

Why and how to keep XML succinct

Raymond Wong

National ICT Australia

raymond.wong@nicta.com.au



Australian Government
**Department of Communications,
Information Technology and the Arts**
Australian Research Council

NICTA Members



Department of State and
Regional Development

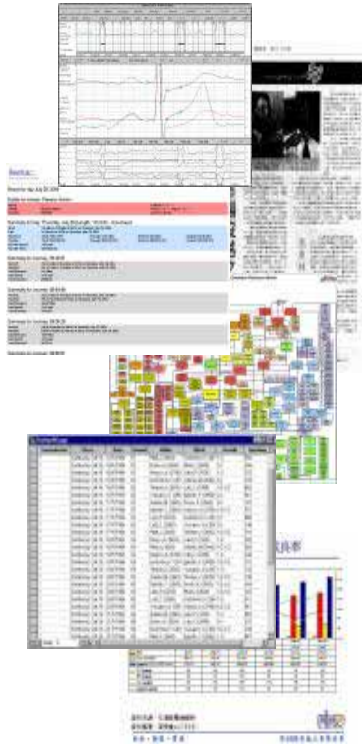


NICTA Partners

Outline

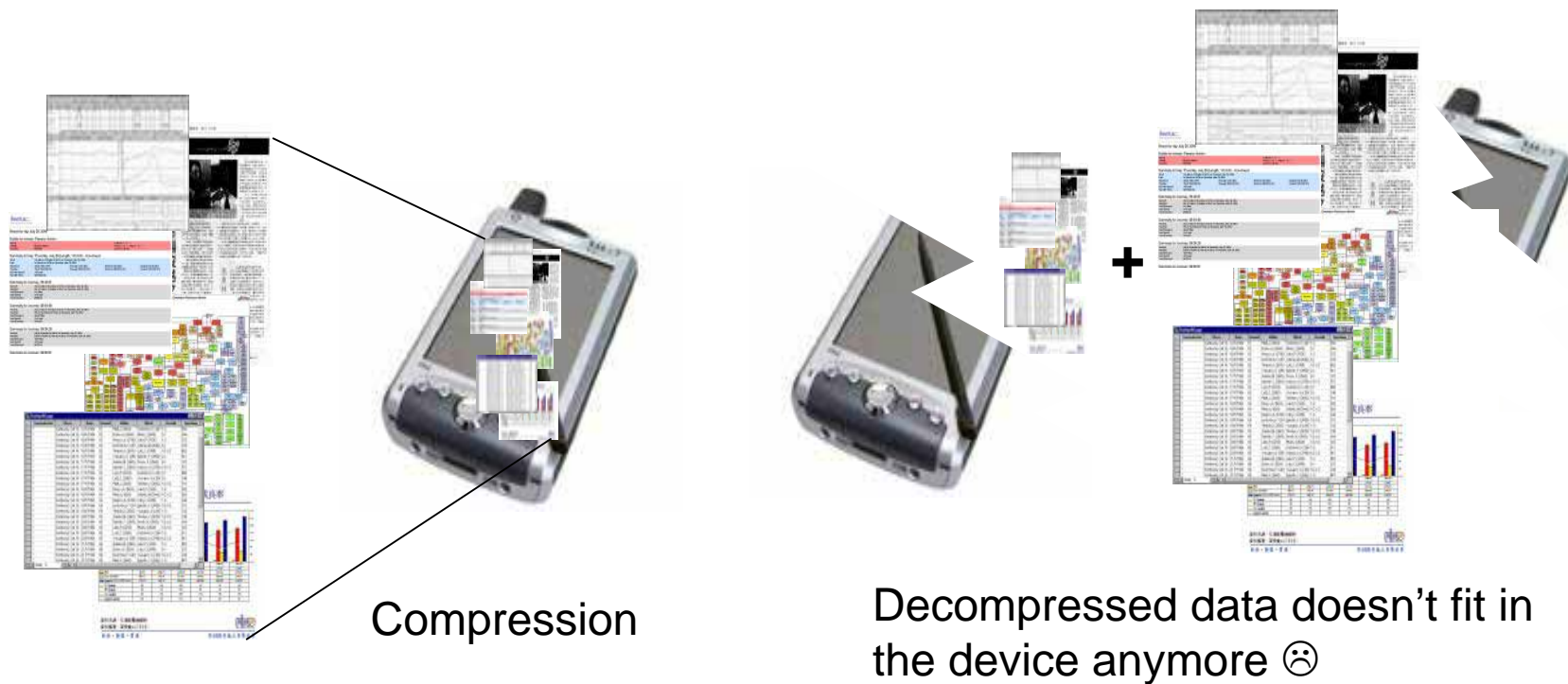
- Motivations
- Architecture
- How it works
- Experiments
- Conclusion

Motivations: XML everywhere



Problem of XML Compression

- When reading the compressed data
 - Need to decompress the data
 - Memory requirement = (compressed + decompressed) data



Even you have a large CF card

- For powerful hardware, DOM may be too large even if decompressed data fits in the device
- e.g., DOM = 10 x original doc size
=> = 50 x compressed doc size

So size of memory footprint is critical !!!



Decompression



Runtime footprint

Requirements

1. Space does matter for many applications
2. Generally reducing space improves cache locality
3. Indirection is expensive
4. Support fast navigations
5. Support fast insertion and deletion
6. Support efficient joins
7. Separate topology, text and schema

Our Solution – An Integrated Succinct XML (ISX) storage scheme

- A space-efficient storage scheme for XML data without compromising both query and update performances

Example

- 100 MB DBLP document
- 5 million XML nodes

- ISX: 1MB topology

Another example

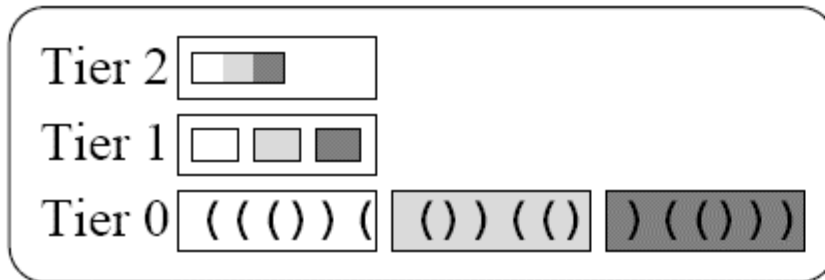
- Core Duo 1.83GHz
- 1GB RAM
- 5400 RPM Harddrive
- MS Vista

5M DBLP	MSXML	ISX
Runtime (loading)	15MB	4MB
Loading time	0.54s	0.035s
Runtime (//www)	21MB	4MB
//www	0.096s	0.004s

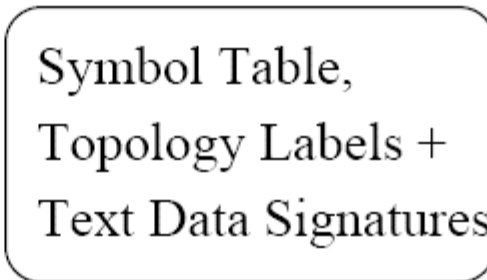
100M DBLP	MSXML	ISX
Runtime (loading)	329MB	67MB
Loading time	17.8s	0.67s
Runtime (//www)	333MB	67MB
//www	1.814s	0.143s

Proposed Storage Structure

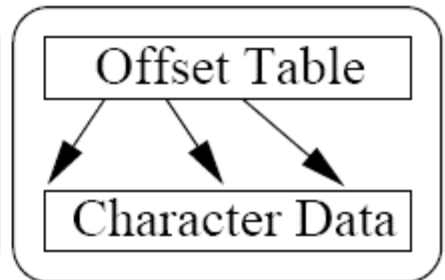
Topology Layer



Internal Node Layer (Tags)

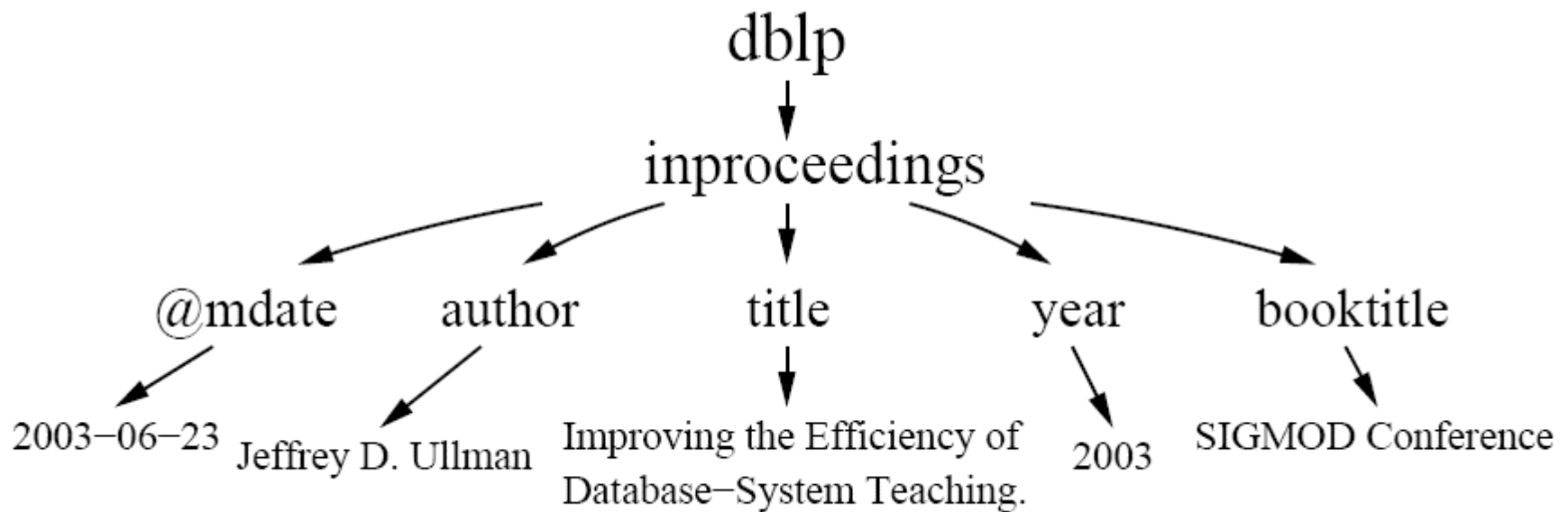


Leaf Node Layer (Text Data)

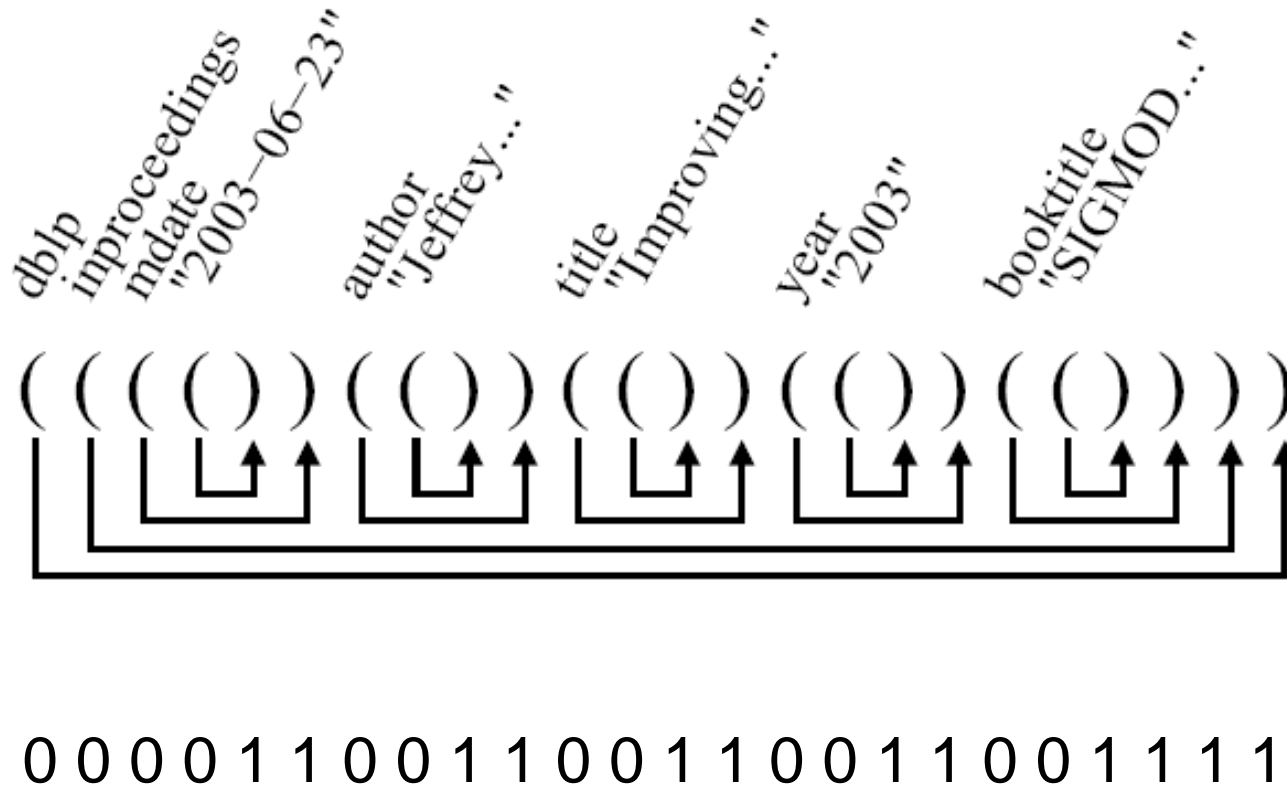


The ISX Structure

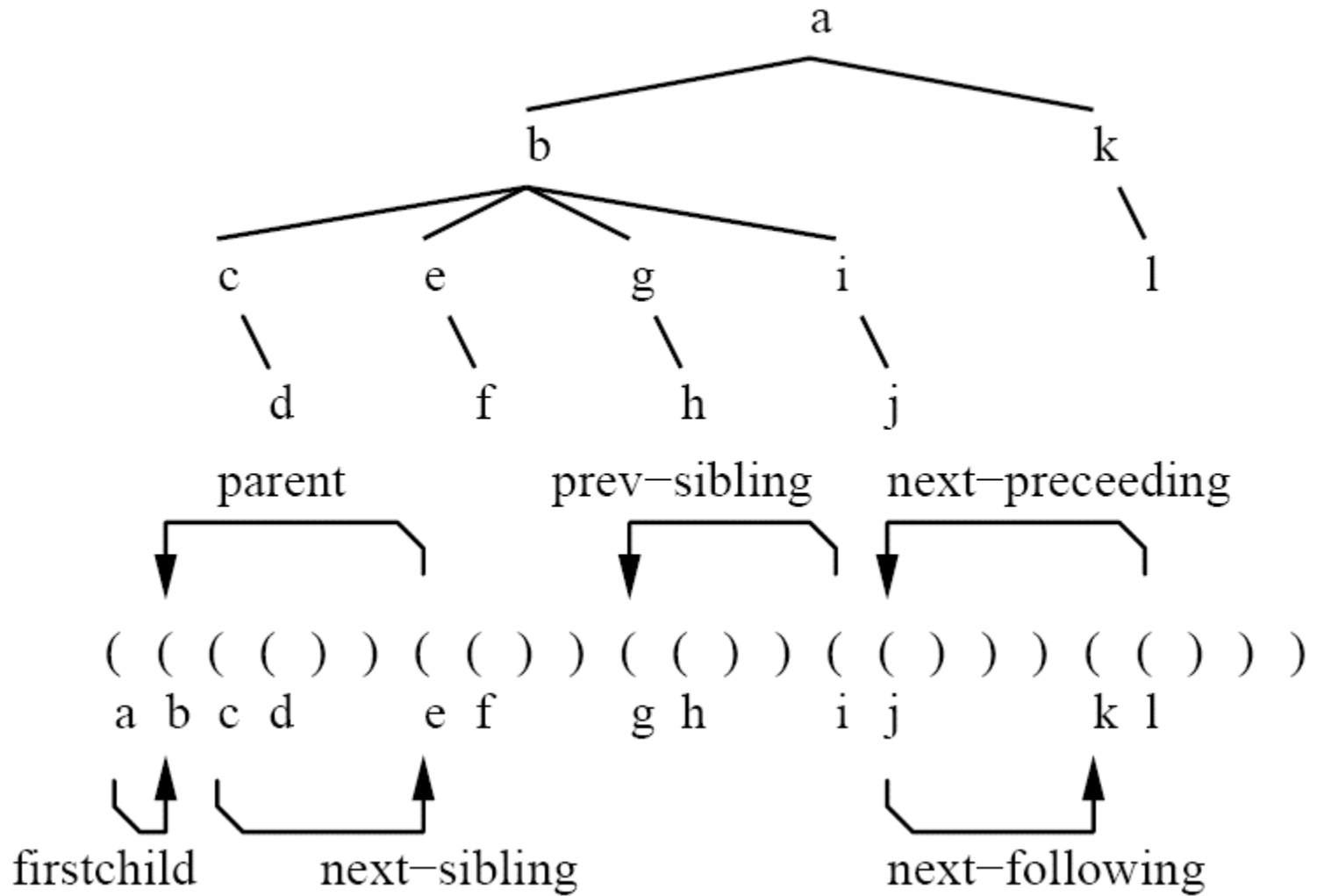
Sample DBLP XML Fragment



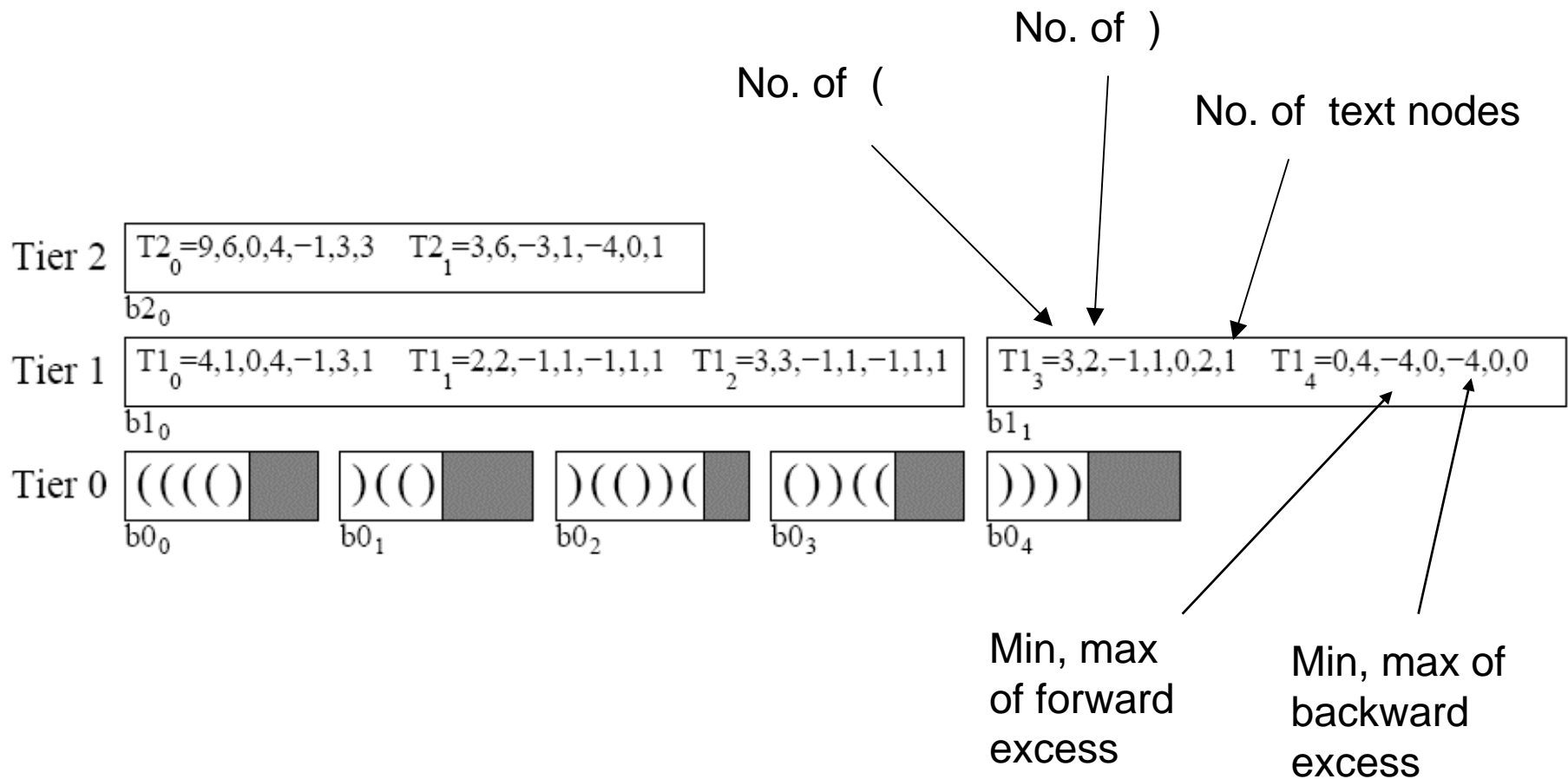
Balanced Parenthesis Encoding



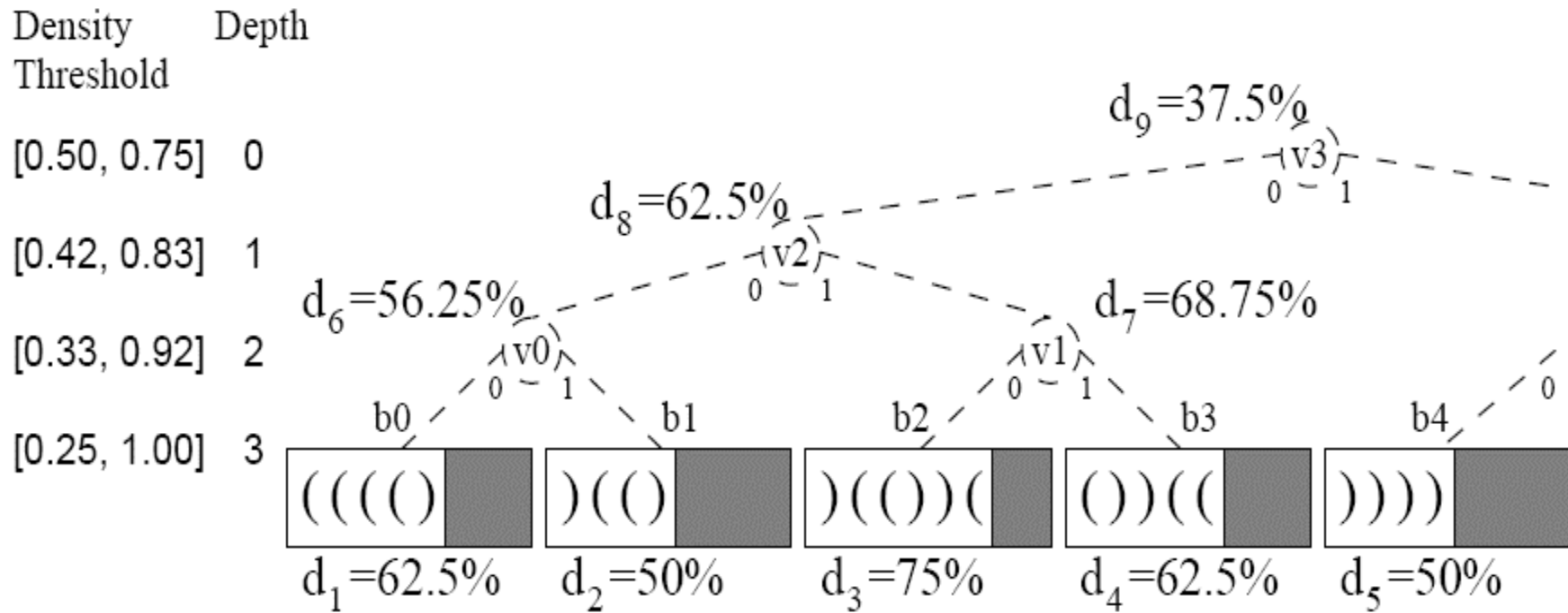
Node Navigations



Topology Tiers



Efficient Updates



d: density within a range of blocks

height of virtual binary trie: 3

ISX Features

Features	XMill	XGrind	NoK	TIMBER	ISX
Compression	✓	✓			✓
Document Traversal	✓	✓	✓	✓	✓
Node Navigation of All Axes			uncertain	✓	✓
Update Operation				✓	✓
Support XPath Query		✓	✓	✓	✓

TABLE I
COMPARISON OF SUPPORTED FEATURES

Experiments

Setup

- Fixed at **64MB memory buffer**
- Up to 16 GB XML document
- E.g. 16 GB DBLP contains > 770 million nodes
- **NO** index or query optimization has been employed for ISX (*except for ISX Stream where TurboXPath algorithm has been employed*)

Storage Size (ISX vs NoK)

Document Size (MB)	DBLP		PSD		TreeBank	
	NoK	ISX	NoK	ISX	NoK	ISX
5	18	3.64	17.91	3.36	19	3.21
10	35	7.23	36.12	6.82	38	6.36
50	181	36.1	182.42	34.14	196	31.78
100	367	72.1	377.52	68.74	389	63.43
250	918	180.2	950	171.9	974	159

TABLE II
STORAGE SIZE OF ISX vs. NoK

Storage Size (ISX, XMill, XGrind): DBLP

Source Data (MB)	ISX (MB)	ISX Compressed (MB)	XMill (MB)	XGrind (MB)	Source Data (MB)	ISX (MB)	ISX Compressed (MB)	XMill (MB)	XGrind (MB)
1	1	0.4	0.1	0.3	256	182	82.7	31.5	75.0
2	1	0.7	0.3	0.6	500	363	163.7	62.6	Failed
5	3	1.5	0.5	1.3	750	549	249.7	94.0	Failed
8	5	2.5	0.9	2.1	1000	726	327.5	125.3	Failed
16	10	5	1.8	4.3	2000	1452	654.9	250.5	Failed
32	21	10	3.7	8.6	4000	2903	1309.8	501.0	Failed
64	42	20	7.2	17.4	8000	5807	2619.6	978.48	Failed
128	87	40.2	14.9	35.8	16000	9411	4629.9	1952.81	Failed

TABLE III

STORAGE SIZE OF ISX (WITH AND WITHOUT TEXT COMPRESSION), XMILL AND XGRIND ON DBLP

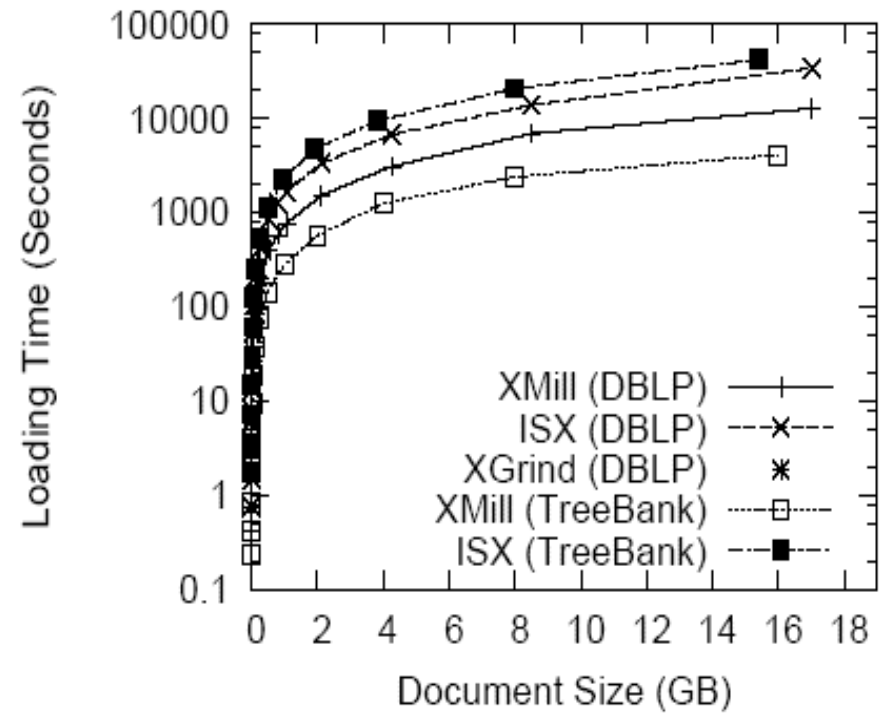
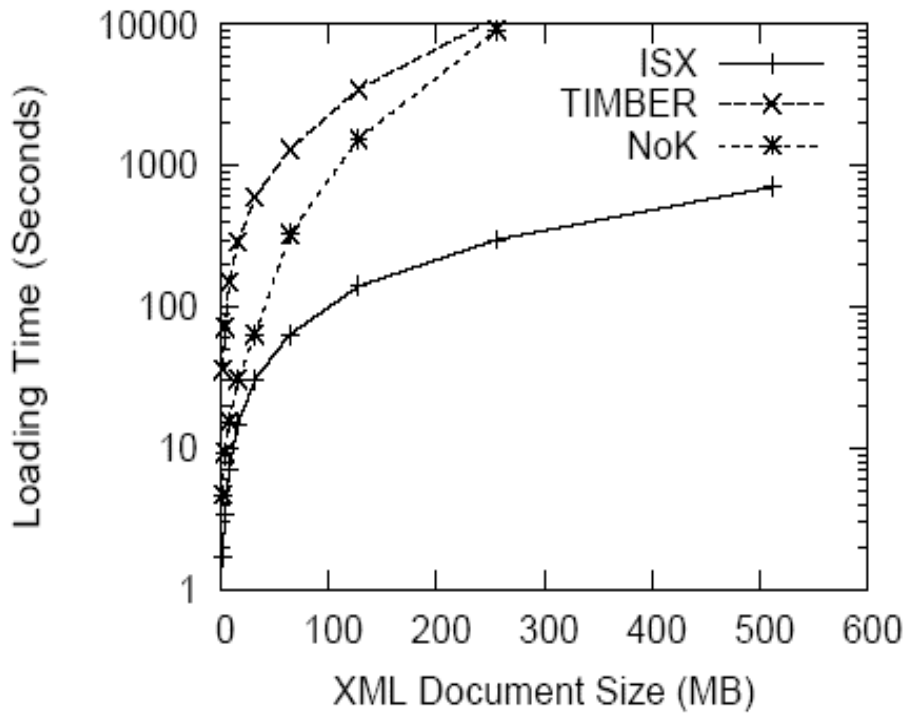
Storage Size (ISX, XMill): TreeBank

Source Data (MB)	ISX (MB)	ISX Compressed (MB)	XMill (MB)	Source Data (MB)	ISX (MB)	ISX Compressed (MB)	XMill (MB)
1	0.51	0.41	0.30	256	131.08	104.53	73.38
2	1.02	0.81	0.58	500	243.72	192.79	146.74
4	2.04	1.63	1.16	750	365.50	289.21	220.10
8	4.09	3.26	2.30	1000	487.43	385.58	293.489
16	8.19	6.53	4.60	2000	974.69	770.98	586.969
32	16.39	13.07	9.19	4000	1949.39	1541.97	1173.93
64	32.77	44.49	18.35	8000	4052.58	3205.59	2347.85
128	65.54	52.26	36.69	16000	7797.56	6167.87	4695.7

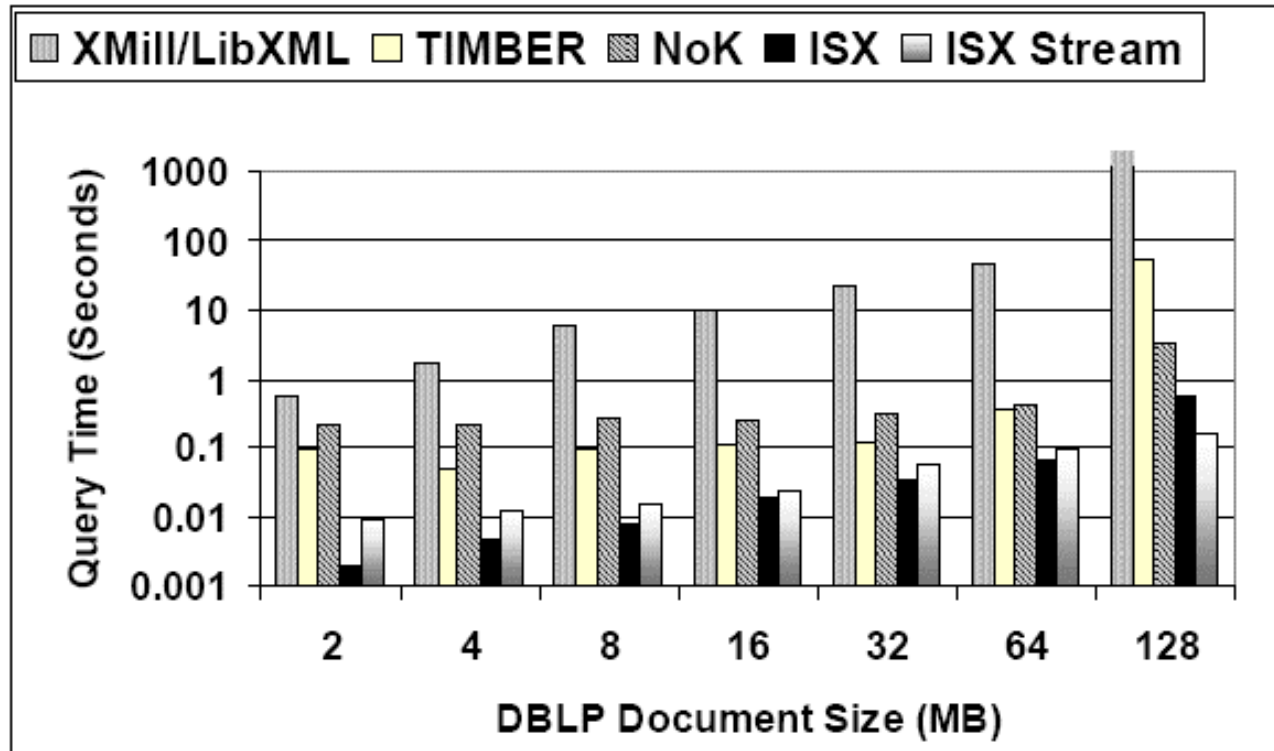
TABLE IV

STORAGE SIZE OF ISX (WITH AND WITHOUT TEXT COMPRESSION), XMILL ON TREEBANK

Bulk Loading Performance

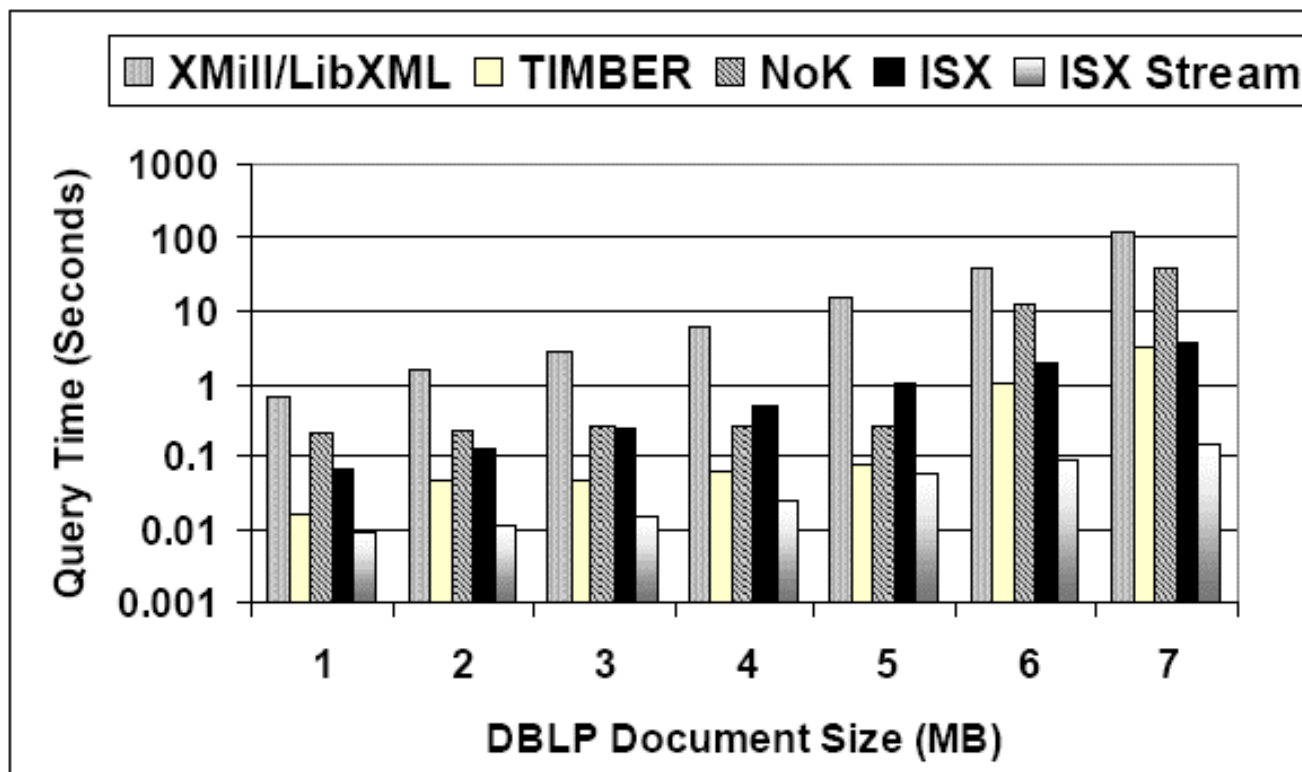


Q1: //inproceedings



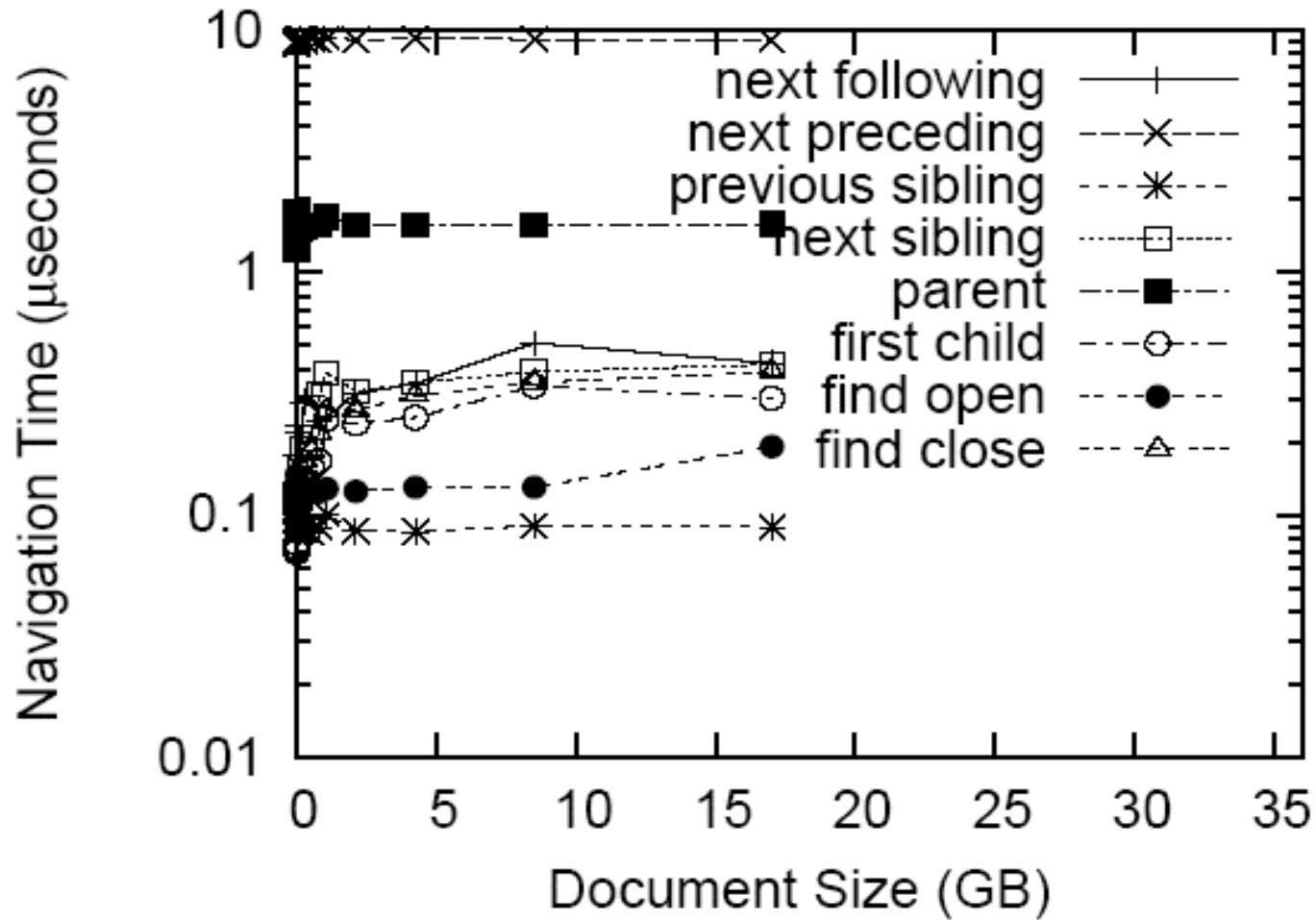
(a) DBLP Q1

Q5: `//article[./month/text() = "July"]//title`

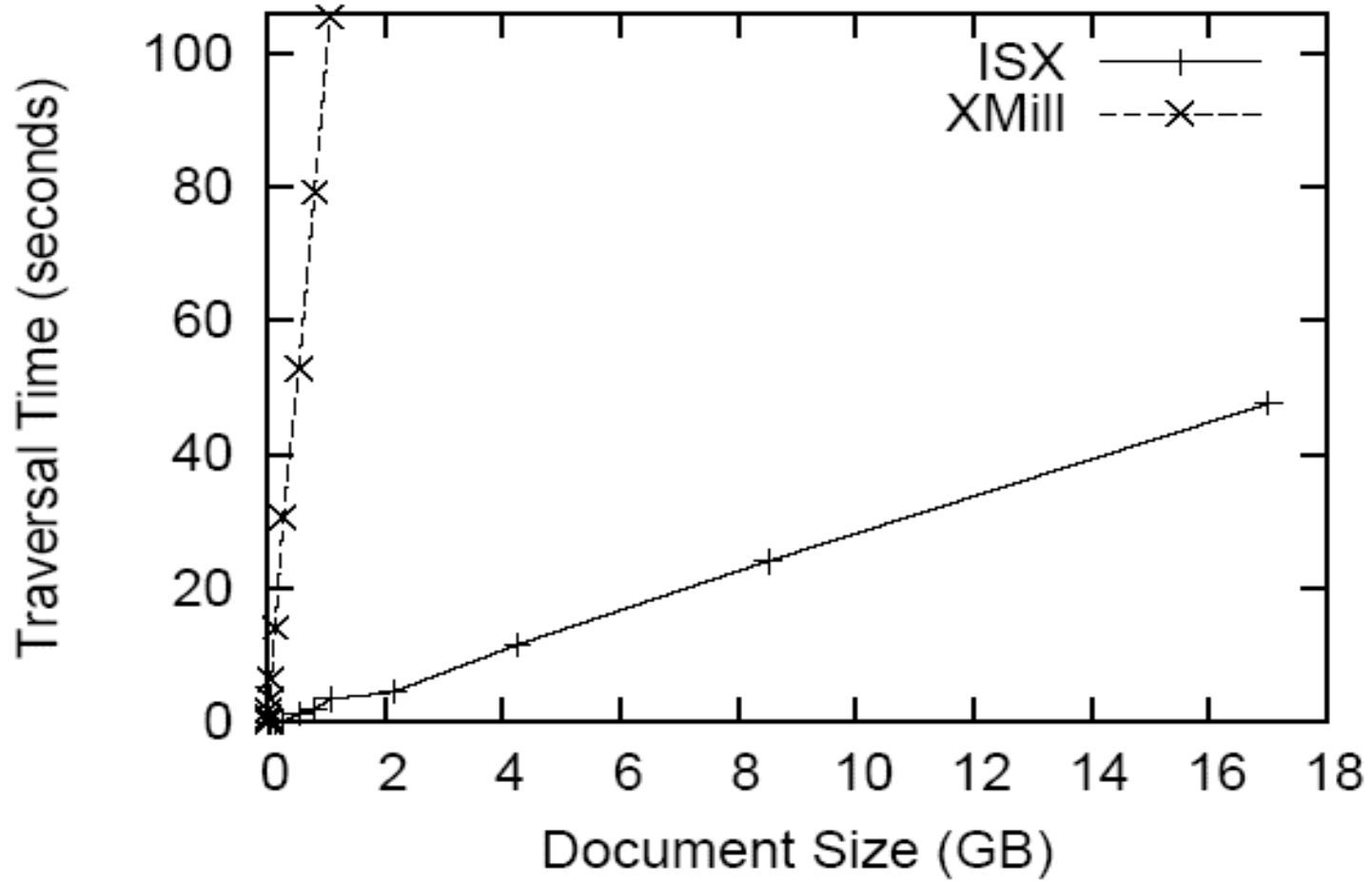


(e) DBLP Q5

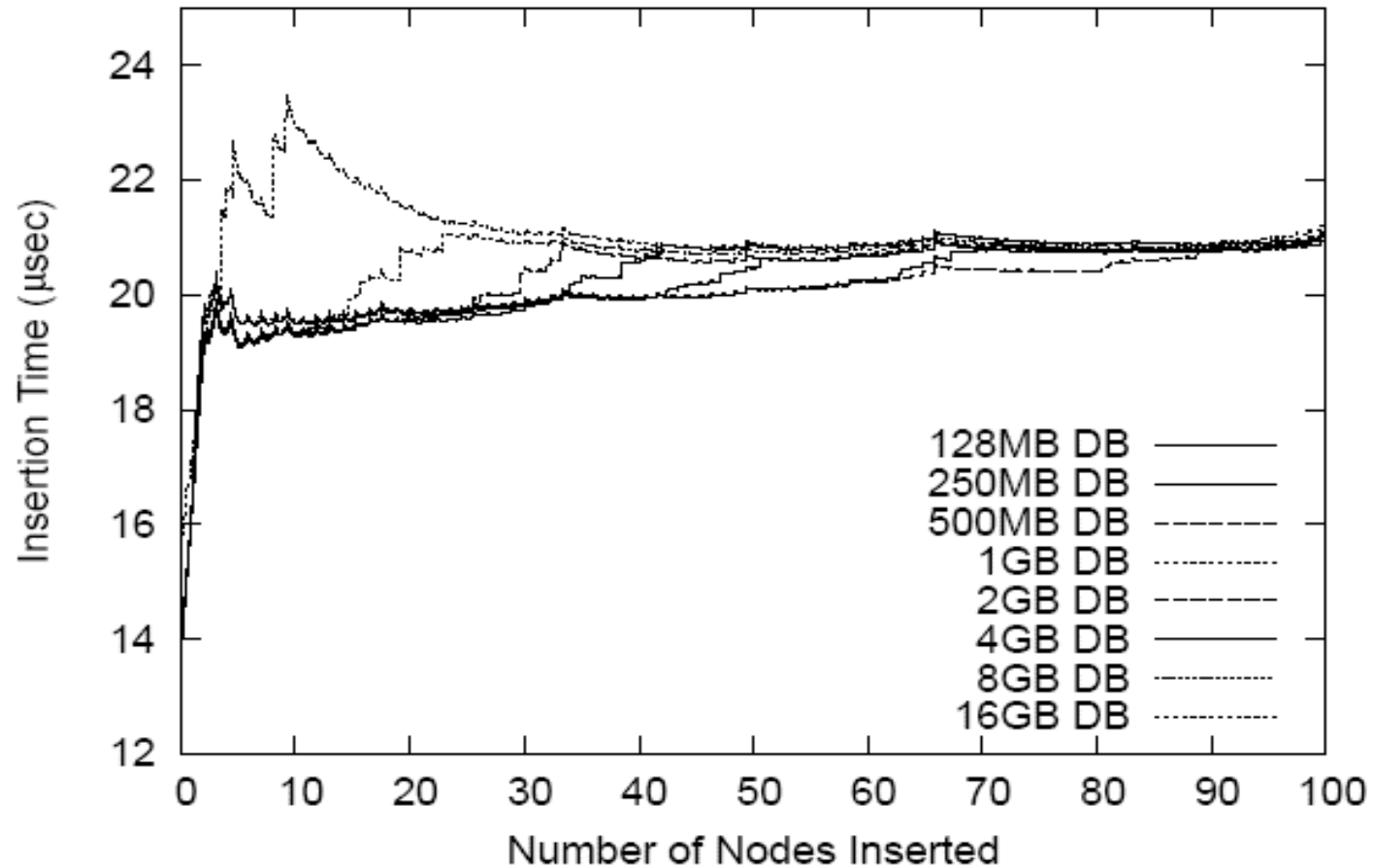
Node Navigation



Full document traversal



Update (Insertion) Performance



Conclusions

- Small storage footprint
- Small runtime footprint
- Fast and consistent performance on navigational access
- Superior query performance (further indexing / query optimization can be added)
- Superior update performance