XBench – A Family of Benchmarks for XML DBMSs

Benjamin B. Yao[1], M. Tamer Özsu[1] and John Keenleyside[2]

[1] University of Waterloo
School of Computer Science, Waterloo, Ontario, Canada N2L 3G1
{bbyao,tozsu}@uwaterloo.ca
[2] IBM Toronto Laboratory
8200 Warden Avenue, Markham, Ontario, Canada
keenley@ca.ibm.com

Abstract. This paper summarizes the XBench family of benchmarks that are under development at the University of Waterloo. The benchmark identifies various classes of XML databases and applications and proposes a set of benchmarks to accommodate these classes.

1 Introduction

There are a number of benchmarks for XML databases that have recently been proposed. These usually assume a single application, and define the database schema and workload accordingly. These benchmarks are very effective when the database deployment corresponds to this application characterization. However, one could argue that no "canonical" application exists, and therefore a family of benchmarks are needed. In this paper we summarize our work in developing such a family of benchmarks.

2 Database Design

We characterize database applications along two dimensions: application characteristics, and document characteristics. Application characteristics indicate whether the database that the application uses is data-centric or text-centric. In data-centric (DC) applications, the database stores data that are captured in XML even though the original data may not be in XML. Examples include e-commerce catalog data or transactional data that is captured as XML. Text-centric (TC) applications manage actual text documents and use a database of native XML documents. Examples include book collections in a digital library, or news article archives.

In terms of document characteristics, we identify two classes: single document[1] (SD) and multiple document (MD). The single document case covers

[1] "Document", in this context, refers to an XML document, not to a document as defined in the previous paragraph.

those databases, such as an e-commerce catalog, that consists of a single document with complex structures (deep nested elements), while the multiple document case covers those databases that contain a set of XML documents, such as an archive of news documents or transactional data. The result is a requirement for a database generator that can handle four cases: DC/SD, DC/MD, TC/SD, and TC/MD (Figure 1).

	SD	MD
TC	Online dictionaries	News corpus, Digital libraries
DC	E-commerce catalogs	Transactional data

Fig. 1. Classes of XML databases and applications

For the TC/SD and TC/MD classes, we have analyzed a number of databases to statistically characterize them and generalized these to define database schemas for that particular class. In particular, for TC/SD, we analyzed Oxford English Dictionary (OED) [3] and GCIDE [2].

For DC classes, there are not sufficient number of large XML datasets to perform similar analysis. Therefore, our design for TC classes uses TPC-W [4] benchmark. For TC/SD, we "simulate" an e-commerce catalog by defining a database based on the ITEM table along with the AUTHOR, ADDRESS and COUNTRY tables. These are enhanced by two additional tables that do not exist in TPC-W: AUTHOR_2, which includes additional author information such as mailing address, phone, and email, and PUBLISHER, which consists of publisher name, fax, phone, and email address.

DC/MD class consists of transactional data. Therefore, we use the eight basic tables of the TPC-W database and map them to XML documents. In particular, we shred the ORDERS, ORDER_LINE, and CC_XACT tables to create a large number of XML documents.

For actual data generation, we use ToxGene [1], which is a template-based tool facilitating the generation of synthetic XML documents.

3 Workload Design

In keeping with the benchmark design philosophy, we have specified one set of workload for each type of application identified in the previous section. Each set focuses on the particular features of XML documents in that category. Furthermore, we have designed a set of core queries that test a set of core functionality even if the specific formulation of the query may differ for each class of database.

The workload for a given class of database is designed such that, together with the core set, the benchmark covers the use cases that have been specified for XQuery. It is conceivable that at this time some DBMSs may not be able to

process all of these queries. However, we expect that most DBMSs will provide full XQuery support in the near future.

References

1. D. Barbosa, A. Mendelzon, J. Keenleyside, and K. Lyons. ToXGene: An extensible template-based data generator for XML, In *Proceedings of 5th International WebDB Workshop*, pages 49–54, 2002.
2. *GCIDE_XML: The GNU version of The Collaborative International Dictionary of English, presented in the Extensible Markup Language*, Available at http://www.ibiblio.org/webster/, 2002.
3. *Oxford English Dictionary*, Oxford University Press, 1994, Available at http://www.oed.com.
4. Transaction Processing Council. *TPC Benchmark W Specification, Version 1.8*, February 2002, Available at http://www.tpc.org/tpcw/.