XRPC: efficient distributed query processing on heterogeneous XQuery engines
Zhang, Y.

Citation for published version (APA):
Zhang, Y. (2010). XRPC: efficient distributed query processing on heterogeneous XQuery engines

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
While P2P applications that provide trivial keywords search and file sharing features (such as Kazaa, eDonkey) have gained enormous popularity in a short time, the development of P2P applications that provide complex distributed data management and querying facilities advances only slowly. This is because that the development of such applications is still a highly cumbersome task, as applications have to deal with information from different data sources. In P2P settings this set of data sources is extremely dynamic and has an enormously large scale, thus foreseeing all possible combinations of available data sources is impractical. This puts a high adaptivity burden on the shoulders of the application programmers.

To ease the development of data-intensive P2P applications, we envision a P2P XDBMS that acts as a database middle-ware system. It manages dynamic collections of heterogeneous XML data sources (i.e., peers with different software installed) and provides a uniform database abstraction to the application. The ultimate goal is to research which features such a database abstraction should offer, and how it can be realized efficiently by extending and combining existing XDBMS systems with P2P technologies.

In our quest for creating P2P XDBMS technology, we first focus on Distributed XDBMS technology, as this area also was unexplored, with an extra requirement that the to be developed technology will serve as a building block for P2P XDBMS technology. The distinction between Distributed and P2P technology is that in the former, users (i.e., application programmers) are aware of on which sites (i.e., peers) data are located. Distributed queries typically involve specific and explicit locations where data are to be queried from. In P2P systems that mainly target large environments where users cannot keep track which data is on which peer and where the group membership is highly volatile (peers enter and leave continuously and unpredictably), users are typically shielded from explicit knowledge where data is located.

In this thesis we have looked into different aspects of Distributed XDBMS including query execution, query optimisation and transaction management. The result of this work is XRPC, a minimal but orthogonal XQuery extension that enables efficient distributed querying of heterogeneous XQuery data sources. XRPC allows any XQuery expressions including the XQUF expressions to be included in a function body and executed on an arbitrary number of (remote) peers using an RPC mechanism. The main design and implementation criteria of XRPC are imposed by the targeted P2P environments: interoperability, efficiency and scalability.

First, the thesis gives a formal definition of the syntax and the semantics of XRPC including the semantics of distributed updates that follow from the use of XQUF updating functions over XRPC. This includes the definitions of two isolation levels for read-only and updating XRPC queries. The experiences in MonetDB/XQuery suggest that adding XRPC to existing XDBMS is easy, as shredding, serialisation and HTTP functionality are usually already present. The work is limited to a small parser extension and stub code generation. Since interoperability is a major goal, XRPC also comprises a SOAP-based network communication protocol SOAP XRPC. Such a SOAP protocol has the additional advantage of seamless integration with web services and AJAX-based GUIs. The SOAP XRPC protocol supports the concept of Bulk RPC, i.e., the execution of multiple function calls can be handled in a
SUMMARY

single message exchange. This amortises network and parsing latencies and can make XRPC a quite efficient communication mechanism. This thesis shows that the loop-lifting technique, which is pervasively applied in the MonetDB/XQuery system for the translation of XQuery expressions to relational algebra, can easily generate such Bulk RPC requests.

Then, the thesis discusses various aspects of using XRPC in distributed XQuery processing in Chapter 4. First, it shows that XRPC can be easily adopted by different XQuery engines, such that complex P2P communication patterns can be programmed using XRPC. To enhance adoption of XRPC, an XRPC Wrapper is described that allows any XQuery data source to handle XRPC calls. The experiments on Saxon show that Bulk RPC enables set-oriented optimisations, such that Bulk RPC execution of a selection function can be handled using a join strategy. To better match the transaction semantics in databases, a deterministic update semantics for XQUF queries is defined and the SOAP XRPC protocol is extended to guarantee a deterministic order in distributed update scenarios. Atomic distributed commit is supported by using a SOAP-based 2PC protocol defined by the industry standard Web Services Atomic Transaction.

Decomposing queries to address multiple data sources is a well-studied optimisation technique in relational, object-oriented and semi-structured databases. While many of the existing techniques can be carried over, the XML data model and the XQuery language introduce a number of particular challenges not met elsewhere that revolve around XML node identities and structural (rather than value-based) relationships between nodes. In Chapter 5, the thesis elaborates a framework for distributed execution of full-fledged XQuery (i.e., including XQUF), focusing on the issue of providing deep-equal query decompositions, in the face of semantic differences when (parts of) nodes are shipped across the network in XML messages. The thesis proposes a series of decomposition techniques such as pass-by-projection and the use of a novel runtime XML projection method for serialising XML messages, that remove virtually all semantic problems and strongly improve performance. The thesis also defines the semantics of updating both local and remote documents using XQUF expressions and additional constraints that should be added to the proposed techniques to guarantee semantic equivalence for such queries. The correctness of all proposed algorithms is formally proven in Chapter 6.

In this thesis we have also taken a first step towards creating powerful P2P XDBMS technology that preserves the full XQuery language (+XQUF) by extending it only with a single new construct, i.e., XRPC. The thesis proposes MonetDB/XQuery\* in which DHTs can be integrated with XDBMS by adding support for a new dht:// protocol in URLs. Thus no further XQuery extensions are required. The thesis discusses the semantics of two ways of coupling (loose and tight) a DHT with an XDBMS, of which the latter is more complex but powerful.

The XRPC remote function execution mechanism and the ideas of MonetDB/XQuery\* are applied in a P2P Information Retrieval application called StreetTiVo. StreetTiVo enables near real-time search in video contents by distributed and parallel execution of compute-intensive video analysis tasks on multiple peers. Our work on the StreetTiVo application confirms the assumption that a P2P middle-ware DBMS could ease the development of data-intensive P2P applications. With XRPC the rather complex functionalities of StreetTiVo were quickly implemented using just a handful of XQuery functions which in turn are executed on the participating machines.