



Danilo Moret Rodrigues

Distributed RDF Graph Keyword Search

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-Graduação em Informática of the Departamento de Informática, PUC-Rio as partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Karin Koogan Breitman

Rio de Janeiro

March 2013



Danilo Moret Rodrigues

**Distributed RDF Graph Keyword
Search**

Dissertation presented to the Programa de Pós-Graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio, as partial fulfillment of the requirements for the degree of Mestre.

Prof. Karin Koogan Breitman

Advisor

Departamento de Informática – PUC-Rio

Prof. Marco Antônio Casanova

Departamento de Informática - PUC-Rio

Prof. José Viterbo Filho

UFF

Prof. Antônio Luz Furtado

Departamento de Informática - PUC-Rio

Prof. José Eugenio Leal

Coordinator of the Centro Técnico Científico –
PUC-Rio

Rio de Janeiro, March 25th, 2013

All rights reserved

Danilo Moret Rodrigues

Graduated in Telecommunications Engineering from Universidade Federal Fluminense - UFF in 2005.

Bibliographic data

Rodrigues, Danilo Moret

Distributed RDF Graph Keyword Search / Danilo Moret ; advisor: Karin Koogan Breitman. – 2013.

66 f. : il. ; 30 cm

Dissertação (Mestrado em Informática)–Pontificia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2013.

Inclui bibliografia

1. Informática – Dissertações. 2. Redes de Computadores e Sistemas Distribuídos. 3. Computação em Nuvem. 4. Linked Data. 5. Computação Distribuída. I. Breitman, Karin. II. Pontificia Universidade Católica do Rio de Janeiro. Departamento de Matemática. III. Busca distribuída em grafo RDF por palavra-chave.

CDD: 004

Acknowledgements

To Graça, Karin and Renata.

Abstract

Moret, Danilo; Breitman, Karin. **Distributed RDF Graph Keyword Search**. Rio de Janeiro 2013. 66pp. MSc Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The goal of this dissertation is to improve RDF keyword search. We propose a scalable approach, based on a tensor representation that allows for distributed storage, and thus the use of parallel techniques to speed up the search over large linked data sets, in particular those published as Linked Data. An unprecedented amount of information is becoming available following the principles of Linked Data, forming what is called the Web of Data. This information, typically codified as RDF subject-predicate-object triples, is commonly abstracted as a graph which subjects and objects are nodes, and predicates are edges connecting them. As a consequence of the widespread adoption of search engines on the World Wide Web, users are familiar with keyword search. For RDF graphs, however, extracting a coherent subset of data graphs to enrich search results is a time consuming and expensive task, and it is expected to be executed on-the-fly at user prompt. The dissertation's goal is to handle this problem. A recent proposal has been made to index RDF graphs as a sparse matrix with the pre-computed information necessary for faster retrieval of sub-graphs, and the use of tensor-based queries over the sparse matrix. The tensor approach can leverage modern distributed computing techniques, e.g., non-relational database sharding and the MapReduce model. In this dissertation, we propose a design and explore the viability of the tensor-based approach to build a distributed datastore and speed up keyword search with a parallel approach.

Keywords

Linked Data, MapReduce, Cloud Computing, Keyword Search

Resumo

Moret, Danilo; Breitman, Karin. **Busca distribuída em grafo RDF por palavra-chave**. Rio de Janeiro 2013. 66pp. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O objetivo desta dissertação é melhorar a busca por palavra-chave em formato RDF. Propomos uma abordagem escalável, baseada numa representação tensorial, que permite o armazenamento distribuído e, como consequência, o uso de técnicas de paralelismo para agilizar a busca sobre grandes bases de RDF, em particular, as publicadas como Linked Data. Um volume sem precedentes de informação está sendo disponibilizado seguindo os princípios de Linked Data, formando o que chamamos de Web of Data. Esta informação, tipicamente codificada como triplas RDF, costuma ser representada como um grafo, onde sujeitos e objetos são vértices, e predicados são arestas ligando os vértices. Em consequência da ampla adoção de mecanismos de busca na World Wide Web, usuários estão familiarizados com a busca por palavra-chave. No caso de grafos RDF, no entanto, a extração de uma partição coerente de grafos para enriquecer os resultados da busca é uma tarefa cara, demorada, e cuja expectativa do usuário é de que seja executada em tempo real. Este trabalho tem como objetivo o tratamento deste problema. Parte de uma solução proposta recentemente prega a indexação do grafo RDF como uma matriz esparsa, que contém um conjunto de informações pré-computadas para agilizar a extração de seções do grafo, e o uso de consultas baseadas em tensores sobre a matriz esparsa. Esta abordagem baseada em tensores permite que se tome vantagem de técnicas modernas de programação distribuída, e.g., a utilização de bases de dados não-relacionais fracionadas e o modelo de MapReduce. Nesta dissertação, propomos o desenho e exploramos a viabilidade da abordagem baseada em tensores, com o objetivo de construir um depósito de dados distribuído e agilizar a busca por palavras-chave com uma abordagem paralela.

Palavras-chave

Linked Data, MapReduce, Cloud Computing, Keyword Search

Contents

1	Introduction	12
1.1.	Motivation	12
1.2.	Assumptions	12
1.3.	Contribution	13
1.4.	Structure	14
2	Linked Data, Non-relational Databases and Cloud Computing	15
2.1.	Linked Data	15
2.2.	Non-relational databases	18
2.3.	Cloud Computing	22
2.4.	Summary	24
3	Search on Linked Data	25
3.1.	Tensor-based search	28
3.2.	Full-paths and templates	28
3.3.	Sparse matrix representation	31
3.4.	Retrieval and maintenance queries	32
3.5.	Summary	32
4	Distributed store for tensor-based RDF Graph Search	34
4.1.	Step one: distributed indexing of nodes, full-paths, and templates	35
4.2.	Step two: sparse matrix assembly, sharding and storage	40
4.3.	Step three: distributed graph queries	42
4.4.	Review of architecture requirements	48
4.5.	Architectural elements	50
4.6.	Proposed architecture	54
4.7.	Summary	54
5	Tools and implementation	56
5.1.	Tools and sharding	56
5.2.	Sharding	58

5.3. Networks	58
5.4. Details of the execution of RDF parsing and matrix building (steps 1 & 2)	59
5.5. Performance of the queries execution	61
5.6. Summary	61
6 Conclusion, related and future work	62
7 References	64

Figures

Figure 1 - The Linking Open Data Cloud Diagram [6]	16
Figure 2 - Two CD's listing example in RDF XML	17
Figure 3 - Two CD's example listing in N-Triples	18
Figure 4 - Subject-Predicate-Object graph representation.....	18
Figure 5 - MapReduce Overview (Wikimedia Commons).....	23
Figure 6 - Movies RDF Example Graph Representation.....	26
Figure 7 - Example search for keyword "Hitchcock" result as RDF Graph and immediate neighbours.....	27
Figure 8 - Sparse matrix as a bi-dimensional matrix of tuples	31
Figure 9 - Step one illustration, from RDF to indexes.....	35
Figure 10 - Architectural elements, orchestrating API and interactions.....	53
Figure 11 - Architecture with matching tools	57

Algorithms

Algorithm 1 - First approach to build nodes index	36
Algorithm 2 - Second approach to build nodes index.....	37
Algorithm 3 - Distributed approach to build nodes index	38
Algorithm 4 - Mapping the distribution of tuple parsing to find nodes.....	38
Algorithm 5 - Recursive DFS to walk all full-paths	39
Algorithm 6 - Mapping the distribution of sources to DFSs	39
Algorithm 7 - Store all positions of a full-path.....	41
Algorithm 8 - Mapping full-paths to store sparse matrix positions	41
Algorithm 9 - High-level description of a Node Query	42
Algorithm 10 - Naïve intersection of the two paths.....	43
Algorithm 11 - Path Intersection Retrieval Query	44
Algorithm 12 - Path Cutting Query	45
Algorithm 13 - Edge Deletion by Label - simplified without MapReduce	46
Algorithm 14 - Edge Deletion - simplified without MapReduce	47

Tables

Table 1 - Movies RDF Example SPO Tuples.....	25
Table 2 - RDF Search example over tuples for the keyword "Hitchcock"	26
Table 3 - Example of tables for nodes, full-paths and templates indexes.....	30
Table 4 - Requirements of generic shared data structures	48
Table 5 - Methods requiring specific distributed data structures.....	49
Table 6 - Requirements for a keyword search engine.....	50
Table 7 - Methods requiring a sparse matrix with optimized access	50
Table 8 - Example of using namespaces for storing collections.....	51
Table 9 - Example of using inverted key naming convention	51
Table 10 - Details of the AWS instances used.....	59
Table 11 - Steps 1 and 2 jobs execution times.....	60
Table 12 - STW RDF graph details	61
Table 13 - Queries response time - times in milliseconds per request.....	61