Dissemination of Heterogeneous XML Data

Yuan Ni*
IBM China Research Lab
niyuan@cn.ibm.com

Chee-Yong Chan
National University of Singapore
chancy@comp.nus.edu.sg

ABSTRACT

A lot of recent research has focused on the content-based dissemination of XML data. However, due to the heterogeneous data schemas used by different data publishers even for data in the same domain, an important challenge is how to efficiently and effectively disseminate relevant data to subscribers whose subscriptions might be specified based on schemas that are different from those used by the data publishers. This paper examines the options to resolve this schema heterogeneity problem in XML data dissemination, and proposes a novel paradigm that is based on data rewriting. Our experimental results demonstrate the effectiveness of the data rewriting paradigm and identifies the tradeoffs of the various approaches.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Systems-Query processing

General Terms

Algorithms, Design, Performance

Keywords

data rewriting, dissemination, heterogeneous, XML

1. INTRODUCTION

The ubiquity of XML data and the effectiveness of the content-based pub/sub-based paradigm of delivering relevant information has led to a lot of interest in content-based dissemination of XML data (e.g., [1]). Existing work on XML data dissemination, however, are all implicitly based on a *homogeneous schema* assumption where both the data published by different publishers as well as the users' subscriptions share the same schema. However, since the data publishers in a pub/sub system are autonomous and independent, they generally do not use the same schemas even when their published data are related and belong to the same domain (e.g., product catalogues). Consequently, if a user's subscription is based on the schema of a specific publisher (say P), then while the user can receive relevant documents from P that match his subscription, it is very likely that his subscription will not match relevant data from another publisher P' if the data schemas used by P and P' are different. Thus, the effectiveness of the pub/sub systems in pushing relevant data to consumers becomes diminished in the presence of heterogeneous data schemas.

In this paper, we address the problem of how to improve the effectiveness of XML data dissemination in the presence of heterogenous data schemas. Our problem, referred to as *heterogeneous data dissemination problem*, can be stated as follows. We consider a pub/sub system where data published by different publishers are based on different schemas. The problem is how to effectively disseminate a document (based on some publisher's schema S) to relevant subscribers whose subscriptions might be based on schemas different from S. For simplicity and without loss of generality, we assume that all the published data are of the same domain such that it is possible to use a single global schema to resolve the structural conflicts among different publishers' schemas (of the same domain). Our problem and proposed techniques can be easily extended to the general case by first partitioning the collection of publishers' schemas into groups of schemas with similar domains, and then generating a global schema for each group of related schemas.

The well-known *data integration problem*[3] handles the problem about how to query multiple data sources with different schemas by adopting a query rewritten based approach (QRA). However, a fundamental difference exists between the data integration problem and our heterogeneous data dissemination problem, which is that the integration problem belongs to a single-query-multiple-data scenario while the dissemination problem belongs to a single-data-multiple-queries scenario. To adopt the QRA in the heterogeneous data dissemination problem would incur a scalability problem as each input subscription needs to be rewritten into one subscription for each local schema. This increases the space overhead for storing and indexing the expanded set of local subscriptions at each router since the number of subscriptions on each router is large.

In this paper, we present a novel paradigm to solve the heterogeneous data dissemination problem that is based on the principle of *data rewriting.* We refer to our new approach as *DRA* for data rewriting approach. The conceptual idea of DRA is as follows. Firstly the collection of local schemas from the publishers is integrated to form a global schema S_g which is then made available to users to specify their subscriptions. Unlike QRA, our DRA does not require query rewriting which means that only the input global subscrip-

*The work was done while the author was at the National University of Singapore

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

tions are indexed at each router. For each incoming data D_ℓ (conforming to some local schema S_ℓ) to a router, our DRA rewrites D_ℓ to D_g (D_g may not be materialized here) such that the evaluation of subscriptions is actually conducted against D_g.

In contrast to QRA, our proposed DRA is more effective for the heterogeneous data dissemination problem because pub/sub systems are typically characterized by two properties : (1) the number of subscriptions at each router is large (which limits the scalability of QRA); and (2) the data being disseminated is relatively small (which incurs only a small processing overhead for data rewriting).

2. DATA REWRITING FRAMEWORK

This section presents our framework to solve the heterogeneous data dissemination problem by using *data rewriting*.

2.1 System Architecture

We use S_ℓ to denote some publisher's local schema, and S_g to denote a global schema integrated from a collection of local schemas of the same domain. We use D_ℓ (resp., D_g) to denote a document conforming to schema S_ℓ (resp. S_g).

Similar to existing pub/sub systems, we have a *mediator agent (MA)* that serves as a coordinator between the data publishers and routers [2]. Besides collecting schemas from publishers and registering queries for users, the MA is also responsible for resolving the structural conflicts among various schemas to generate a global schema. The MA creates a *schema mapping* for each local schema S_ℓ that is integrated to a global schema S_g. The schema mapping is essentially a data transformation specification that enables an input document D_ℓ to be mapped into an output document D_g that preserves the appropriate information content of D_ℓ.

2.2 Data Rewriting Approaches

2.2.1 Static Data Rewriting (SDR)

In the *static data rewriting (SDR)* approach, each published data D_ℓ is rewritten to D_g statically (but only once) by the mediator agent (denoted as MA). The advantage of employing the MA to rewrite the data is that the publishers are shielded from the details of the schema mappings and rewriting processing; this requires each publisher to first forward D_ℓ to the MA for the rewriting before the MA forwards the transformed data to the routers for dissemination.

Once D_ℓ has been rewritten to D_g, both D_ℓ and D_g are forwarded together to the network of routers for dissemination. Since the subscriptions stored in each router are based on the global schema S_g, D_g is used for matching against the subscriptions to detect matching subscriptions and decide to which router(s) the data needs to be forwarded next; D_ℓ (possibly with an attached digital signature for verification of data integrity) is forwarded to any matching local subscribers at a router.

One advantage of SDR is that it is a non-intrusive approach that can be easily implemented. However, the tradeoff is that the amount of data that is being forwarded is roughly doubled compared to the conventional approach.

2.2.2 Dynamic Data Rewriting (DDR)

To avoid the transmission overhead of SDR, an alternative strategy is for each router to forward only D_ℓ but the tradeoff is that each router now needs to rewrite the data D_ℓ *dynamically*. We refer to this approach as *dynamic data rewriting (DDR)* approach. Note that DDR does not modify D_ℓ and also does not physically materialize D_g. Instead, the rewriting of D_ℓ to D_g is performed dynamically as D_ℓ is being parsed. Specifically, the parsed events from D_ℓ are used to generate parsed events corresponding to D_g which are matched against the subscriptions, and D_ℓ is then forwarded to any matching routers/subscribers.

We have proposed two dynamic data rewriting approaches based on where the data rewriting is performed.

NDDR. The first option is to perform the rewriting outside of the matching engine by installing a new software component, called the *data rewriter*, between the document parser and matching engine. The data rewriter essentially rewrites D_ℓ to D_g by intercepting the sequence of events E_ℓ that is generated by the event-based XML parser (as it parses the input document D_ℓ) and generating a modified sequence of events E_g to the matching engine such that E_g is equivalent to the sequence of events generated by parsing D_g. We refer to this approach as *non-intrusive dynamic data rewriting (NDDR)* approach since it does not require making any changes to the existing XML parser and matching engine components. The implementation of NDDR requires additional memory to cache some parsed events.

IDDR. The second option is to rewrite the data within the matching engine itself. We refer to this approach as *intrusive dynamic data rewriting (IDDR)* approach as it entails making modifications to the matching engine. The IDDR approach avoids the using of additional memory, however it makes the matching more complicated.

3. DISCUSSIONS

Based on our experimental results, we have the following observations on the efficiency of various approaches. First, SDR does not perform well due to the transmission of additional data, especially when the bandwidth is small or the number of hops to subscribers is large. IDDR does not scale well as the number of subscriptions or the size of documents increases due to its more complicated matching engine. Finally, our experiments show that NDDR overall achieves the best performance. Moreover, the memory space that NDDR incurs for the dynamic data rewriting is small: among the set of documents we experimented, the maximum memory overhead of NDDR is about 32% of the document size, and only three of the documents actually require memory space overhead of over 20% of the document size. For the majority of the documents, the memory space overhead is only around 5% of the document size. Since the size of the documents in data dissemination is usually small, the memory space overhead of NDDR is small which makes NDDR an attractive approach.

4. REFERENCES

[1] C.-Y. Chan, P. Felber, M. Garofalakis, and R. Rastogi. Efficient filtering of XML documents with XPath expressions. *VLDB*, 11(4), 2002.

[2] Y. Diao, S. Rizvi, and M. J. Franklin. Towards an internet-scale XML dissemination service. In *VLDB*, 2004.

[3] I. Manolescu, D. Florescu, and D. Kossmann. Answering XML queries over heterogeneouse data sources. In *VLDB*, 2001.