Which XML Storage for Knowledge and Ontology Systems?

Martin Bukatovič, Aleš Horák, and Adam Rambousek

Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic
{xbukatov,hales,xrambous}@fi.muni.cz

Abstract. New research concerning knowledge and ontology management systems in many cases need the versatility of native XML storage for manipulations with diverse and changing data structures. Within the DEB (Dictionary Editor and Browser) development platform, the efficiency of the background data storage for all kinds of structures and services including dictionaries, wordnet semantic networks, classical ontologies or lexical databases, tends to be a crucial property of the system. In this paper, we describe a large set of tests that were run on four selected (out of twenty) available XML database systems, where the tests were run with the aim to recommend the best XML database for knowledge and ontology storage.

1 Introduction

The main advantages of storing data in the XML format is the data portability across systems and the versatility of the data structures – the storage systems, including query languages for manipulation, can handle arrays, hierarchies, texts, named substructures or links in one defined entity type with the possibility of automatic advanced schemata for syntactic checks of correctness. However, with such properties, the storage systems tend to be less efficient than standard relational databases, when it comes to processing large amounts of data, speaking of sizes in tens of megabytes or more [1, 2].

After 5 years of development of the Dictionary Editor and Browser (DEB) platform that is designed to provide common useful features of dictionary writing systems, there are now more then ten actively used dictionary writing systems and lexicographic projects, which are based on the DEB platform. Two of them, DEBDict [3], general dictionary browser providing access to many dictionaries and lexical resources in several languages, and DEBVisDic [4], wordnet editor and browser used to build more than fifteen wordnets in different languages, are currently in use by more than 700 of registered users from all over the world. The freely available DEB server is currently installed in ten institutions from three continents, where it is used mostly as a XML-based data storage, presentation and manipulation system.

With the current deployment of the DEB platform, the current database storage is not able to efficiently process some kinds of search queries. Thus we have decided to analyze and compare available native XML database systems and provide a recommendation of the best performance for knowledge and ontology systems.

Database systems working with XML data (both native XML databases and XML enabled relational databases) are already widespread and used in many areas. Their performance was benchmarked by many projects using several benchmarks. In [5], a generally applicable benchmark XMach-1 is described and compared to other benchmarks. Results for several databases are presented, showing that native XML databases perform better than XML-enable relational databases. Unfortunately, no database is named, so the results are only general.

Nambiar et al. [6] use XOO7 benchmark to compare several XML-enabled and native XML databases. Their results suggest that XML enabled relational databases process data manipulation queries more efficiently. Native XML databases, on the other hand, are more efficient in navigational queries which rely on the document structure.

Extensive comparison experiments were conducted by Lu et al. [7]. Their results suggest that different XML benchmarks can show different weak and strong points of each database systems.

Differences in the results leads to the conclusion that customized XML benchmarks are needed in addition to a general XML benchmark to fully test the requirements of the application developed. For example for the business XML systems, Nicola, Kogan and Schiefer in [8] offer specialized benchmark, called “Transaction Processing over XML” (TPoX). This benchmark aims to provide good comparison of XML databases suitable for the business process modelling.

In the following sections, we present the results of comparing the XML enabled as well as native XML databases over data commonly used in dictionary writing systems.

2 The DEB Platform for Dictionary Writing Systems

The DEB (Dictionary Editor and Browser, http://deb.fi.muni.cz/) is an open-source software platform for the development of applications for viewing, creating, editing and authoring of electronic and printed dictionaries. The platform is based on the client-server architecture (see the DEB platform schema in Figure 1). Most of the functionality is provided by the server side, and the client side offers (computationally simple) graphical interfaces to users. The client applications communicate with the server using the standard web HTTP protocol.

The server part is built from small, reusable parts, called servlets, which allow a modular composition of all services. Each servlet provides different functionality such as database access, dictionary search, morphological analysis or a connection to corpora.

The overall design of the DEB platform focusses on modularity. The data stored in a DEB server can use any kind of structural database and combine the

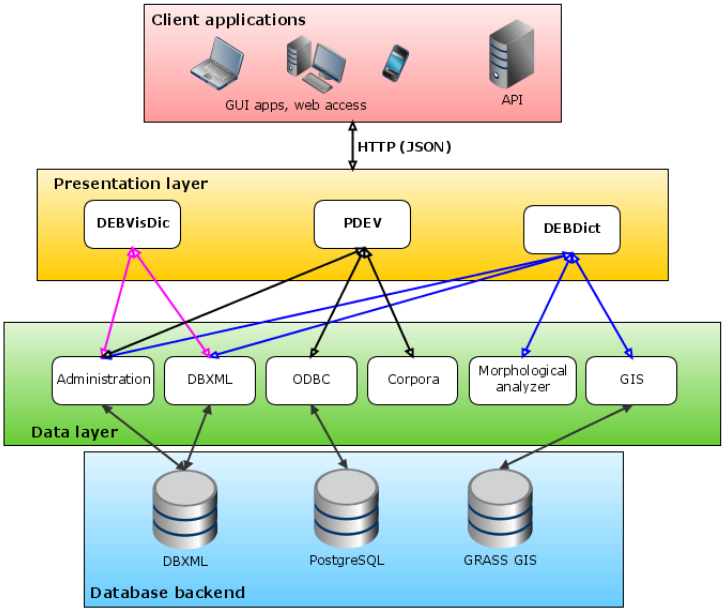

Fig. 1. The DEB platform schema

results in answers to user queries without the need to use specific query languages for each data source. The main data storage is currently provided by the Oracle Berkeley DB XML [1]. However, it is possible to switch to another database backend easily, without any changes to the client parts of the applications.

The main assets of the DEB development platform can be characterized by the following points:

- All the data are stored on the server and a considerable part of the functionality is also implemented on the server, while the client application can be very lightweight.
- Very good tools for team cooperation; data modifications are immediately seen by all the users. The server also provides authentication and authorization tools.
- Server may offer different interfaces using the same data structure. These interfaces can be reused by many client applications.
- Homogeneity of the data structure and presentation. If an administrator commits a change in the data presentation, this change will automatically appear in every instance of the client software.
- Integration with external applications.

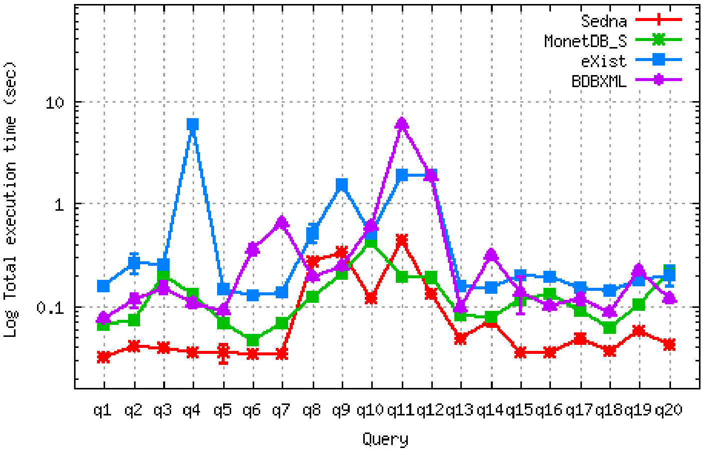

Fig. 2. Total execution time (in seconds) for a 1.8MB document

3 Selected Databases

Although there are many native XML database, we have to select databases that correspond to the licence and technologies applied in the DEB platform. The most important features are the open source licence, active development and support of XML-related standards.

From more than 20 native XML or XML-enabled databases, we have chosen the following four systems according to the designated requirements.

3.1 eXist

The eXist database [9] is developed in Java and licensed under LGPL, active since 2000 and currently developed by the group of independent developers. The database supports XQuery, XSLT and XUpdate standards for data manipulation, and DTD, XML Schema, RelaxNG and Schematron for validation.

Users are able to specify structural indexes (element and attribute structure in documents), range indexes (*contains*, *starts-with* and similar functions), and full-text indexes (Apache Lucene [10] is used for full-text indexing).

3.2 MonetDB/XQuery

The MonetDB/XQuery database [11] is developed by CWI Amsterdam and several Linux distributions and MS Windows are officially supported. The database is licensed under a customised Mozilla Public License.

The main goal of MonetDB is to design a database for processing very large (in GBs) XML documents. The default database settings are optimized for document reading, offering indexing for quick query execution, although the indexes have to be rebuilt after every document update. Another option is an

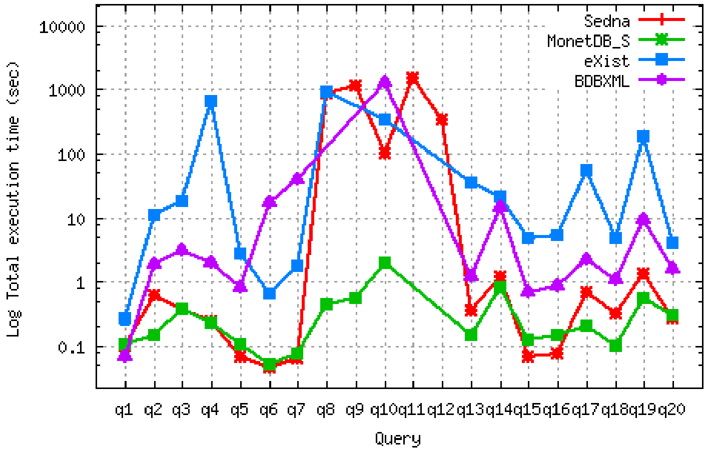

Fig. 3. Total execution time (in seconds) for a 114MB document

optimization for document updating, with simpler index structure and slower performance for search queries.

The database supports XQuery and partly XQuery Update [12]. It is also possible to use MonetDB internal query language. Indexing is automatized, without the possibility to alter settings in any way. The PF/Tijah [13] text search system is utilized for full-text searching.

3.3 Sedna

The Sedna database system [14] is developed by the Russian Academy of Sciences, and released under Apache Licence. Official packages for Windows, Linux, MacOS, FreeBSD and Solaris are available.

The database supports XQuery and custom variant of XQuery Update for data manipulation, and XML Schema for validation. Indexes have to be set manually and a special function must be used in the query to access the index. Full-text indexing is provided by external commercial tool dtSearch. Sedna offers several extensions, such as the capability of an SQL connection from XQuery, or the trigger support.

3.4 Oracle Berkeley DB XML

Berkeley DB XML [1] was created as an extension of Berkeley DB. Currently, the database is developed by Oracle and released for Windows and Linux. Users can choose between open source and commercial licence.

The underlying structure is still based on Berkeley DB and each document container is stored in a single file. The database supports XQuery and part of XQuery Update. The document validation according to a supplied XML Schema

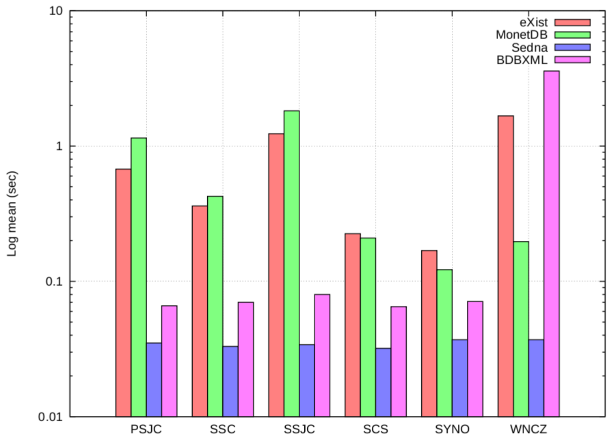

Fig. 4. Average time (in seconds) for the *equality* query

is checked only during document storage, later changes can render the document invalid. Users have to specify indexes manually, full-text indexing is also supported, although it is not possible to use regular expressions in queries.

4 Database Comparison

Because of the special focus on dictionary writing systems, we have decided to run two different test suites. For the general database performance, we have evaluated and selected the XMark benchmark [15], and for the knowledge and ontology test, we have prepared a custom set of the most frequent queries and tasks.

In the tests, we have used the following database versions (preferring stable release over the development one): eXist 1.4.0, MonetDB/XQuery 2009-Aug-SP1, Sedna 3.2.91, and Oracle Berkeley DB XML 2.5.13.

4.1 The XMark Tests

The XMark benchmark was developed in CWI Amsterdam with the aim to provide a benchmark suite for users and developers to choose the right XML database and to tune the database settings.

The benchmark includes the tool `xmlgen` to generate an XML document of a given size. The data and the structure are always the same, users are able to change just the document size. The test suite itself consists of 20 XQuery queries that model different operations with several collections of XML documents, ranging from simple search to complex linking and result generation.

We have tested the queries on documents of size from 1.8MB to 114MB. You can see the results for the smallest and the largest document in Figures 2 and 3.

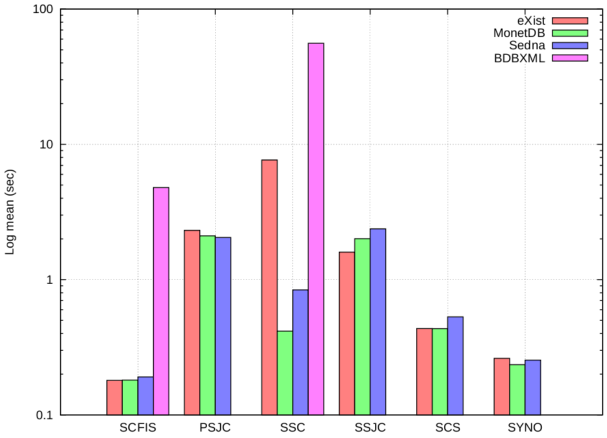

Fig. 5. Average time (in seconds) for the *full-text* query

For the smallest document, all the queries were executed in less than a second, except for some queries in eXist and Berkeley DB XML. The problematic queries `q11` and `q12` are combining data from two collections and building very large result set. Although this is a complex task, it should not take so long on such a small document.

On the other hand, it is understandable in the case of the large document. While increasing the document size, the execution times are getting longer. Fort the 114MB document, much more queries are carried out in times above one second. MonetDB is providing the best results for large documents, and Sedna can be better for less complex queries on smaller documents.

Although the results of XMark tests can help users to pick the right database, real data and tasks should be taken into account, because the results vary significantly according to the document size and query complexity.

4.2 Knowledge and Ontology Data Benchmark

Another step was to test the databases on real data and most frequent tasks of DEB applications. For the benchmark, the following lexicons and ontologies were used:

- The Dictionary of Literary Czech (SSJC), 180.000 entries,
- The Reference Dictionary of Literary Czech (PSJC), 200.000 entries,
- The Dictionary of Written Czech (SSC), 49.000 entries,
- The Dictionary of Words with Foreign Origin (SCS), 46.000 entries,
- The Dictionary of Czech Synonyms (SYNO), 23.000 entries,
- The Dictionary of Czech Phraseologisms and Idioms (SCFIS), 14.000 entries,
- The English WordNet (WNEN), 117.000 entries,
- The Czech WordNet (WNCZ), 28.000 entries.

We have analyzed the operations and have selected the most frequent query types as well as several queries requested by the users.

Equality Query In the first run, XQuery was used to select entries with an element equal to a given value. In the second run, the query was rewritten as an XPath query. With this optimization, databases performed much better, significant improvement was seen for eXist and Berkeley DB XML. The results are shown in Figure 4.

Full-text Search A more or less standard data set for full-text benchmarks is the INEX collection [16]. The current version of INEX collection 2009 contains 2.666.190 semantically annotated Wikipedia articles. The full-text search over the INEX database tests the database performance with a huge amount of data and complex-linked full-text structure. The tested databases have often problems with the kind of data structures used in INEX (e.g. Sedna was not able to build indexes for INEX at all, eXist did not return answers to many queries, MonetDB could not load the databases into 4 GB of memory). However, for the purpose of dictionary applications, the full-text search is usually used within short texts, such as definitions or examples. We thus offer the results of the comparison of full-text search over standard dictionary tags.

For the eXist database, the Lucene module was used for full-text search. We were unable to install PF/Tijah module on the testing server for MonetDB. And for Sedna, the commercial module was not tested. The results are shown in Figure 5.

Considering that full-text modules for MonetDB and Sedna were not used, it is surprising that these databases processed the queries in times comparable to eXist (sometimes even faster). Berkeley DB XML results are missing for most of the dictionaries, because several queries of the test suite were not completed in five minutes.

Document Updates Because DEB platform applications are designed for editing the knowledge and ontology data, the documents are updated by teams of users. Another feature we needed to test is the performance during document deleting and saving. For this test, only the largest dictionary (PSJC) was used. The tests was run several times and each time five random documents were deleted and then saved again.

The average results are shown in Figure 6. Surprisingly, differences between the databases are quite significant.

5 Evaluation

According to the results of the tests, none of the available native XML databases can supersede the others for all kinds of operations needed for knowledge and ontology storage and manipulation. Berkeley DB XML cannot efficiently solve

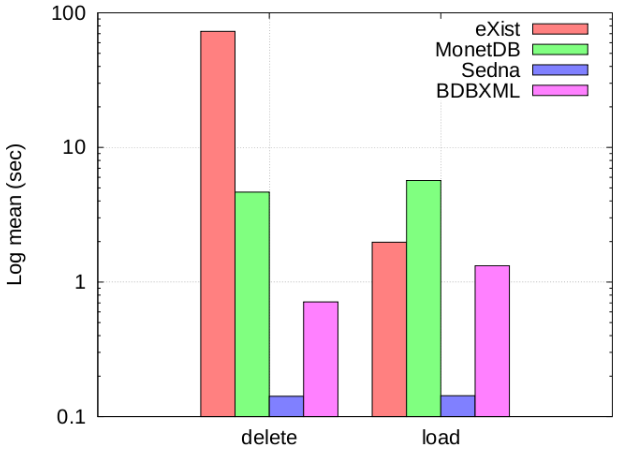

Fig. 6. Average time (in seconds) for document update

the queries involving multiple nodes and full-text queries. The eXist database contains the Lucene module for text search and supports many XML standards, so it can be recommended for deployment where these features are more important than the database performance. On the other hand the MonetDB database can be, according to its specific architecture, conveniently used for when working with very large amounts of XML data. For middle-size data collections, the Sedna database can provide the same performance as MonetDB, while offering richer set of features. The potential drawbacks of Sedna are the need to use special queries for the defined data indexes and the use of commercial tool for optimized full-text queries.[1]

6 Conclusions

Considering the results of XMark and the custom knowledge and ontology benchmark, the MonetDB/XQuery and the Sedna databases represent a good choice for the knowledge and ontology systems. MonetDB offers very good performance for very large documents, on the other hand, Sedna provides much more advanced features. Unfortunately, Sedna supports index usage only with its own special functions, so the queries need to be changed accordingly.

As a next step, both MonetDB and Sedna will be included in the DEB platform and compared in real operation.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National

[1] However, the full-text queries without this optimization are already comparably fast.

Research Programme II project 2C06009 and by the Czech Science Foundation under the projects P401/10/0792 and 102/09/1842.

References

1. Chaudhri, A.B., Rashid, A., Zicari, R., eds.: XML Data Management: Native XML and XML-Enabled Database Systems. Addison Wesley Professional (2003)
2. Krishnamurthy, R., Kaushik, R., Naughton, J.: XML-to-SQL query translation literature: The state of the art and open problems. Lecture notes in computer science (2003) 1–18
3. Horák, A., Pala, K., Rambousek, A., Rychlý, P.: New clients for dictionary writing on the DEB platform. In: DWS 2006: Proceedings of the Fourth International Workshop on Dictionary Writings Systems, Italy, Lexical Computing Ltd., U.K. (2006) 17–23
4. Horák, A., Pala, K., Rambousek, A., Povolný, M.: First version of new client-server wordnet browsing and editing tool. In: Proceedings of the Third International WordNet Conference - GWC 2006, Jeju, South Korea, Masaryk University, Brno (2006) 325–328
5. Böhme, T., Rahm, E.: Multi-user evaluation of XML data management systems with XMach-1. Efficiency and Effectiveness of XML Tools and Techniques and Data Integration over the Web (2008) 148–159
6. Nambiar, U., Lacroix, Z., Bressan, S., Lee, M., Li, Y.: Efficient XML data management: an analysis. E-Commerce and Web Technologies (2002) 261–266
7. Lu, H., Yu, J., Wang, G., Zheng, S., Jiang, H., Yu, G., Zhou, A.: What makes the differences: benchmarking XML database implementations. ACM Transactions on Internet Technology (TOIT) **5**(1) (2005) 154–194
8. Nicola, M., Kogan, I., Schiefer, B.: An XML transaction processing benchmark. In: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM (2007) 937–948
9. Meier, W., et al.: eXist: An open source native XML database. Lecture Notes in Computer Science (2003) 169–183
10. Foundation, A.S.: Apache Lucene (2006) http://lucene.apache.org.
11. Boncz, P., Grust, T., van Keulen, M., Manegold, S., Rittinger, J., Teubner, J.: MonetDB/XQuery: a fast XQuery processor powered by a relational engine. In: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, ACM (2006) 490
12. W3C: XQuery Update Facility 1.0 (2009) (http://www.w3.org/TR/xquery-update-10).
13. Hiemstra, D., Rode, H., van Os, R., Flokstra, J.: PF/Tijah: text search in an XML database system. In: Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR). (2006) 12–17
14. Fomichev, A., Grinev, M., Kuznetsov, S.: Sedna: A Native XML DBMS. Lecture Notes in Computer Science **3831** (2006) 272
15. CWI: XMark – An XML Benchmark Project (2009) http://www.xml-benchmark.org.
16. Schenkel, R., Suchanek, F., Kasneci, G.: YAWN: A semantically annotated Wikipedia XML corpus. In: Datenbanksysteme in Business, Technologie und Web (BTW 2007), Aachen, Germany, Verlagshaus Mainz (2007) 277–291