Schema Caching for Improved XML Query Processing in P2P Systems

Christos Doulkeridis
Dept.Informatics, AUEB
Athens, Greece
cdoulk@aueb.gr

Kjetil Nørvåg
Dept.Computer Science, NTNU
Trondheim, Norway
noervaag@idi.ntnu.no

Michalis Vazirgiannis*
INRIA FUTURS
Parc Club Orsay Univ., France
mvazirg@aueb.gr

Abstract

The advent and popularity of the World Wide Web (WWW) has enabled access to a variety of semi-structured data and, when available, this data follows some common XML schema. On the other hand the distribution of content has made centralized solutions inappropriate, entering the era of peer-to-peer (P2P) computing, where content is stored in XML databases residing on peers. In this paper, we propose XML schema caching as a summary indexing technique for searching in P2P networks. We study XML query routing in unstructured P2P networks, comparing different search strategies and showing the advantages of our approach in terms of completeness of the search.

1 Introduction

Peer-to-peer computing (P2P) is emerging as an interesting paradigm that can be very suitable for managing and accessing the large amounts of data provided by a high number of entities. P2P also offers a high degree of autonomy and low administration cost, and such systems are intrinsically failure tolerant. In the recent years XML has emerged as the de-facto standard format for information exchange. Assuming data is available on peers in XML format, the challenge is to make it feasible to search and query this data in a P2P context. In this work we will focus on unstructured P2P systems, and present an approach for efficient search and querying of XML documents stored on the peers.

The basic search technique in unstructured P2P networks is *flooding*. However, this technique is not scalable, and in practice imposes a limited query horizon to the search. Several techniques have been proposed for reducing the search cost, and one particular class of techniques is those based on *summary-caching*. This technique aims at caching some summary of remote data, for example using Bloom filters, to assist peer selection during query forwarding. In this paper, we present a new approach based on *caching schemas of XML documents stored at remote nodes*. Assuming a query based on path expression (which is the case for both XPath and XQuery), the schemas can be used for determining peers that contain matching data. Thus the query can be more directed than flooding, as it can be forwarded to peers that are more likely to contain relevant data.

The main contribution of this paper compared to previous related work [1, 2, 4] is the use of cached XML schemas to guide query forwarding in the P2P network and the associated search strategies. Our experiments show that recall roughly doubles compared to flooding.

2 Schema-assisted caching and routing

We consider a system of peers connected in an unstructured (Gnutella-like) P2P network. Data on peers is in the form of XML documents, which can be stored either as files or in a database. Although documents might have a filename, in a system-wide context the file names are not of interest as our approach is data-centric. XML documents may or may not have an associated description using XML Schema or a DTD. We expect for new data-centric XML applications the use of schemas will be the rule, so on each peer a number of schemas are used. In the general case they will not be related to schemas on other peers, but in many cases peers will use standardized schemas, which is necessary in the case of communication and, e.g., e-Business. It should be mentioned that although our approach is most useful when schemas are used, we expect that a certain fraction of documents will still be without an associated schema. In order to improve query efficiency also for queries involving these documents, a "pseudo DTD" can be created that is compliant with the current instance of the document.

Our system-wide query model is based on a subset of XPath. Given a query issued by a user/application on a peer (we denote this peer the *querying peer*), only the XPath part of it is actually forwarded to other peers. The resulting el-

*The work of Dr. Vazirgiannis is supported by the EU Marie Curie Intra-European Fellowship

Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing (P2P'06)
0-7695-2679-9/06 $20.00 © 2006 IEEE

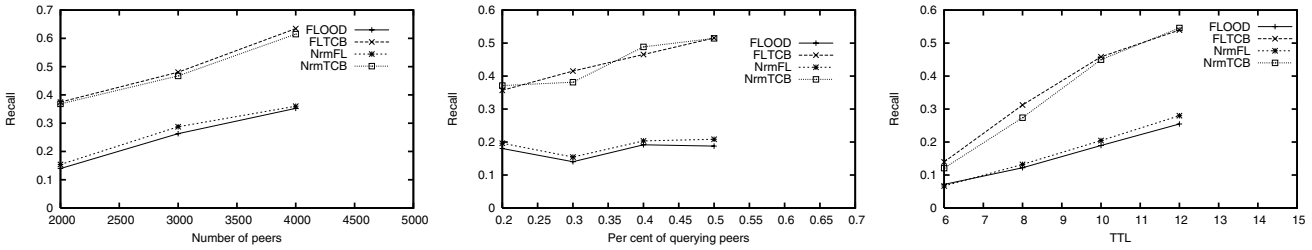

Figure 1. Measurements from using different network sizes, per cent of querying peers and TTL.

ements from applying the XPath expression are returned to the querying peer and they can then be used for further manipulation and query result construction. Instead of applying result caching, we cache the schema of contents located at remote peers, which is returned together with the query results. Such summary information is more compact and robust to changes in actual contents.

When a lookup in the cache (XSCache) for schemas matching the XPath expression gives as result a number of matching schemas and the associated peers, the query is forwarded directly to one of these peers (this forwarding is called a *jump*) in addition to neighbors. Note that unlike routing indices that only maintain information of the neighborhood, information about contents at remote peers (beyond the query horizon) can be contained in the XSCache. In the case when a peer does not find a match in the XSCache, the query is forwarded using the basic query routing algorithm. Note that even if a query match is found at a peer and results returned to the querying peer, the query is further forwarded until the TTL reaches zero.

The XSCache is a fairly traditional cache using one of the traditional cache replacement algorithms. In addition, each item (schema) has an associated expiration time, so that if the validity of the schema with respect to the associated peer has not been confirmed within a certain time, the schema will be removed from the cache.

3 Experimental results and future work

We evaluate the performance of the proposed approach through simulations, in order to test its scalability and feasibility. We use a simulator, created by our group and written in Java, for studying the performance of various routing strategies in unstructured P2P networks. Due to lack of space, we show only results from synthetic topologies with connectivity degree $d = 8$. Resources on peers are described by XML schemas, which may have common parts. We assume a total of $N_S = 500$ schemas in the network. The total number of available resources in the network is $N_R = 50 \times N_P$, where N_P denotes the number of peers. The resource allocation to peers follows the 80/20 rule, i.e., 20% of the peers hold 80% of the resources. Each resource is described by a path expression of the peer's local schema.

At each simulator execution a set of $q_p \times N_P$ ($q_p \leq 1$) peers act as querying peers. Each peer initiates N_Q queries, which are XPath expressions picked from the union of available XML schemas. The queries are randomly generated using a zipfian distribution to simulate the users' interests.

In our simulation study, we assess the different routing strategies using as quality measure the achieved recall. We use as baseline two variants of flooding, naive flooding (FLOOD) and normalized flooding (NrmFL) [3]. For the normalized flooding variants we pick at most 6 neighbors for query forwarding. Our approaches based on flooding and normalized flooding are denoted FLTCB and NrmTCB, in which one (the best) entry is selected from the cache and the remaining $d - 1$ neighbors randomly. It should also be stressed that we compare the performance of all strategies *using the same number of messages* and the same setup parameters, in order to present directly comparable results.

In Figure 1, the scalability of our approach with respect to the number of peers is exhibited. The increasing recall is due to the fact that we also increase the TTL with the network size. Recall also increases with the per cent of querying peers (results for 2000 peers). In a real P2P system all peers are expected to issue requests for resources, so this is another argument in favor of the scalability of our approach. Increasing TTL values also results in higher achieved recall.

Plans for future work include integrating actual XML processors in an implementation to make a P2P XML search engine. We also plan to study in more detail how schemas mediation can best be realized in such a system.

References

[1] I. Brunkhorst, H. Dhraief, A. Kemper, W. Nejdl, and C. Wiesner. Distributed queries and query optimization in schema-based P2P-systems. In *Proceedings of DBISP2P'03*, 2003.

[2] L. Galanis, Y. Wang, S. Jeffery, and D. DeWitt. Processing queries in a large peer-to-peer system. In *Proceedings of CAISE'03*, 2003.

[3] C. Gkantsidis, M. Mihail, and A. Saberi. Hybrid search schemes for unstructured peer-to-peer networks. In *Proceedings of INFOCOM'05*, 2005.

[4] M. Karnstedt, K. Hose, and K.-U. Sattler. Query routing and processing in schema-based P2P systems. In *Proceedings of DEXA workshops 2004*, 2004.