Rainbow: Multi-XQuery Optimization Using Materialized XML Views

Xin Zhang, Katica Dimitrova, Ling Wang, Maged El Sayed, Brian Murphy, Bradford Pielech, Mukesh Mulchandani, Luping Ding, Elke A. Rundensteiner
Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609
(xinz,katica,lingw,maged,rudie,winners,mukesh,lisading,rundenst)@cs.wpi.edu

Outlook. We present multiple XQuery optimization based on materialized XML view technology in the Rainbow system. In this demo we in particular show: (1) Rainbow's support for defining and incrementally maintaining materialized XQuery views, (2) XQuery optimization by query rewriting to use materialized views, (3) Performing multiple query optimization by merging multiple XML queries (XATs) into one global access plan to decide upon materialization of intermediate results as views, and (4) Query processing of updates issued on XML views that wrap relational data by decomposing the updates into SQL update statements and consistency checks on the relational base data.

The Rainbow System. We have extended Rainbow [1], our existing XML data management system, as shown in Figure 1. Rainbow accepts an XQuery query or an update request in an extended XQuery syntax from the user. The XQuery is parsed into an algebraic representation, called XML Algebra Tree (XAT) [3]. The XAT is then optimized by the global query optimizer using algebraic rewrite rules [2]. We have introduced a separate phase of XAT cleanup [2] which includes the XAT table schema cleanup and cutting of unnecessary XML operators. This optimization often significantly improves the query performance. The optimized XAT is then executed by the query manager.

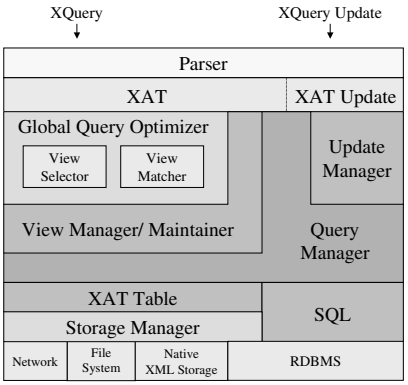

Figure 1: Architecture of the Rainbow System.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGMOD 2003, June 9-12, 2003, San Diego, CA
Copyright 2003 ACM 1-58113-634-X/03/06 ...$5.00.

Unlike the original system, which executes all XML-specific operations that could not be pushed down for execution into the underlying relational engine in main memory, Rainbow now functions in a limited memory environment. This is supported by a storage manager which maps storage requests to the appropriate storage medium and provides support for transparent storage and both value-based and reference-based access to intermediate results.

Rainbow Features. Multiple queries, each represented by an XAT algebra tree, can be merged by the global query optimizer into one global access plan. This then enables the view selector to decide upon the materialization of intermediate results as views. Alternatively, using an extended XQuery syntax, a database administrator can also explicitly specify certain XQuery views to be materialized. Thereafter, the XQuery optimizer would apply query containment matching and query rewriting techniques to make use of materialized views when answering newly incoming user queries.

Our enhanced query processor now not only supports queries but also XQuery update statements directly on the XML views used to wrap the relational data store. Such updates on views are parsed and then handled by the update manager that pushes them down to SQL, if a relational engine was used to store the base data. This includes both update decomposition as well as consistency checking of the to-be-modified base data.

Once the base data has been either directly updated by a base user or indirectly by the above update manager, then the view maintenance process is triggered. The view maintainer is responsible for incrementally refreshing the materialized XML views. Using a special order-encoding mechanism, Rainbow is the first system to support the incremental maintenance of (ordered) XML views defined using the XQuery language. Our approach, being algebraic in nature, efficiently supports not only complex base updates expressed in XQuery but also is easily extensible should future enhancements to the XQuery language itself by made by the community.

1. REFERENCES

[1] X. Zhang, M. Mulchandani, S. Christ, B. Murphy, and E. A. Rundensteiner. Rainbow: Mapping-Driven XQuery Processing System. In *Demo Session Proceedings of SIGMOD'02*, page 614, 2002.

[2] X. Zhang, B. Pielech, and E. A. Rundensteiner. Honey, I Shrunk the XQuery! — An XML Algebra Optimization Approach. In *WIDM*, pages 15–22, Nov. 2002.

[3] X. Zhang and E. A. Rundensteiner. XAT: XML Algebra for the Rainbow System. Technical Report WPI-CS-TR-02-24, Worcester Polytechnic Institute, July 2002.