

Routing XML Queries

Nick Koudas, Michael Rabinovich, Divesh Srivastava
AT&T Labs—Research
{koudas, misha, divesh}@research.att.com

Ting Yu
North Carolina State University
yu@csc.ncsu.edu

1. Introduction

A vast amount of information currently accessible over the Web, and in corporate networks, is stored in databases. Increasingly, these databases join into networked collections and cooperate with each other. One typical example of such cooperation is provided by global companies like AT&T, which have a large number of separate databases serving distinct businesses (e.g., long distance, dial-up Internet access) or business processes (e.g., ordering, provisioning, billing). Although these databases are maintained autonomously, there is frequent need for accessing information across these databases. As another example, various hospitals and health care providers often cooperate and exchange patient data (e.g., medical history) in the course of delivery of health care. In both examples, the resulting collection of cooperating databases can be viewed as a data-sharing peer-to-peer (P2P) network.

As in file-sharing P2P networks, a fundamental problem is that of identifying databases that are relevant to user queries. This problem is referred to as the *location* problem in P2P literature. While research in file-sharing P2P networks resulted in a number of sophisticated solutions to the location problem, these solutions only allow queries by the name (i.e., object id) of the file, or keyword queries, which are not expressive enough for database applications. Furthermore, these solutions (such as Gnutella) are oriented towards finding *any* node containing a query answer (e.g., a requested song), while database applications typically require finding *all* such nodes, and retrieving all answers to a given query. For example, in the health care application, it is essential to retrieve all relevant records of a patient. Locating relevant databases in a data-sharing P2P network can be a difficult task, made even more complicated if this network dynamically changes: new databases appear, database schemas evolve, and databases (like Web sites) disappear.

With the recent emergence of XML as a standard for data exchange, we assume that each database supports access to its data as XML. Indeed, one can represent the contents of relational databases, legacy IMS databases, and formats such as ASN.1, as XML. XML query mechanisms, such as

XPath and XQuery, can then be used for uniformly posing queries against these databases. The common XML standard allows us to abstract away native data representations of each database in the P2P network and formulate the location problem we are addressing as follows:

Given a dynamically evolving collection of XML database servers \mathcal{N} and an XPath query Q , find databases in \mathcal{N} that contain data potentially relevant to query Q .

2. Contributions

Naive solutions to the location problem include identifying each database as relevant (i.e., sending query Q to all database nodes, or “flooding” the network in P2P parlance) or maintaining and using a global XML index of all the data contained in the network to identify relevant databases. Neither solution scales: the former requires each database to process every query, and the latter requires the global index to process every query and every update to data anywhere in the network. Replicating the global index would make it scalable with respect to query processing but not with respect to data updates.

We propose a scalable solution to the location problem in a data-sharing P2P network, consisting of a network of XML database nodes and XML router nodes, and make the following contributions:

- We develop the internal organization and routing protocols for the XML router nodes, to enable scalable XPath query and update processing, under the open and the agreement cooperation models between nodes.
- Since router nodes tend to be memory constrained, we facilitate a space/performance tradeoff by permitting aggregated routing states, and developing algorithms for generating and using such aggregated information.
- We experimentally demonstrate the scalability of our approach, and the performance of our query and update protocols, using a detailed simulation model, varying key design parameters.