

# VU Research Portal

## 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)**

Urbani, J. (2013). *On Web-scale Reasoning*.

### **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](mailto:vuresearchportal.ub@vu.nl)

---

Contents

---

<b>1. Introduction</b>	<b>1</b>
1.1. Scope of research . . . . .	3
1.2. Summary of chapters . . . . .	5
1.3. Collaborations . . . . .	7
<b>I Reasoning before query time</b>	<b>9</b>
<b>2. Forward-chaining reasoning with MapReduce</b>	<b>11</b>
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 . . . . .	17
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
<b>3.</b>	<b>Distributed RDF data compression</b> . . . . .	<b>45</b>
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
<b>4.</b>	<b>Querying RDF data with Pig</b> . . . . .	<b>67</b>
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

---

<b>II</b>	<b>Reasoning at query time</b>	<b>87</b>
<b>5.</b>	<b>Hybrid-reasoning</b>	<b>89</b>
5.1.	Hybrid reasoning: Overview . . . . .	90
5.2.	Hybrid Reasoning: Backward-chaining . . . . .	92
5.2.1.	Our approach . . . . .	95
5.2.2.	Exploiting the precomputation for efficient execution. . . . .	101
5.3.	Hybrid Reasoning: Pre-Materialization . . . . .	102
5.3.1.	Pre-Materialization algorithm . . . . .	102
5.3.2.	Reasoning with Pre-Materialized Predicates . . . . .	104
5.4.	Hybrid reasoning for OWL RL . . . . .	108
5.4.1.	Detecting duplicate derivation in OWL RL . . . . .	111
5.5.	Evaluation . . . . .	113
5.5.1.	Performance of the pre-materialization algorithm . . . . .	113
5.5.2.	Performance of the reasoning at query time . . . . .	115
5.5.3.	Discussion . . . . .	120
5.6.	Related Work . . . . .	121
5.7.	Conclusions . . . . .	122
<b>6.</b>	<b>Reasoning and SPARQL on a distributed architecture</b>	<b>125</b>
6.1.	System architecture . . . . .	126
6.2.	Data Storage . . . . .	127
6.3.	Rule Execution . . . . .	129
6.4.	SPARQL queries . . . . .	135
6.5.	Evaluation . . . . .	138
6.5.1.	Performance . . . . .	139
6.5.2.	Scalability . . . . .	141
6.5.3.	Efficiency . . . . .	143
6.6.	Related Work . . . . .	144
6.7.	Future Work and Conclusions . . . . .	146
<b>III</b>	<b>Discussion and conclusions</b>	<b>149</b>
<b>7.</b>	<b>Conclusions: Towards a reasonable Web</b>	<b>151</b>
7.1.	1 <sup>st</sup> Law: Treat schema triples differently . . . . .	153
7.2.	2 <sup>nd</sup> Law: Data skew dominates the data distribution . . . . .	154
7.3.	3 <sup>rd</sup> Law: Certain problems only appear at a very large scale . . . . .	156
7.4.	Conclusions . . . . .	158

<b>IV Appendices</b>	<b>161</b>
<b>A. MapReduce Reasoning algorithms</b>	<b>163</b>
A.1. RDFS MapReduce algorithms . . . . .	163
A.2. OWL MapReduce algorithms . . . . .	167
<b>B. SPARQL queries</b>	<b>173</b>
B.1. Queries for Yahoo! use-case . . . . .	173
B.2. BSBM queries . . . . .	174
B.3. LUBM queries . . . . .	175
<b>Bibliography</b>	<b>177</b>