

MQX: Multi-Query Engine for Compressed XML Data

Xiaoling Wang, Aoying Zhou, Juzhen He
 Department of Computer Science and Engineering
 Fudan University
 Shanghai 200433, China
 {wxling, ayzhou, juzhenhe}@fudan.edu.cn

Wilfred Ng
 Department of Computer Science and Engineering
 Hong Kong University of Science and Technology
 Hong Kong
 wilfred@cse.ust.hk

Categories and Subject Descriptors

H.2.4 [Systems]: Query processing; H.3.3 [Information Systems]: Information filtering.

General Terms

Management, Performance.

Keywords

XML, Multi-Query, Compression.

1. OVERVIEW

XML data are verbose due to their repeated tags and structures. Previous work has studied the techniques for efficient evaluation path expressions on original XML document [1] or single-query processing on compressed data [4] based on the different XML compression methods. However, it is still not clear how to exploit the existing XML compression techniques to evaluate multi-query and optimization strategies in XML subscribe/dissemination applications [2]. In this demonstration, we present MQX which is capable of processing multiple subscribed XPath query over compressed XML data. Different from most information retrieval or filtering systems which return the whole document, MQX obtains the matched XML elements, which is a fragment of the whole document.

To demonstrate the novel features [3] of MQX, we build a cooperative network and implement a content-based subscription system. Assume that there are some co-operation relationships among clients where the server keeps a large number of compressed XML documents, and clients request to obtain information or news from the server in a cooperative way, i.e., each client can send results to others. To prevent the server from becoming a bottleneck, we adopt a distributed approach where all clients participate in the dissemination process. The most distinguishing feature of MQX is that multi-query over compressed XML data can be processed as a whole, resulting in faster data dissemination. Another important feature is that the engine reduces the bandwidth consumption greatly compared to the traditional XML query engines such as SAXON.

MQX engine consists of four main components as shown in Fig.1. XCT facilitates data compression and storage; MQEngine (Multi-Query processor) supports query rewriting, multi-query organization and optimization; DM (Dissemination Manager) sends the results to specific clients; and GUI (the interface unit).

We demonstrate MQX using the datasets XMark [5] and NITF [6]. The queries for NITF are created by YFilter path generator. We will show the efficiency of the MQX by comparing it with SAXON. We will also demonstrate the great benefit of bandwidth consumption reduction by comparing our approach and common processing strategies which return original XML fragments.

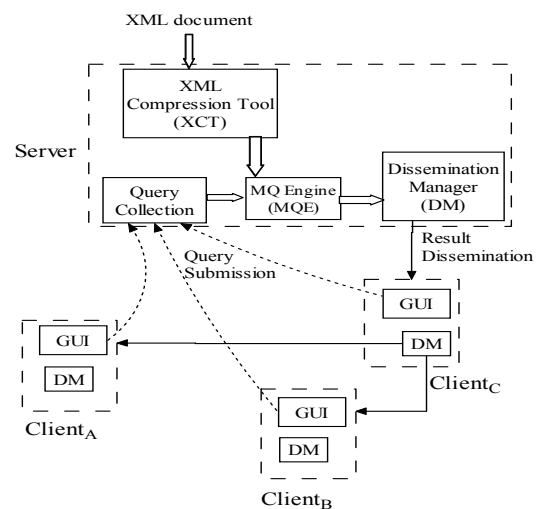


Figure 1 The Architecture of MQX

2. ACKNOWLEDGMENTS

This work is supported by NSFC grants (No.60403019 and No. 60673137) and National Hi-Tech 863 program under grant 2006AA01Z103.

3. REFERENCES

- [1] X. Dong, A.Y. Halevy, I. Tatarinov. *Containment of nested XML queries*, In *Proc. of VLDB*, 2004, 132-143.
- [2] Y. Diao, S. Rizvi, M. J. Franklin. *Towards an internet-scale XML dissemination service*. In *Proc. of VLDB*, 2004, 612-623.
- [3] J. He, W. Ng, X. Wang and A. Zhou. *An efficient cooperative framework for multi-query processing over compressed XML data*. In *Proc. of DASFAA*, 2006, 218-232.
- [4] J. Min, M. Park, and C. Chung. *XPRESS: a queryable compression for XML data*. In *Proc. of ACM SIGMOD*, 2003, 122-133.
- [5] XMark benchmark. <http://www.xml-benchmark.com/>
- [6] NITF dataset. <http://www.nitf.org/index.php>