

Chapter 4

XML Benchmark

Ke Geng

University of Auckland, New Zealand

Gillian Dobbie

University of Auckland, New Zealand

ABSTRACT

Benchmarks are widely used in database-related research, helping users choose suitable database management systems and helping researchers evaluate their new methods. Recently benchmarks for XML have been designed to support the development of XML tools and systems. In this chapter, XML benchmarks are categorized into four groups: application benchmark, micro benchmark, XML generator and real dataset. Characteristics of each benchmark are discussed and compared. Finally, the future direction of XML benchmarks are discussed.

INTRODUCTION

A benchmark is a standard that enables users to evaluate the performance of a system. A benchmark also helps researchers decide what to evaluate and how to evaluate their system (Gray, 1993). DBMS (database management system) benchmarking is used by researchers and industry practitioners to evaluate products to determine their strengths and weaknesses. In particular, industry practitioners use benchmarks to determine which database system best meets their needs while researchers use benchmarks to evaluate the results of their research. A

benchmark usually consists of a dataset and a set of queries. The dataset may be provided or the user may generate it to meet their own requirements. Users can also choose which queries to execute to fulfill their evaluation. Serious work on benchmarks has been carried out in DBMS-related areas for many years, for the different flavours of database systems, e.g. hierarchical, relational, object-oriented, etc.

XML (eXtensible Markup Language) has attracted attention because of its flexibility and powerful semi-structured data management capability. XML is designed to store, transport and display semi-structured data. Research into XML databases, including both native XML databases and XML-enabled databases, has been carried out and

DOI: 10.4018/978-1-61520-727-5.ch004

XML Benchmark

XML databases have been widely implemented. Traditional DBMS benchmarks, such as the Wisconsin benchmark (Gray,1993) and the TPC series benchmarks (Gray,1993), cannot satisfy the demand of XML DBMSs' implementation and development. In particular, traditional benchmarks cannot satisfy the evaluation of some characteristics, such as hierarchy structure, which exists in XML but is not found in traditional databases. Benchmarks for XML are needed to support the development of XML products and to evaluate the performance of XML-related systems.

Current XML benchmarks can be classified into two main groups: application benchmarks and micro benchmarks. Application benchmarks are designed to evaluate the overall performance of an XMLDB (XML database system). With application benchmarks, users can evaluate and compare XML database systems to identify the main characteristics of the XML database system, such as: whether a system supports data-centric document operations or whether a system has effective methods for dealing with text. These benchmarks are useful for choosing which XML engine/database system meets the users' specific demands.

Micro benchmarks evaluate single operations on XML database systems. Usually, these benchmarks provide single or multiple datasets that imitate possible characteristics, including both structure and value characteristics, which can be found in real data. Individual operations can be executed against the provided datasets and the system's performance can be recorded and evaluated. These benchmarks are important for finding more specific information in XML-research, in areas such as query processing and optimization.

Sometimes, the existing XML benchmarks do not satisfy the demands of users, and so the users build XML documents with particular characteristics for their own research. Several XML generators have been developed to undertake experiments or evaluations that need specific characteristics in

their datasets. An XML generator allows users to set and control characteristics, such as selectivity of elements for experimentation.

Real dataset is another option for evaluating XML-related research. Some organizations publish their data in XML format and these XML documents can be used for proving or demonstrating new methods/algorithms.

In this chapter, application benchmarks, micro benchmarks and XML document generators are introduced. Characteristics for different benchmarks are discussed and compared. This comparison will help users choose which benchmark best fits their needs. We will also introduce three popular real datasets and discuss the future work of XML benchmarks.

APPLICATION BENCHMARK

An application benchmark evaluates the overall performance of a DBMS or query engine. For example, a user may need a database that handles large documents where the data is seldom updated, such as an electronic dictionary. While other people may want a database to store a large number of small XML documents where the data is updated frequently, such as e-commercial transactions. In this situation, application benchmarks may help users to understand the characteristics of the XML databases they evaluate and choose a database system that is suitable for their specific needs. We use the following features to compare application benchmarks:

Document-centric & Data-centric: XML documents can be classified into two groups: document-centric documents and data-centric documents. It is possible that an XML DBMS will perform better on either document-centric documents or data-centric documents, or equally well on both. Similarly some benchmarks concentrate more on one type of document than the other. In order to explore the performance of the XML DBMS and choose the suitable XML DBMS for specific

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/xml-benchmark/41500

Related Content

On the Co-Evolution of SSADM and the UML

Richard J. Botting (2005). *Software Evolution with UML and XML* (pp. 134-151).

www.irma-international.org/chapter/evolution-ssadm-uml/29612

PRAISE: A Software Development Environment to Support Software Evolution

William C. Chu, Chih-Hung Chang, Chih-Wei Lu, Yi-Chun Peng and Don-Lin Yang (2005). *Advances in UML and XML-Based Software Evolution* (pp. 105-140).

www.irma-international.org/chapter/praise-software-development-environment-support/4933

A Survey on JSON Mapping With XML/RDF

Gbéboumé Crédo Charles Adjallah-Kondo and Zongmin Ma (2019). *Emerging Technologies and Applications in Data Processing and Management* (pp. 92-113).

www.irma-international.org/chapter/a-survey-on-json-mapping-with-xmlrdf/230685

Business Processes in UML

Peter Rittgen (2003). *UML and the Unified Process* (pp. 315-331).

www.irma-international.org/chapter/business-processes-uml/30549

From Business Rules to Application Code: Code Generation Patterns for Rule Defined Associations

Jens Dietrich (2009). *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches* (pp. 326-347).

www.irma-international.org/chapter/business-rules-application-code/35865