Distributed Fragmented Database Set Queries

Clive C. Sanford
University of North Texas

Robert E. Markland
University of South Carolina

Patrick R. Philipoom
University of South Carolina

Kenneth Darby-Dowman
The Polytechnic of Central London

This study investigates the distributed multiple condition set query optimization problem for a single relational database with horizontally partitioned data. A mixed integer linear programming formulation constructs a minimum cost query graph which ships combined and/or uncombined condition sets to a subset of the network sites which contain the partitioned data. Two heuristics of different computational complexity are evaluated against a plant location lower bounding procedure. Parameters central to the design of the heuristics are the ranges of (1) the number of network nodes, (2) the arc unit transmission costs, (3) the amount of data at the network sites, and (4) the selectivity factors between the partitioned data. The results of a full factorial experiment evaluated over these range sets are presented.

Introduction

Advances in communication technology allow organizations to distribute their processing requirements across decentralized departments and divisions and promote the strengthening of strategic alliances. These distributed communication networks enable firms to capitalize on strategic opportunities through on-line links to customers, agents, distributors, retailers, and suppliers. For example, the installation of terminals in clinics and hospitals by American Hospital Supply was instrumental to its success at preempting competing hospital suppliers. The American and United Airlines reservation system also exemplifies how a competitive advantage can be created by using a distributed information system (Copeland and McKenney, 1988).

Distributed database management systems (DDBMSs) are the means by which the end user obtains access to distributed data. The data is usually distributed according to the degree of managerial control required at remote sites and/or according to the frequency of use. This type of distribution gives organizations the opportunity to add, modify, and delete their data interactively on a functional level with increased managerial control and a higher degree of accuracy. An important function of DDBMSs is to aid users in forming integrated models of the database. These models are an integral part of query processing, or the ability to access, retrieve, and transform the data into information.

Query processing in a distributed database system corresponds to the translation of user requests, formulated in a high level language on one computer in the network into a sequence of elementary instructions which retrieve
the data. An important goal of distributed query optimization procedures is to minimize the response times and/or the communication costs incurred when processing and answering queries. In this study, the data is organized in a relational database (Codd, 1970, 1971; Date, 1981; and Ullman, 1982) which is horizontally partitioned in order to accommodate the organizational distribution requirements. Distributing the data creates the need for communication between the network sites which contain the data. Non-procedural relational database languages allow users to construct ad hoc queries interactively, and this creates a need to formulate efficient algorithms which generate cost effective sequences of relational operations and data transmissions.

Prior research in query optimization for distributed relational data has focused on procedures which minimize data transmission costs in a non-partitioned environment. Models that attempt to characterize and solve this problem have been formulated by Hevner and Yao (1979), Cheung (1982), Yu, Ozsoyoglu, and Lam (1984), and others. These studies differ substantially in their methodology and depend on the assumptions made and the structure of the objective function.

This study investigates the multiple condition set query problem and considers different strategies for combining the condition sets prior to their transmission to the sites with the partitioned data. We begin with a discussion of the problem definition and examples of how this problem applies to real world situations. This general overview is then followed by a brief description of the relational algebra which is used when answering set queries. We also describe the data transmission cost function and its use in modeling the cost function and the structure of the objective function.

This study investigates the multiple condition set query problem and considers different strategies for combining the condition sets prior to their transmission to the sites with the partitioned data. We begin with a discussion of the problem definition and examples of how this problem applies to real world situations. This general overview is then followed by a brief description of the relational algebra which is used when answering set queries. We also describe the data transmission cost function and its use in modeling the arc costs in query graphs. We conclude this section with an example set query problem and a discussion of the enumeration of the potential transmission strategies for the condition sets. A mixed integer linear programming formulation for the multiple condition set query problem is then presented, and due to its computational complexity, we present the computational results of a full factorial experiment on two heuristics. These results are used to determine the quality of the solutions and possible implementation in a distributed environment. Finally, we conclude with a brief summary and a discussion of potential directions for future research.

**Problem Definition and Applications**

A **General Overview**

A single condition set query is represented by a sequence of relational operations preceded and/or followed by set operations (e.g., set intersection) between the condition sets and the fragments (Gavish and Segev, 1986). A general expression for set operations is \([\text{target set} \cup \text{condition set}]\), where the target and condition sets are horizontally partitioned relational tuples which form the variable and constant operands of a set operation. Fragments are the union of the disjoint target sets at a network site and are geographically dispersed across the network. The relational operations generate the view of the *initial* condition set, which is a subset of a fragment. Multiple condition set queries generate geographically dispersed views of the initial condition sets.

**Example 1**

Consider a distributed database system for a hypothetical airline which contains the number of unsold seats on each departing flight. Management notices that there are an inordinate number of unticketed departures from New Orleans, and is interested in finding out if this has any effect on the number of ticketed departures at the destinations. A database query could be constructed which asks the question:

“List the terminals and their flight numbers which had more unticketed seats on departure flights than New Orleans on June 30.”

The *condition set* is the summation of all the unticketed seats leaving New Orleans on June 30. The *fragments* are the data tables at all of the other terminal sites listed in the database. The names of these sites indicate a different location, although in some database systems it is possible that different names would not indicate a different physical location. A *target set* is a row of data in a fragment which lists the terminal name, flight number, the date, and the number of unticketed seats. The answer to this query would then be generated by comparing the number of unticketed seats in the New Orleans condition set with the summation over the target sets of all the unticketed seats in each fragment.

**Example 2**

Suppose that it is known that St. Louis also has a large number of unticketed seats. Then the following question could be submitted:

“List the terminals and their flight numbers which had more unticketed seats on departure flights than New Orleans and St. Louis on June 30.”

Now we have two geographically distributed condition sets, that is, the number of unticketed seats at New Orleans and at St. Louis. Prior to the comparative operations with the fragments, we have the option of (1) generating the two condition sets, each containing the number of unticketed seats, or (2) by performing (1) and then selecting the condition set with the highest number of unticketed seats. The second option incurs the communication cost of either shipping the New Orleans number to St. Louis or vice versa. The benefit would be the potential...