



UvA-DARE (Digital Academic Repository)

Querying XML : benchmarks and recursion

Afanasiev, L.

Publication date
2009

[Link to publication](#)

Citation for published version (APA):

Afanasiev, L. (2009). *Querying XML : benchmarks and recursion*. [Thesis, fully internal, Universiteit van Amsterdam].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Contents

Acknowledgments	v
1 Introduction	1
1.1 Research questions	3
1.2 Main contributions	5
1.3 Organization of the thesis	6
1.4 Publications underlying the thesis	7
2 Background	9
2.1 The Extensible Markup Language (XML)	9
2.1.1 The XML tree model	11
2.1.2 XML data characteristics	12
2.2 XML query languages	14
2.2.1 XPath	14
2.2.2 XQuery	15
2.3 XML query processing	17
2.3.1 Approaches and implementations	18
2.4 Performance evaluation of XML query processors	19
2.4.1 Benchmarking	19
I Benchmarks	23
3 Analysis of XQuery Benchmarks	25
3.1 Introduction	26
3.2 Summary of existing XQuery benchmarks	29
3.2.1 Introducing the benchmarks	29
3.2.2 Characterizing and comparing the benchmarks	34
3.2.3 Conclusions	38

3.3	Benchmark query analysis	38
3.3.1	Language feature coverage	39
3.3.2	Query language coverage	42
3.3.3	Conclusions	44
3.4	Survey of benchmark usage	45
3.5	Correcting and standardizing the benchmark queries	48
3.5.1	Detecting outdated syntax and errors	48
3.5.2	Correcting the queries	50
3.5.3	Other issues	54
3.5.4	Conclusion	57
3.6	Running the benchmarks	57
3.6.1	Failed measurements	58
3.6.2	Comparing the performance of different engines	61
3.6.3	Performance on language features	62
3.6.4	Conclusions	63
3.7	Micro-benchmarking with MBench	63
3.7.1	MBench join queries	64
3.7.2	Evaluating Qizx/Open on the MBench join queries	66
3.7.3	Conclusions	67
3.8	Conclusions	68
3.8.1	Recommendations and next steps	70
4	Repeatability of Experimental Studies	71
4.1	Introduction	71
4.2	SIGMOD repeatability review setup	73
4.3	Describing experimental studies	74
4.3.1	Assessing repeatability	77
4.4	Results	78
4.5	Authors survey	81
4.6	Lessons learned and conclusions	83
5	XCheck: a Tool for Benchmarking XQuery Engines	85
5.1	Introduction	85
5.2	XCheck	87
5.2.1	Architecture and workflow	88
5.2.2	Collecting performance times	91
5.3	XCheck in action	93
5.3.1	Running XMark	93
5.3.2	XCheck's coverage	103
5.4	Related systems	105
5.5	Summary and conclusion	106

6	A Repository of Micro-Benchmarks for XQuery	109
6.1	Introduction	109
6.2	The MemBeR micro-benchmarking methodology	111
6.2.1	Micro-benchmark design principles	112
6.2.2	Micro-benchmark structure	114
6.2.3	Micro-benchmarking methodology	115
6.2.4	Preliminary classification of micro-benchmarks	116
6.2.5	Data sets for micro-benchmarking	117
6.3	The MemBeR repository	118
6.3.1	An example of MemBeR micro-benchmark	119
6.3.2	Meeting the design principles	123
6.4	Conclusions	124
7	A Micro-Benchmark for Value-Based Equi-Joins	125
7.1	Introduction	125
7.2	A micro-benchmark for value-based equi-joins	127
7.2.1	Target	127
7.2.2	Measure	127
7.2.3	Parameters	128
7.2.4	Documents	130
7.2.5	Queries	131
7.2.6	Running scenarios	134
7.2.7	Analyzing benchmark results	135
7.3	The micro-benchmark in action	137
7.3.1	SaxonB	139
7.3.2	Qizx/Open	141
7.3.3	Galax	143
7.3.4	MonetDB/XQuery	146
7.3.5	Lining up the micro-benchmark results	147
7.4	Conclusions	148
II	Recursion	151
8	An Inflationary Fixed Point Operator for XQuery	153
8.1	Introduction	153
8.2	Defining an Inflationary Fixed Point operator for XQuery	156
8.2.1	Using IFP to compute Transitive Closure	158
8.2.2	Comparison with IFP in SQL:1999	159
8.3	Algorithms for IFP	160
8.3.1	<i>Naïve</i>	160
8.3.2	<i>Delta</i>	161
8.4	Distributivity for XQuery	162

8.4.1	Defining distributivity	162
8.4.2	Trading <i>Naïve</i> for <i>Delta</i>	164
8.4.3	Translating TC into IFP	165
8.4.4	Undecidability of the distributivity property	167
8.5	A syntactic approximation of distributivity	167
8.5.1	The distributivity-safe fragment of XQuery	168
8.5.2	Distributivity-safety implies distributivity	171
8.6	An algebraic approximation of distributivity	177
8.6.1	An algebraic account of distributivity	179
8.6.2	Algebraic vs. syntactic approximation	181
8.7	Practical impact of distributivity and <i>Delta</i>	183
8.8	Related work	187
8.9	Conclusions and discussions	188
9	Core XPath with Inflationary Fixed Points	191
9.1	Introduction	191
9.2	Preliminaries	193
9.2.1	Core XPath 1.0 extended with IFP (CXP+IFP)	193
9.2.2	Propositional Modal Logic extended with IFP (ML+IFP)	195
9.3	ML+IFP vs. CXP+IFP	196
9.4	CXP+IFP and ML+IFP are undecidable on finite trees	199
9.4.1	2-Register machines	199
9.4.2	The reduction	201
9.5	Discussions and conclusions	208
9.5.1	Remaining questions	209
10	Conclusions	211
10.1	Answering the research questions	211
10.2	Outlook and further directions	216
A	LiXQuery: a Quick Syntax Reference	219
	Summary	231
	Samenvatting	233