TIMBER: A Native System for Querying XML

Stelios Paparizos* Shurug Al-Khalifa* Adriane Chapman*
H. V. Jagadish* Laks V. S. Lakshmanan† Andrew Nierman*
Jignesh M. Patel* Divesh Srivastava‡ Nuwee Wiwatwattana*
Yuqing Wu* Cong Yu*

ABSTRACT

XML has become ubiquitous, and XML data has to be managed in databases. The current industry standard is to map XML data into relational tables and store this information in a relational database. Such mappings create both expressive power problems and performance problems.

In the TIMBER [7] project we are exploring the issues involved in storing XML in native format. We believe that the key intellectual contribution of this system is a comprehensive set-at-a-time query processing ability in a native XML store, with all the standard components of relational query processing, including algebraic rewriting and a cost-based optimizer.

1. SYSTEM OVERVIEW

The central idea in the TIMBER [4] project is to construct an XML database that has an overall architecture similar to that of a relational database. In this way, we attempt to reuse existing technologies developed for relational databases, while redesigning and tailoring certain components for the XML domain.

Relational Algebra plays a central role in relational database implementation. An equivalent algebra for XML is developed in TAX [5]. This algebra manipulates sets of ordered labeled trees, rather than sets of tuples. The primary difficulties addressed include the complex and variable structure of trees in a set, and issues of ordering.

Once a query has been parsed into an algebraic representation, this algebraic expression is manipulated by a query optimizer, which then chooses an appropriate query plan to evaluate the query. In [9] we point out some key differences for XML query optimization vis-a-vis relational optimization.

Cost-based query optimization requires good size estimates for both intermediate and final query results. Due to the complex structure of XML data, obtaining these estimates is not necessarily straightforward. We discuss effective techniques for this purpose in [8].

XML query evaluation often involves the computation of a structural join between ancestor/descendant and parent/child nodes. One has to determine the satisfaction of predicates at multiple structurally related nodes in the database. New access methods for the computation of such joins are introduced in [1, 2].

*University of Michigan, contact: {*spapariz, shurug, apchapma, jag, andrewdn, jignesh, nuwee, yuwu, congy*}*@umich.edu*.

Supported in part by NSF, under grants IIS-0208852 and IIS-0219513, and by an IBM equipment grant.

†University of British Columbia, contact: *laks@cs.ubc.ca*

‡AT&T Labs Research, contact: *divesh@research.att.com*

SIGMOD 2003, June 9-12, 2003, San Diego, CA.

2. THE DEMONSTRATION

We will demonstrate our overall system by loading XML documents and allowing users to execute XQueries against them. A graphical representation of each stage of the query evaluation process will be presented. First, the query will be naïvely parsed into our algebra, and subsequently optimized by using both algebraic rewriting and cost-based techniques. After the query plan has been created, the evaluator will construct the matching result trees by performing the appropriate structural joins. Finally, the XML output is returned for the given query.

We will also demonstrate two extensions to the basic query processing functionality, namely probabilistic [6] and structured text querying [3].

3. REFERENCES

[1] S. Al-Khalifa and H. V. Jagadish. Multi-level operator combination in XML query processing. In *Proc. CIKM Conf.*, Nov. 2002.

[2] S. Al-Khalifa, H. V. Jagadish, N. Koudas, J. M. Patel, D. Srivastava, and Y. Wu. Structural joins: A primitive for efficient XML query pattern matching. In *Proc. ICDE Conf.*, Mar. 2002.

[3] S. Al-Khalifa, C. Yu, and H. V. Jagadish. Querying structured text in an XML database. In *Proc. SIGMOD Conf.*, Jun. 2003.

[4] H. V. Jagadish, S. Al-Khalifa, A. Chapman, L. V. S. Lakshmanan, A. Nierman, S. Paparizos, J. M. Patel, D. Srivastava, N. Wiwatwattana, Y. Wu, and C.Yu. TIMBER: A native XML database. *VLDB Journal*, 11(4), 2002.

[5] H. V. Jagadish, L. V. S. Lakshmanan, D. Srivastava, and K. Thompson. TAX: A tree algebra for XML. In *Proc. DBPL Conf.*, Sep. 2001.

[6] A. Nierman and H. V. Jagadish. ProTDB: Probabilistic data in XML. In *Proc. VLDB Conf.*, Aug. 2002.

[7] University of Michigan. The TIMBER project. `http://www.eecs.umich.edu/db/timber`

[8] Y. Wu, J. M. Patel, and H. V. Jagadish. Estimating answer sizes for XML queries. In *Proc. EDBT Conf.*, Mar. 2002.

[9] Y. Wu, J. M. Patel, and H. V. Jagadish. Structural join order selection for XML query optimization. In *Proc. ICDE Conf.*, Mar. 2003.