On Web-scale Reasoning
Urbani, J.

2013

document version
Publisher's PDF, also known as Version of record

Link to publication in VU Research Portal

citation for published version (APA)

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 20. Aug. 2024
## Contents

1. Introduction 1  
   1.1. Scope of research 3  
   1.2. Summary of chapters 5  
   1.3. Collaborations 7  

I Reasoning before query time 9  

2. Forward-chaining reasoning with MapReduce 11  
   2.1. The MapReduce programming model 12  
      2.1.1. A simple MapReduce example: term count 13  
      2.1.2. Characteristics of MapReduce 14  
   2.2. RDFS reasoning with MapReduce 14  
      2.2.1. Example rule execution with MapReduce 15  
      2.2.2. Problems of RDFS reasoning with MapReduce 15  
      2.2.3. Loading schema triples in memory 18  
      2.2.4. Data preprocessing to avoid duplicates 19  
      2.2.5. Ordering the application of the RDFS rules 20  
   2.3. OWL reasoning with MapReduce 23  
      2.3.1. Challenges with OWL reasoning with MapReduce 24
2.3.2. Limit duplicates when performing joins between instance triples ........................................ 26
2.3.3. Build sameAs table to avoid exponential derivation ................................................................. 28
2.3.4. Perform redundant joins to avoid load balancing problems .................................................. 30
2.4. Evaluation ................................................. 31
   2.4.1. Implementation ........................................ 32
   2.4.2. Experimental parameters ........................................ 32
   2.4.3. Dataset and reasoning complexity ......................... 35
   2.4.4. Scalability ............................................... 36
   2.4.5. Platform ................................................. 39
2.5. Related work .............................................. 40
2.6. Conclusion ................................................ 41

3. Distributed RDF data compression ........................................ 45
   3.1. Dictionary Encoding ........................................... 46
   3.2. MapReduce Data compression ........................................ 48
       3.2.1. Job 1: caching of popular terms ....................... 50
       3.2.2. Job 2: deconstruct statements, and assign IDs to terms ............... 51
       3.2.3. Job 3: reconstruct statements ....................... 54
       3.2.4. Storing the term IDs .................................. 54
   3.3. MapReduce data decompression ....... .......................... 55
       3.3.1. Job 2: join with dictionary table ....................... 56
       3.3.2. Job 3: join with compressed input ....................... 56
   3.4. Evaluation ................................................. 57
       3.4.1. Runtime ............................................... 58
       3.4.2. Performance of the popular-term cache ............... 60
       3.4.3. Scalability ............................................... 61
   3.5. Related work .............................................. 64
   3.6. Conclusions and Future Work ........................................ 65

4. Querying RDF data with Pig ........................................... 67
   4.1. SPARQL with Pig: overview ........................................ 69
       4.1.1. Runtime query optimization ....................... 70
       4.1.2. Pig-aware cost estimation ....................... 72
       4.1.3. Dealing with Skew .................................. 73
   4.2. Evaluation ................................................. 77
       4.2.1. Experiments ............................................... 78
   4.3. Related Work .............................................. 83
   4.4. Conclusions ................................................ 85
IV  Appendices

A. MapReduce Reasoning algorithms  163
   A.1. RDFS MapReduce algorithms  163
   A.2. OWL MapReduce algorithms  167

B. SPARQL queries  173
   B.1. Queries for Yahoo! use-case  173
   B.2. BSBM queries  174
   B.3. LUBM queries  175

Bibliography  177