Efficient Query Subscription Processing in a Multicast Environment *
Extended Abstract

Arturo Crespo, Orkut Buyukkokten, and Hector Garcia-Molina
Stanford University

With information dissemination (information push), data is delivered from a set of *producers* to a (typically) larger set of *consumers*. Examples of dissemination-based applications include information feeds (e.g., stock and sports tickers), traffic information systems, etc. We focus on dissemination systems where the consumers in advance submit *subscriptions* defining their interests. Each subscription may include one or more queries over the data that the producers hold or generate. The producers run the queries periodically, disseminating information of specific interest to the consumers.

In a multicast network, subscriptions can be combined to improve efficiency. For example, consider the case where n consumers issued exactly the same query in their subscriptions. A multicast-based service can establish a "channel" for the answer and transmit the answer only once.

However, in many applications, it is unlikely that a large number of consumers will issue exactly the same query, preventing us to fully exploit the advantages of a multicast network. In [1] we have studied algorithms for merging not only identical queries but also queries with answers that overlap significantly. By merging these queries, the producer has to process fewer queries and the amount of information sent may be reduced. On the negative side, the merged answers may contain some data that is irrelevant to a consumer. As a result, the consumer needs to apply a post-filtering *extraction query* over the received data to obtain the answer to its original query. For example, say we merge queries $q_1 : \sigma_{2 \leq A \leq 40} R(A)$ and $q_2 : \sigma_{3 \leq A \leq 41} R(A)$ into $q_3 : \sigma_{2 \leq A \leq 41} R(A)$. The producer can then process this single query and send the result, $ans(q_3)$, to the consumers that issued q_1 and q_2. The q_1 consumer will need to *extract* the q_1 answer from $ans(q_3)$ by applying the extraction query $q_1 : \sigma_{A \leq 40}(q_3)$. Similarly, the q_2 consumer applies its own extraction query to eliminate all elements less than 3.

The goal of the *Query Merging* (QM) problem is to reduce the cost of answering a set of query subscriptions made by consumers to a producer. The solution identifies a (possibly) different set of queries, with lower processing and transmission costs, from which the consumers can derive the answers to their original queries.

Specifically, the producer receives a set of queries Q and outputs a set $\mathcal{M}$ where each of its elements is a set of queries to be merged. The QM problem is to find the set $\mathcal{M}$ with the minimum cost. The input for the problem is a cost function $cost()$, a merge procedure $mrg()$, and a set of queries Q. The output is a collection $\mathcal{M}$ such that the total cost, $cost(\mathcal{M})$, is minimized.

The cost of processing the queries and sending the answers back is represented by the total resources consumed by the producer, the network, and the consumers. The costs involved in our model are: (i) producer cost to run the merging algorithm and to process the merged queries; (ii) cost of transmitting the answers of the merged queries; (iii) consumer cost of applying the extraction procedure.

There are solutions to the QM problem that are polynomial when $|Q| \leq 2$. Unfortunately, we have proved that in the general case ($|Q| > 2$) the QM problem is NP-hard, with doubly exponential complexity on the number of queries (under certain conditions $O(n^n)$ complexities are possible). This high complexity order makes exhaustive algorithms impractical. Therefore, we developed three heuristic algorithms to the QM problem: the Pair Merging Algorithm, the Directed Search Algorithm and the Clustering Algorithm. To experimentally evaluate the performance of the algorithms developed, we implemented a simulator that uses selection queries as a representative example.

In conclusion, in the extended version of this paper, we study the QM Problem. We develop a very general framework and cost model for evaluating merging, and we create a variety of merging algorithms. To illustrate and experimentally evaluate performance, we consider selection queries as a representative example. Our results show that dissemination costs can be significantly decreased by using a merging algorithm, and that our heuristic algorithms work well.

*This work was partially supported by DARPA, as part of the BADD Technical Enhancements Program and the NASA Interchange for Joint Research NCC 2-5347.

[1] A. Crespo, O. Buyukkokten, and H. Garcia-Molina. Efficient query subscription processing in a multicast environment. Technical Report (in www-db.stanford.edu/pub/papers-/badd.ps), Stanford University, 1999.