Semantic Query Routing and Processing in P2P Digital Libraries

George Kokkinidis[1], Lefteris Sidirourgos[1], Theodore Dalamagas[2], and Vassilis Christophides[1]

[1] Institute of Computer Science - FORTH
Vassilika Vouton, PO Box 1385, GR 71110, Heraklion, Greece and
Department of Computer Science, University of Crete
GR 71409, Heraklion, Greece
{kokkinid, lsidir, christop}@ics.forth.gr

[2] School of Electr. and Comp. Engineering,
National Technical University of Athens, Greece
{dalamag}@dblab.ece.ntua.gr

Abstract. This paper investigates the peer-to-peer (P2P) resource-sharing paradigm for highly distributed Digital Libraries (DL). The objective is to support decentralized sharing of data and services in a network of autonomous and heterogeneous DL nodes. P2P DLs can operate without a central coordination and offer important advantages such as a very dynamic environment where peers can join and leave the network at any time, while the network can scale up to a large number of peers. The advanced structuring and retrieval functionality of peers poses new challenges in query routing and processing over autonomous, distributed and dynamic networks of DL. The paper considers two fundamental aspects of P2P Digital Libraries (P2P DLs): query routing and processing. Specifically, we design and implement effective and efficient query routing in P2P DLs, exploiting intensional indexing of DL node views. Also, we study interleaved query routing and processing algorithms in P2P DLs to produce as quickly as possible the first query results.

1 Introduction

The digital library community envisions the availability of digital content on a global scale through Digital Libraries (DL) that can be accessed, integrated and individualized for any user, anytime and anywhere. A key point in such a vision is the interaction with multiple DL nodes to support integrated access. We believe that such interaction is far beyond the traditional information integration technologies, which impose restrictions on representation and communication languages used at both the semantic and the structural levels, since:

1. DL nodes should be autonomous. Ideally, a node must not have restrictions on how to organize its data and what kind of query capabilities to offer.

2. DL services should support decentralized sharing and management of data through a network of DL nodes. In such a network, a DL node must be able to provide data to other DL nodes and, at the same time, to have access to data of other DL nodes.
3. The diversity of DL nodes in terms of availability, processing power and interface options, makes a DL network a highly heterogeneous environment in terms of hardware/software setup in addition to the data being provided.
4. Finally, the system needs to be evolving in the sense of DL nodes joining and leaving the network at their own will. DL node arrivals and departures affect the data that is available.

Our work explores the application of the peer-to-peer (P2P) paradigm in DL technologies. In particular, schema-based P2P systems [2, 6] exploit schema information to specify what kind of data is provided by the involved peers. The advantages of this approach lies to the fact that (a) more sophisticated than keyword-based queries can be posed and (b) more efficient approaches can be developed for identifying peers that are capable of answering the queries. A natural candidate for representing descriptive schemas of information resources (ranging from simple structured vocabularies to complex reference models [8]) is the Resource Description Framework/Schema Language (RDF/S). RDF schemas offer rich semantics. The primitives of RDF schemas are classes and properties. Classes describe general concepts or entities. Properties describe the characteristics of classes or the relationships between classes.

RDF/S (a) enables a *modular design* of descriptive schemas based on the mechanism of *namespaces*; (b) allows easy *reuse* or *refinement* of existing schemas through *subsumption* of both class and property definitions; (c) supports partial descriptions since *properties* associated with a resource are by default *optional and repeated* and (d) permits *super-imposed descriptions* in the sense that a resource may be multiply classified under several classes from one or several schemas. These modelling primitives are crucial for schema-based P2P systems where monolithic RDF/S schemas and resource descriptions cannot be constructed in advance and DL nodes may have only incomplete descriptions about the available resources.

The advanced structuring and retrieval functionality of schema-based P2P systems raises new challenges for view integration, query routing and processing over autonomous, distributed and dynamic networks of DLs.

The main contributions of our work presented in this paper are (a) the design and implementation of effective and efficient query routing in P2P DLs, exploiting intensional indexing of DL node views and (b) the study of interleaved query routing and processing algorithms in P2P DLs in order to produce as quickly as possible the first query results.

1.1 Related Work

Several projects address query processing issues in general P2P systems [9, 7]. However, they require a priori knowledge of the relevant to a query peers. Mutant Query Plans (MQPs) [10] implement efficient query routing. Unlike our

approach, MQP reduces the optimization opportunities by simply migrating possibly big XML fragments of query plans along with partial results of subqueries. In [11] indices are used to identify peers that can handle containment queries (e.g., in XML). However there are no details on how a set of semantically related peers can actually execute a complex query involving vertical and horizontal distribution. RDFPeers [12] is a scalable distributed RDF/S repository which efficiently answers multi-attribute and range queries. This approach ignores RDF/S schema information during query routing, while distributed query processing and execution policies are not addressed. In [13], a P2P architecture is introduced, based on the extension of an existing RDF/S store. Although schema information is used for indexing, RDF/S class and property subsumption is not considered. A schema-based P2P infrastructure for the Semantic Web is described in [6]. Their approach involves exact matching of basic class and property pattern and does not consider run-time adaptability of query plans.

2 P2P Digital Libraries

In order to design an efficient P2P DL infrastucture we need to address the following issues: (a) How DL nodes advertise their bases?, (b) How DL nodes formulate queries?, (c) How DL nodes route queries?, and (d) How DL nodes process queries?

2.1 Advertisements of DL Nodes

A schema-based P2P DL infrastructure requires that each DL node advertises its local base content to other DL nodes. Using these advertisements, a DL node becomes aware of the bases hosted by other nodes in the DL. In our approach, we assume that there are global RDF/s schemas for various communities, in which DL nodes have access through the mechanism of namespaces. However, a global RDF/S schema may contain numerous classes and properties not necessarily populated in a DL node. Therefore, we need a fine-grained definition of schema-based advertisements. We employ *RVL views* [5] to specify the subset of a community RDF/S schema(s) for which all classes and properties are populated in a DL node base. These views may be broadcasted to (or requested by) other DL nodes, thus informing the rest of the P2P DL of the information actually available in the DL nodes.

2.2 Query Formulation in DL Nodes

In this work, queries and views in a P2P DL are formulated by nodes in the RQL/RVL [4, 5] language. RQL is a typed functional language in the form of OQL. It uniformly queries both RDF data descriptions and schemas. RVL extends RQL by supporting views on RDF/S. In RQL/RVL, class and property path patterns allow users to navigate through the RDF/S schema of a DL node to retrieve resources. RQL queries allow us to retrieve the contents of any DL

node base, namely resources classified under schema classes or associated to other resources using schema properties. It is worth noticing that RQL queries imply both intensional (i.e., schema) and extensional (i.e., data) filtering conditions.

2.3 Query Routing in P2P Digital Libraries

Query routing is responsible for finding the relevant to a query DL nodes (or more precisely their views) by taking into account data distribution (vertical, horizontal and mixed) of their bases committing to an RDF schema. The query-routing algorithm takes as input a query and the available DL node views and detects which DL nodes can actually answer the query as a whole or fragments of it. The latter is important, since there might be answers that can be received by joining partial answers from different DL nodes. Our approach exploits query/view subsumption algorithms [1] to check whether the classes or properties of the view are subsumed by the respective classes or properties used in the query. In this way, query routing takes into consideration semantic information from the RDF Schemas of the involved DL nodes.

Specifically, a fragmentor breaks the given query into subqueries, whose number is bounded by an input variable. The query/view subsumption algorithms of [1] are employed to determine which part of a query can be answered by a DL node view. For maintaining a distributed catalog of views published by the DL nodes in a P2P DL, appropriate DHT structures have been designed.

2.4 Query Processing in P2P Digital Libraries

Query processing is responsible for generating query plans according to the results returned by the routing algorithm (i.e., which DL nodes can actually answer the query as a whole or fragments of it). If more than one DL nodes can answer the same query fragment, the results from each such DL node base are "unioned" (horizontal distribution). The results obtained for different query fragments that are connected at a specific domain or range class are "joined" (vertical distribution). The generated query plan reflects the data distribution of the system and uses it for obtaining at execution time both complete and correct results.

The resulting query plan can be optimized. Compile-time optimization relies on algebraic equivalences (e.g., distribution of joins and unions) and heuristics allowing us to push, as much as, possible query evaluation to the same DL nodes. Additionally, cost-based optimizations based on statistics about the DL node bases enable to reorder joins and choose between different execution policies for the query plans (e.g., data versus query shipping).

A key feature of our approach is that *query routing and processing are interleaved* in several iteration steps. This leads to the creation and execution of multiple query (sub)plans that when "unioned" offer completeness in the results. Specifically, starting with the initial query, at each iteration step, smaller subqueries are considered in order to find the relevant DL nodes (i.e., routing) that can actually answer them (i.e., processing). The routing information, i.e., remote DL node views, is acquired by the lookup service offered by the system on top of

intensional DHT structures. The interleaved query evaluation terminates when the initial query is decomposed into its basic class and property patterns.

The main advantage of the interleaved query routing and processing algorithm is that the query results are collected as quickly as possible since they require fewer intra-DL node joins. More precisely, each query fragment is looked up as a whole and only DL nodes that can fully answer it are actually involved in each query processing iteration step.

References

1. Christophides V, Karvounarakis G, Koffina I, Kokkinidis G, Magkanaraki A, Plexousakis D, Serfiotis G, Tannen V (2003) The ICS-FORTH SWIM: A Powerful Semantic Web Integration Middleware. In Proc. of the 1st Int'nal Workshop on Semantic Web and Databases (SWDB), Berlin, Germany.
2. Halevy AY, Ives ZG, Mork P, Tatarinov I (2003) Piazza: Data Management Infrastructure for Semantic Web Applications. In Proc. of the 12th Int'nal World Wide Web Conf. (WWW).
3. Ives ZG (2002) Efficient Query Processing for Data Integration. phD Thesis, University of Washington.
4. Karvounarakis G, Alexaki S, Christophides V, Plexousakis D, Scholl M (2002) RQL: A Declarative Query Language for RDF. In Proc. of the 11th Int'nal World Wide Web Conf. (WWW), Honolulu, Hawaii, USA.
5. Magkanaraki A, Tannen V, Christophides V, Plexousakis D (2003) Viewing the Semantic Web Through RVL Lenses. In Proc. of the 2nd Int'nal Semantic Web Conf. (ISWC).
6. Nejdl W, Wolpers M, Siberski W, Schmitz C, Schlosser M, Brunkhorst I, Loser A (2003) Super-Peer-Based Routing and Clustering Strategies for RDF-Based P2P Networks. In Proc. of the 12th Int'nal World Wide Web Conf. (WWW), Hungary.
7. Sahuguet A (2002) ubQL: A Distributed Query Language to Program Distributed Query Systems. phD Thesis, University of Pennsylvania.
8. Magkanaraki A, Alexaki S, Christophides V, Plexousakis D (2002) Benchmarking RDF Schemas for the Semantic Web. In Proc. of the 1st Int'nal Semantic Web Conf. (ISWC'02).
9. Kemper A, Wiesner C (2001) HyperQueries: Dynamic Distributed Query Processing on the Internet. In Proc. of the Int'nal Conf. on Very Large Data Bases (VLDB), Rome, Italy.
10. Papadimos V, Maier D, Tufte K (2003) Distributed Query Processing and Catalogs for P2P Systems. In Proc. of the 2003 CIDR Conf.
11. Galanis L, Wang Y, Jeffery SR, DeWitt DJ (2003) Processing Queries in a Large P2P System. In Proc. of the 15th Int'nal Conf. on Advanced Information Systems Engineering (CAiSE).
12. Cai M, Frank M (2004) RDFPeers: A Scalable Distributed RDF Repository based on A Structured Peer-to-Peer Network. In Proc. of the 13th Int'nal World Wide Web Conf. (WWW), New York.
13. Stuckenschmidt H, Vdovjak R, Houben G, Broekstra J (2004) Index Structures and Algorithms for Querying Distributed RDF Repositories. In Proc. of the Int'nal World Wide Web Conf. (WWW), New York, USA.