Chapter 14
Query Performance Optimization in XML Data Warehouses

Hadj Mahboubi
Université de Lyon, France

Jérôme Darmont
Université de Lyon, France

ABSTRACT
XML data warehouses form an interesting basis for decision-support applications that exploit complex data. However, native-XML database management systems (DBMSs) currently bear limited performances and it is necessary to research for ways to optimize them. In this chapter, the authors present two such techniques. First, they propose an XML join index that is specifically adapted to the multidimensional architecture of XML warehouses. It eliminates join operations while preserving the information contained in the original warehouse. Second, the authors present a strategy for selecting XML materialized views by clustering the query workload. To validate these proposals, the authors measure the response time of a set of decision-support XQueries over an XML data warehouse, with and without using their optimization techniques. The authors’ experimental results demonstrate their efficiency, even when queries are complex and data are voluminous.

INTRODUCTION
Decision-support applications aim at facilitating the decision-making process. They collect data from operational databases and various sources, transform them into information available to decision-makers in a consolidated and consistent manner (Kimball & Ross, 2002). Furthermore, the development of the Web 2.0 and the proliferation of multimedia documents contributed to the analysis of data are not only numerical nor symbolic. Indeed, such data can be represented in various formats (databases, texts, images, sounds, videos...); diversely structured (relational databases, XML document repositories...); originating from several different sources (distributed databases, the Web...); described through several channels or points of view (x-ray photographs and audio diagnosis of a physician,
In this context, XML proves a very interesting tool for integrating and warehousing complex data for analysis thanks to its self-description (akin to warehouse metadata) and extensibility features (Darmont et al., 2003). Moreover, XML has become a standard for representing complex business data (Beyer et al., 2005). Hence, many efforts toward XML data warehousing have been achieved in the past few years (Park et al., 2009; Pokorný, 2002).

However, decision-support queries are generally complex because they involve several join and aggregation operations, while most XML-native database management systems (DBMSs) present relatively poor performances when data volume is very large and/or queries are complex.

In classical (i.e., relational) data warehouses, these issues are customarily addressed by indexing data and materializing views (Gupta & Mumick, 2005). Indexes and materialized views are physical data structures that improve data access time. An index allows direct (vs. sequential) access to data, while a materialized view precomputes query results and avoids accessing the whole original data. Both these physical data structures require additional storage space and induce some refreshing process overhead. It is thus crucial to select them wisely.

Several solutions have been proposed for XML data indexing in the literature. However, the existing techniques support single-labeled path expressions within one single XML document (Goldman & Widom, 1997; Chung et al., 2002). Such path expressions help explore an XML document and extract a specific node (element) or sub-tree (subdocument). They cannot perform join operations over several XML documents. In the context of XML data warehouses, decision-support queries are complex and involve several path expressions. Data are also generally distributed into several XML documents due to their large volume. Hence, XML queries should use specific indexes to access these documents.

In the context of relational data warehouses, several studies address the materialized view selection problem (Agrawal et al., 2000; Aouiche et al., 2006). Views that are relevant to materialize are selected to minimize the processing time of a given workload under maintenance cost and/or storage space constraints (Kotidis & Roussopoulos, 1999). Unfortunately, no such view materialization approach exists for XML databases and XML data warehouses in particular.

In this chapter, we propose a new index structure that is specifically adapted to multidimensional XML data warehouses. This structure is able to maintain a star schema of several XML documents and to preserve the information contained in these documents. It is actually a join index that ensures faster execution of decision-support XQueries by eliminating join costs.

Our second contribution consists in adapting Aouiche et al.’s (2006) query clustering-based relational view selection approach to the XML context. We cluster queries and build candidate XML views that can resolve multiple similar queries belonging to the same cluster. Our approach exploits XML-specific cost models to select XML views that are pertinent to materialize.

The remainder of this chapter is organized as follows. We first discuss previous research related to XML indexing and materialized view selection, respectively. Then, we introduce the technical context of our studies, namely the XML data warehouse model we use, before detailing our join index for XML data warehouses and our XML materialized view selection strategy. To validate our proposals, we also present some experimental results. Finally, we conclude this chapter and hint at future research issues.
Related Content

User-Driven Documentation Building for the ERP System
[www.igi-global.com/chapter/user-driven-documentation-building-for-the-erp-system/78777?camid=4v1a](www.igi-global.com/chapter/user-driven-documentation-building-for-the-erp-system/78777?camid=4v1a)

BROOD: Business Rules-driven Object Oriented Design
[www.igi-global.com/chapter/brood-business-rules-driven-object/28636?camid=4v1a](www.igi-global.com/chapter/brood-business-rules-driven-object/28636?camid=4v1a)

Validating the End-User Computing Satisfaction Instrument for Online Shopping Systems
[www.igi-global.com/chapter/validating-end-user-computing-satisfaction/44185?camid=4v1a](www.igi-global.com/chapter/validating-end-user-computing-satisfaction/44185?camid=4v1a)

Open Innovation through Intermediaries in the Web: A Comparative Case Study
[www.igi-global.com/chapter/open-innovation-through-intermediaries-web/60231?camid=4v1a](www.igi-global.com/chapter/open-innovation-through-intermediaries-web/60231?camid=4v1a)