Optimization of XPath queries over a network

Yoshiki FUKUI (210078)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 13, 2004

Keywords: XML, XPath, Query Processing, Distributed Database, Optimization.

Recently, XML is regarded as a standard data format for information exchange and dissemination over the internet. As a result of that, XML data is distributed over a network, therefore a good method of retrieving XML data over a network is becoming more important these days and there are many researches on various style of XML information system. Most of those systems use some kind of query languages. Some of them use their own query languages, but recently a language called XPath has become very popular. When a XPath query is applied to a XML data, it returns a set of subtrees rooted by nodes that match to the tree pattern.

When a client issues a set of XPath queries to a XML database over a network, a set of answer sets that are sent back to the client may include redundancy in two ways: some elements in one answer set may also appear in other answers, and some elements in one answer set may be subelements of elements in other answer. Therefore, sending those answer sets separately to the client over a network is not optimal with respect to the communication cost. Even when a client issues only one query, the answer can have a self-redundancy because some elements may be subelements of other elements in that answer. To solve this problem, we proposed a method of minimizing the communication cost in XPath processing over a network. In that approach, a set of original queries are transformed into a set of queries corresponding to a minimal size view set that can answer to all the original queries. The server sends this view set to the client, and the client produces answer from it.

Although that approach reduces the communication cost, in some cases, it increases the sever side computation cost instead. It is because the transformed XPath queries corresponding to the minimal size view set is more complex than the original queries. For example, queries corresponding to a minimal view set usually include many negations and set intersection operations in order to remove redundancy. Transformed queries become

complex especially when an answer to the original query include self-redundancy, and the transformed queries have to include operations to remove the self-redundancy. To solve this problem, we propose a new strategy of the query transformation that produces queries which can be evaluated more efficiently.

In addition to the computation cost at the servers, the computation cost at the clients also increases in our framework because the clients need some additional processing to extract the answer to the original queries from the minimal views. When the clients are machines with a limited computation power, such as PDAs or cellular phones, this additional processing can be infeasible. To reduce the cost at the client side, we propose two approaches. The first approach is to add some additional information to the data sent to the client so that the client can extract the answer by easier queries. The second approach is to send a list of byte offsets indicating the positions from which the client should extract the answer. In the latter approach, the client can extract the answer even without parsing the given XML data, whose cost is an important factor of the processing cost at the client side.

We conducted experiments to evaluate these approaches, and our experimental results show that our approach can reduce the computation cost without much increase of the communication cost.