, is considered to contain over 50 billion triples on domains as diverse as medicine, culture, and science, just to name a few. Two other major sources of RDF data are building up with the *RDF in attributes* (RDFa) standard, where attributes are to be understood in an (X)HTML context, and the **Schema.org** initiative, which is supported by Google, Yahoo!, Bing, and **Yandex** (the largest search engine in Russia). The incentive of being well referenced in these search engines already motivates all kinds of Web contributors (i.e., companies, organizations, etc.) to annotate their web page content with descriptions that can be transformed

into RDF data. In the next section, we will present some original functionalities that can be developed with Linked Data, such as querying and reasoning at the scale of the Web.

As a conclusion on Big Data, the direct impact of the original three V's is the calling for new types of database management systems. Specifically, those that will be able to handle rapidly incoming, heterogeneous, and very large data sets. Among others, a major advantage of these systems will be to support novel, more efficient data integration mechanisms. In terms of features expected from these systems, Franklin and colleagues (2005) were the first to propose a new paradigm. Their *DataSpace Support Platforms (DSSP)* are characterized by a pay-as-you-go approach for the integration and querying of data. Basically, this paradigm is addressing most of the issues of Big Data. Later, in Dong and Halevy (2007), more details on the possible indexing methods to use in data spaces were presented. Although not mentioning the term *RDF*, the authors presented a data model based on triples that matches the kind of systems this book focuses on and that are considered in Part 2 of this book.

1.2 WEB OF DATA AND THE SEMANTIC WEB

The Web, as a global information space, is evolving from linking documents only, to linking both documents and data. This is a major (r)evolution that is already supporting the design of innovative applications. This extension of the Web is referred to as the *Web of Data* and enables the access to and sharing of information in ways that are much more efficient and open than previous solutions. This efficiency is due to the exploitation of the infrastructure of the Web by allowing links between distributed data sources. Three major technologies form the cornerstone of this emerging Web: URIs, HTTP, and RDF. The first two have been central to the development of the Web since its inception and respectively address the issues of identifying Web resources (e.g., web pages) and supporting data communication for the Web. The latter provides a data representation model, and the management of such data is the main topic of this book.

The term *semantics* is getting more and more attention in the IT industry as well as in the open-source ecosystem. It basically amounts to providing some solutions for computers to automatically interpret the information present in documents. The interpretation mechanism is usually supported by annotating this information with vocabularies the elements of which are given a well-defined meaning with a logical formalism. The logical approach enables some dedicated reasoners to perform some inferences. Of course, the Web, and in particular the Web of Data, is an important document provider; in that context, we then talk about a Semantic Web consisting of a set of technologies that are supporting this whole process. In Berners-Lee et al. (2001), the Semantic Web is defined as "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (p. 1). This emphasizes that there is no rupture between a previous non-semantic Web and a semantic one.