

X

XML Benchmarks

DENILSON BARBOSA¹, IOANA MANOLESCU²,
JEFFREY XU YU³

¹Department of Computer Science, University of
Calgary, Calgary, AB, Canada

²INRIA Futurs, Le Chesnay, France

³Department of Systems Engineering and Engineering
Management, The Chinese University of Hong Kong,
Hong Kong, China

Definition

An XML benchmark is a specification of a set of meaningful and relevant tasks, intended to assess the functionality and/or performance of an XML processing tool or system. The benchmark must specify the following: (i) a deterministic workload, consisting of a set of XML documents and/or a procedure for obtaining these and a set of operations to be performed; (ii) detailed rules for executing the workload and making the measurements; (iii) the metrics used to report the results of the benchmark; and (iv) standard ways of interpreting the results.

Historical Background

XML has quickly become the preferred format for representing and exchanging data on the Web age. The level of acceptance of XML is astonishing, especially when one considers that this technology was introduced only in 1997. XML is an enabling technology with applications in virtually all domains of information processing. At the time of writing, XML is widely used in content distribution on the Web (e.g., RSS feeds), as the foundation of large initiatives such as the Semantic Web and Web Services, and is the basis for routinely used productivity tools, such as text editors and spreadsheets.

Such complexity led to the development of fairly narrowly-scoped benchmarks for XML. Moreover, at

print time, there is still no clear understanding of what an XML benchmark should look like.

The first XML benchmarks were developed in academia for testing specific processing tasks and/or relatively narrow applications. In fact, some of these earlier benchmarks are categorized as micro-benchmarks ([EDS ENTRY ON MICRO-BENCHMARKS]). For example, XMach emulated the scenario of an XML file servers using a large number of simple XML files, while XMark modeled an online auction application using a single complex XML document. Over time, XML benchmarks evolved to contemplate more realistic and complex application scenarios, in an attempt to emulate the typical workload of larger applications. For instance, XBench modeled four scenarios, resulting from the combination of two factors: (i) the nature of the documents (data-centric vs. document-centric); and (ii) the number of documents in the workload (single-document vs. multi-document).

The first XML Benchmark developed entirely by industry was TPoX (Transaction Processing over XML), which simulates a financial application in a multi-user environment with concurrent access to the XML data. TPoX is intended for testing all aspects of a relational-based XML storage and processing system.

Another development worth of mention concerns the declarative synthetic data generators for XML. While, strictly speaking, these are not benchmarks, these tools help in obtaining appropriate testing data with reasonably low effort and high enough customizability.

Scientific Fundamentals

Benchmarks should be simple, portable, scale to different workload sizes, and allow the objective comparisons of competing systems and tools [2]. It should be noted that the diversity and complexity of its applications, developing meaningful and realistic benchmarks for XML is a truly herculean task. Also, XML processing tools fall into many categories, from simple storage services to sophisticated query processors, thus

adding to the complexity of developing relevant and *realistic* XML benchmarks.

The two factors above have led to the development of benchmarks that are relatively narrow in scope, focusing on very specific tasks. Moreover, the lack of universally accepted, comprehensive XML benchmarks, resulted in the development of general-purpose synthetic data generators, which allow the user to obtain customized test data with relatively low effort. This is significant, as such tools were not popular until the advent of XML.

XML Microbenchmarks

A micro-benchmark is a narrowly-defined benchmark aimed at testing very specific aspects of a tool and/or system ([CITATION TO EDS ENTRY ON MICRO-BENCHMARK]).

The Michigan Benchmark The Michigan Benchmark is an XML Micro-benchmark developed at the University of Michigan [8]. It uses a single synthetic document that does not resemble a typical document from any real world application domain. Instead, it is carefully designed to allow the testing of the following query processing operations: matching attributes by value; selecting elements by name; evaluation of positional predicates; selection of nodes based on predicates over their parent, children, ancestors or descendants; join operations; computing aggregate functions; and processing updates. The authors applied the benchmark to three database systems: two native XML DBMSs, and a commercial ORDBMS.

XML Application Benchmarks

An application benchmark is a comprehensive set of tasks that approximates the workload of a typical application in the respective domain ([CITATION TO EDS ENTRY ON APPLICATION BENCHMARKS]). Four important XML application benchmarks are XMach-1 [4], XMark [9], XBench [10], and TPOX [7].

XMach-1 XMach-1 (*XML Data Management Benchmark*) is a multi-user benchmark developed at the University of Leipzig, Germany [4]. Unlike most existing XML benchmarks that are designed to test the query processors of database management systems, XMach-1 is designed to test database management systems which include a query processor as well as the other key components. In terms of measurement, XMach-1

evaluates systems based on throughput performance (XML queries per second) instead of response time for user-given XML queries.

XMach-1 considers a system architecture to support web applications, which consists of four main components, namely, XML database, application servers, loaders and browser clients. In the XML database, there are multiple schemas, and there are between 2 and 100 documents per schema. Each document is generated using 10,000 most frequent English words, and occupies between 2 and 100 KB of storage. The workload contains eight queries and three update operations. Some evaluation results can be found in [6].

XMark XMark is the result from the XML benchmark project, led by a team at CWI [9]. It models an Internet auctioning application. The workload consists of a large database, in a single XML document, containing: (i) items for auction in geographically dispersed areas; (ii) bidders and their interests; and (iii) detailed information about existing auctions, which can be open or closed. XMark's workload includes 20 queries that cover the following broad kinds of operations: simple node selections; document queries for which order information is relevant; navigational queries; and computing aggregate functions.

The XMark data generator employs several kinds of probability distributions and uses several real data values (e.g., names of countries and people) to produce realistic data; also, the textual content uses real words of the English language. XMark is by far the most widely used XML benchmark at the time of writing.

XBench XBench is a benchmark suite developed at the University of Waterloo [10]. XBench defines application benchmarks categorized according to two criteria: single-document versus multi-document and *data-centric* versus *text-centric* domains. The latter criterion distinguishes data management and exchange scenarios from content management applications (e.g., electronic publishing). Document collections in XBench range in size from a few kilobytes to several gigabytes, and its workload consists of bulk-loading as well as various query and text-based search operations. Results of an evaluation of four different systems, comprising both native XML stores as well as relational-based stores ([CITATION TO EDS ENTRY ON XML STORAGE]), are provided in [10].

TPoX TPoX (Transaction Processing over XML) is a comprehensive application benchmark developed jointly by IBM and Intel [7]. TPoX simulates a financial application domain (security trading) and is based on the industry-standard XML Schema specification FIXML [5]. The testing environment in TPoX covers several aspects of XML management inside DBMS, including the use of XQuery, SQL/XML, updates, and concurrent access to the data. The authors report on an experimental evaluation of the IBM DB2 product for storing and processing XML data [7].

Synthetic Data Generators

Synthetic data have other applications besides benchmarking, such as testing specific components of a complex system or application. In this setting, an important requirement for a data generator, besides generating *realistic* data (i.e., synthetic data whose characteristics match those of typical real data in the application), is the ability of easily customizing the test data (e.g., its structure).

Declarative synthetic data generators, on the other hand, are tools that produce synthetic data according to specifications that describe *what* data to generate, as opposed to *how* to generate such data, thus facilitating the generation of synthetic data. Declarative data generators are intended for easing the burden in obtaining test data, unlike the data generators of standardized benchmarks, which have the characteristics of the data they produce embedded in their source code.

Declarative data generators rely on formalisms providing higher levels of abstraction than programming languages, such as conceptual schema languages annotated with probabilistic information (for describing the characteristics of the intended data). Such probabilistic information are needed because schema languages specify only what content is allowed in valid document instances ([EDS ENTRY ON XML VALIDITY]). A realistic data generator must allow the specification of the characteristics of *typical* documents as well. For example, while a schema formalism will specify that a book element may contain between 1 and 10 authors, a realistic data generator will allow one to define a probability distribution for the number of authors in the test data. Another desirable feature of realistic synthetic data is that it satisfies integrity and referential constraints. For instance, an XML document describing a book review should refer to an existing book in the test data.

Two examples of declarative XML data generators are the IBM XML Generator [3], whose data specifications are based on Document Type Definitions ([EDS ENTRY ON DTDS/XML STANDARD]), and ToXgene [1], which relies on XML Schema specifications ([EDS ENTRY ON XML SCHEMA]). Both tools allow the specification of skewed probability distributions for elements, attributes, and textual nodes. ToXgene, being based on XML Schema, supports different data types, as well as key and referential constraints. ToXgene also offers a simple declarative query language that allows one to model relatively complex dependencies among elements and attributes involving arithmetic and string manipulation operations. For instance, it allows one to model that the total price of an invoice should be the sum of the individual prices of the items in that invoice multiplied by the appropriate tax rates. Finally, ToXgene offers support for generating recursive XML documents.

Key Applications

Meaningful benchmarks are essential for the development of new technologies, as they allow developers to assess progress and understand intrinsic limitations of their tools. Applications include functionality testing, performance evaluation, and system comparisons.

Future Directions

Optional. Open problems and discussions.

Cross-references

- ▶ Measurement
- ▶ Performance Benchmark

Recommended Reading

1. Barbosa D. and Mendelzon A.O. Declarative generation of synthetic XML data. *Softw. Pract. Exper.* 36(10):1051–1079, 2006.
2. Gray J. (ed.). *The Benchmark Handbook for Database and Transaction Systems* (2nd edn.). Morgan Kaufmann, San Francisco, CA, USA, 1993, ISBN 1-55860-292-5.
3. IBM XML Generator. Available at: <http://www.alphaworks.ibm.com/tech/xmlgenerator>, 2007.
4. Böhme T. and Rahm E. XMach-1: a benchmark for XML data management. In *Proceedings of the German Database Conference BTW2001*. Springer, Berlin, 2001, pp. 264–273; Multi-user evaluation of XML data Management Systems with XMach-1. *Lecture Notes in Computer Science (LNCS) 2590*, Springer, Berlin, 2003, pp. 148–159.
5. Financial Information Exchange Protocol. FIXML 4.4 Schema Version Guide. Available at: <http://www.fixprotocol.org>.

6. Lu H., Yu J.X., Wang G., Zheng S., Jiang H., Yu G., and Zhou A. What makes the differences: benchmarking XML database implementations. *ACM Trans. Internet Technol.*, 5(1):154–194, 2005.
7. Nicola M., Kogan I., and Schiefer B. An XML transaction processing benchmark. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, New York, NY, USA, 2007, pp. 937–948.
8. Runapongsa K., Patel J.M., Jagadish H.V., Chen Y., and Al-Khalifa S. The Michigan benchmark: towards XML query performance diagnostics. *Inf. Syst.*, 31(2):73–97 (2006).
9. Schmidt A., Waas F., Kersten M.L., Carey M.J., Manolescu I., Busse R. XMark: a benchmark for XML data management. In *Proceedings of the Very Large Database Conference*: 974–985 (2002).
10. Yao B.B., Özsu M.T., and Khandelwal N. XBench Benchmark and Performance Testing of XML DBMSs. *ICDE 2004*:621–633.