XML Benchmarking

Irena Mlynkova

irena.mlynkova@mff.cuni.cz

Charles University
Faculty of Mathematics and Physics
Department of Software Engineering
Prague, Czech Republic

Introduction

- XML = a standard for data representation and manipulation
  - A number of methods for efficient managing, processing, exchanging, querying, updating, compressing, … of XML documents

⇒ Question: How to find the optimal one for a particular application?

- Problems:
  - Methods are tested on distinct data
  - The implementations are not always available
  - Gathering testing data is not easy

Goals of the Presentation

- Overview, classification and evaluation of existing approaches to XML benchmarking
- Identification of the most striking open issues
- Discussion of possible solutions

Purpose?

- First step towards proposal and implementation of a robust and comprehensive XML benchmark

Content

1. **Overview and classification of existing approaches**
2. **Discussion of open issues**
3. **Conclusion**

Classification of Existing Methods

- Type of data
  - Real-world vs. synthetic
    - Realistic, but too simple, contain errors
  - Fixed vs. dynamic data sets/operations
- Tested application
  - XML parsers, validators, management systems, query engines, XSL processors, XML compressors, …
- Tested technology
  - DTD vs. XML Schema, XPath vs. XQuery, XPath 1.0 vs. XPath 2.0, …

Testing Sets of XML Data

- **Typical approach: fixed sets of (real-world) XML data**
  - **Rather interesting than useful**
    - **The Bible in XML, Shakespeare's plays, …**
  - **XML exports of databases – most common**
    - ***IMDb* (movies and actors), *DBLP* (scientific papers), *Medical Subject Headings* (medical terms), …**
  - **Repositories of real-world XML – some not originally in XML format**
    - ***INEX*, *Ibiblio, …***
  - **Special real-world XML data – uncommon structure**
    - **Protein sequences, RNAs, astronomical NASA data, linguistic trees, …**
- **Problem: Simple, without respective operations**

Benchmark Projects for XML Parsers and Validators (1)

- Primary application for XML data processing
- W3C: XML Conformance Test Suites
  - XML 1.0, XML 1.1 and Namespaces in XML 1.1
  - 2.000 XML documents
    - Valid, invalid and non-well-formed documents
    - Well-formed errors tied to external entity
    - Documents with optional errors
  - Binary tests:
    - Parser must accept/reject the document correctly
  - Output tests:
    - Parser must report information as required

Benchmark Projects for XML Parsers and Validators (2)

- Types of parsers
  - Event-driven – while reading they return data fragments
    - Push – reading cannot be influenced
    - Pull – read the next data only if they are “asked” to
  - Object-model – read the document and built it completely in memory
  - Various combinations

⇒ Need to be compared and tested

⇒ Number of papers which evaluate efficiency of subsets of known implementations

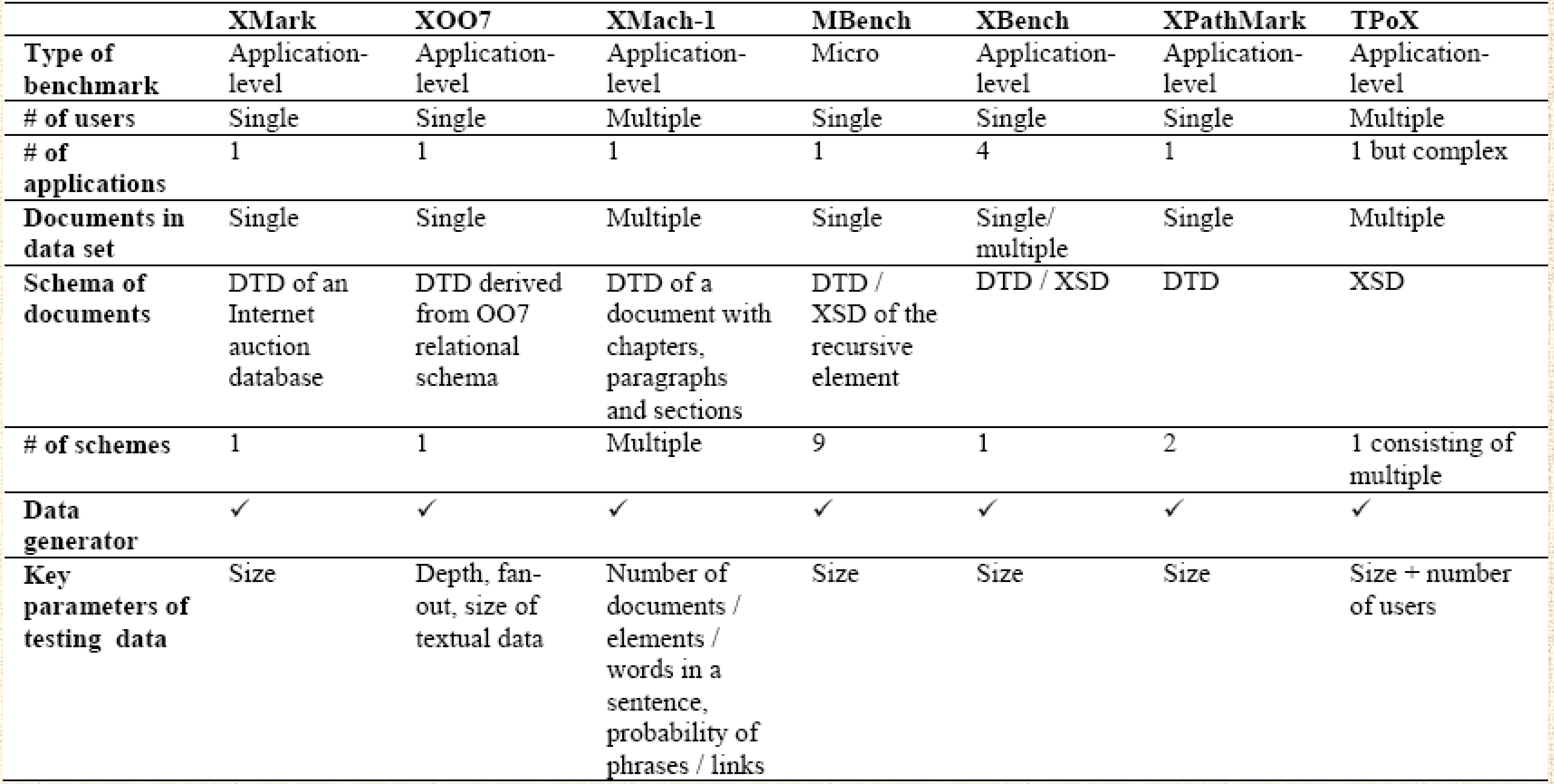
  - Compare same/different types of parsers
  - All the related data are available

- Problem: No true benchmarking project for parsers/validators

Benchmark Projects for XML MS and QE (1)

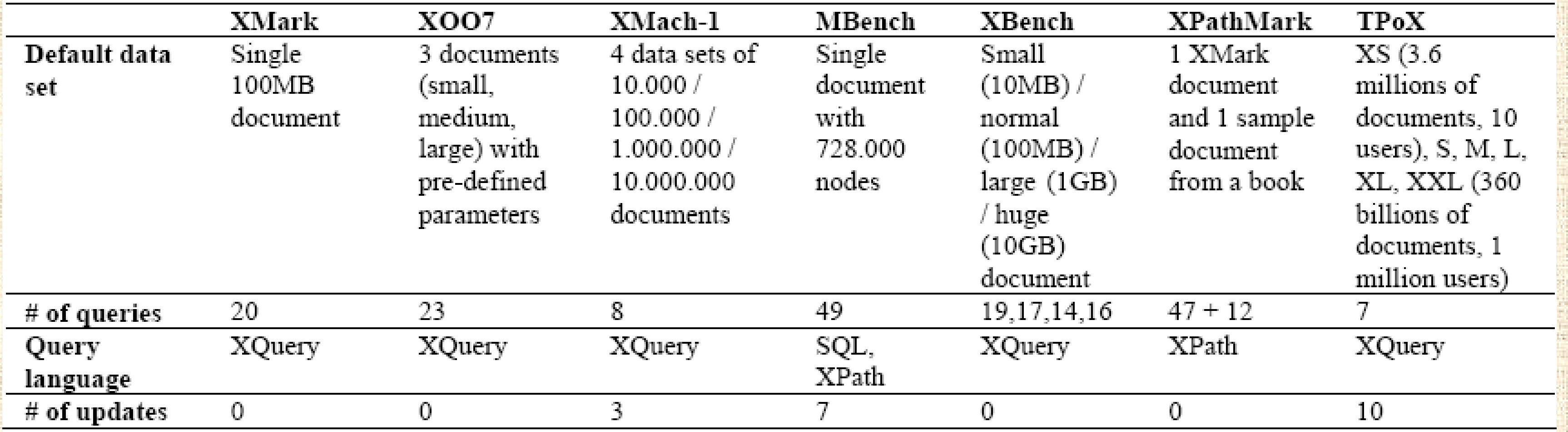
- The biggest set of benchmarks
- Test the amount of supported query constructs + efficiency of evaluation
  - Assumption: correct results ⇒ not tested
- Classification: query language, amount of users, …
- W3C:
  - XML Query Use Cases – not a benchmark, a set of examples of XML query applications
  - XML Query Test Suite – 15.000 test cases (queries and expected results), test support of XML Query constructs
- Best known representatives: XMark, XOO7, XMach-1, MBench, XBench, XPathMark, TPoX

Benchmark Projects for XML MS and QE (2)



	XMark	XOO7	XMach-1	MBench	XBench	XPathMark	TPoX
Type of benchmark	Application-level	Application-level	Application-level	Micro	Application-level	Application-level	Application-level
# of users	Single	Single	Multiple	Single	Single	Single	Multiple
# of applications	1	1	1	1	4	1	1 but complex
Documents in data set	Single	Single	Multiple	Single	Single/ multiple	Single	Multiple
Schema of documents	DTD of an Internet auction database	DTD derived from OO7 relational schema	DTD of a document with chapters, paragraphs and sections	DTD / XSD of the recursive element	DTD / XSD	DTD	XSD
# of schemes	1	1	Multiple	9	1	2	1 consisting of multiple
Data generator	✓	✓	✓	✓	✓	✓	✓
Key parameters of testing data	Size	Depth, fan-out, size of textual data	Number of documents / elements / words in a sentence, probability of phrases / links	Size	Size	Size	Size + number of users

Benchmark Projects for XML MS and QE (3)



	XMark	XOO7	XMach-1	MBench	XBench	XPathMark	TPoX
Default data set	Single 100MB document	3 documents (small, medium, large) with pre-defined parameters	4 data sets of 10.000 / 100.000 / 1.000.000 / 10.000.000 documents	Single document with 728.000 nodes	Small (10MB) / normal (100MB) / large (1GB) / huge (10GB) document	1 XMark document and 1 sample document from a book	XS (3.6 millions of documents, 10 users), S, M, L, XL, XXL (360 billions of documents, 1 million users)
# of queries	20	23	8	49	19,17,14,16	47 + 12	7
Query language	XQuery	XQuery	XQuery	SQL, XPath	XQuery	XPath	XQuery
# of updates	0	0	3	7	0	0	10

- **Type of benchmark:**
  - **Application-level – compare and contrast distinct applications ⇒ queries are highly different**
  - **Micro – evaluate performance of a single system in distinct situations ⇒ similar queries, differentiate, e.g., in selectivity**
    - **MBench**

Benchmark Projects for XML MS and QE (4)

- Purpose of benchmark:
  - Number of users, applications, documents
  - Most: single-user, single-application, with single document
    - XBench – 4 classes of XML applications
      - Text-centric/single document, data-centric/multiple documents, …
    - XMach-1, TPoX – multi-user, test other XML management aspects
      - Indexing, schema validation, concurrency control, transaction processing, network characteristics, …
- Data sets:
  - All projects involve DTD/XSD and a simple data generator
    - Typical parameter: size of data
- Operations:
  - All projects involve a set of XQuery queries
  - XMach-1, MBench, TPoX – involve update operations
  - XMach-1, TPoX (multi-user benchmarks) ⇒ additional, less XML-like operations

Benchmark Projects for XML MS and QE (5)

- Analysis of benchmarks
  - Only 1/3 of papers use a kind of benchmark
  - 38% of benchmark queries are incorrect/out-dated
    - 29% of the queries are XPath 1.0 queries
    - 61% are XPath 2.0 queries
    - Only 10% cannot be expressed in XPath
  - XMark – most popular, simple ⇒ users do not want to bother with complex application
- Benchmark repository
  - Observation: A fixed set of queries ⇒ cannot test various aspects of applications

  ⇒ MemBeR repository of micro-benchmarks
    - New micro-benchmark/new result set must be specified as an XML document
    - Categories of benchmarks: XPath, query stability and XQuery

Other XML Technologies

- Basic: parsing, validating, querying
- Advanced: transformations, compressing, … ⇒ need for special purpose benchmarks
  - Problem: low number, representatives are obsolete
- Example: XSLT
  - XSLTMark – from 2000, not maintained, constructs of version 1.0 (from 1999, obsolete)
  - Analyses of implementations use XSLTMark
- Do we need special-purpose benchmarks?
  - NO: They are based on basic operations
  - YES: Exploitation of basic operations can differ

???

Content

1. Overview and classification of existing approaches
2. **Discussion of open issues**
3. Conclusion

1. General Requirements for Benchmarks

- 5 recommended requirements for DB benchmarks
- Are they necessary for XML MS benchmarks?
- Portability and scalability are natural
  - Do not restrict OS and/or HW
- Simplicity is user-friendly
  - The most popular benchmark: XMark
    - A fixed set of XML queries, single data parameter: size
- Domain-specificity and relevancy are arguable
  - XML technologies have plenty of usages ⇒ hard to specify a benchmark covering all
  - Benchmark restricted to a single use case cannot have much usage

  ⇒ Solution: Versatile benchmark, highly parameterized, but with pre-defined settings of the parameters

Simplicity

2. More Sophisticated Data Generator

- First step towards the versatile XML benchmark
- Existing benchmarks:
  - Simple data generator/complex data generator + fixed parameters
  - Deal with marginal problems
    - e.g. where to get the textual data
  - For some applications (e.g., XML full-text operations or XML compression) important, but for XML querying not
- Parameters:
  - Structure of XML document trees
  - Semantic of the data
    - DTD: ID, IDREF(S)
    - XSD: unique/key/keyref, assert/report, functional dependencies
- Collides with simplicity requirement $\Rightarrow$ predefined settings of parameters

3. Schema Generator

- Natural requirement: provide XML data with XML schema
- Two perspectives:
  - Data ⇒ schema
    - Techniques for automatic inference of an XML schema
    - Idea: Generalization of a trivial schema
      - “if there are more than three occurrences of an element, it is probable that it can occur arbitrary times”
    - Multiple possibilities how to generalize ⇒ user-specified parameters
  - Schema ⇒ data
    - Characteristics of XML documents are restricted
    - Remaining vague constructs ⇒ user-specified parameters
      - Operator *, recursion
    - Exploited in current data generators
      - XSD + predefined set of annotations
      - e.g. ToXgene generator

4. Query Generator

- Existing works: fixed set of queries ⇒ highly restricted data
- Idea: User knows characteristics of queries
  - Constructs that can be used in the query
    - e.g. axes, predicates, constructors, update operations, …
  - What kind of data the query should access
    - e.g. attributes, keys and foreign keys, mixed-content elements, recursive elements, …
  - Where the data are located
    - e.g. at what levels
  - What amount of data is required
    - e.g. elements with specified structure

5. Theoretic Study of Data Characteristics

- Aim: To support as much data characteristics as possible
- Problem: Subsets of the data are correlated
  - Not all possible settings are available
  - e.g. length of element contents vs. size of the document / number of elements vs. size of the document
  - e.g. depth of the document vs. element fan-out vs. size of the document
- MemBeR generator: brute force
  - Specifying depth, fan-out and size at the same time is not allowed
- Open issue: a theoretic study of the data characteristics
  - Classification, mutual influence and correlation

Content

1. Overview and classification of existing approaches
2. Discussion of open issues
3. **Conclusion**

Conclusion

- Contributions
  - Study on the state of the art and open issues of XML benchmarking projects
  - Aims:
    - To show that XML benchmarking is an up-to-date problem
    - Provide a reasonable source of information for researchers and analysts
- Current and future work:
  - Implementation of sophisticated data generator
    - Present: Huge amount of data characteristics, analysis of correlation, pre-defined sets of settings based on real world statistics
    - Future: Query generator

Thank you