Chapter 6

An Adaptive Probe-Based Technique to Optimize Join Queries in Distributed Internet Databases

Latifur Khan
University of Texas at Dallas, USA

Dennis McLeod and Cyrus Shahabi
University of Southern California, Los Angeles, USA

An adaptive probe-based optimization technique is developed and demonstrated in the context of an Internet-based distributed database environment. More and more common are database systems, which are distributed across servers communicating via the Internet where a query at a given site might require data from remote sites. Optimizing the response time of such queries is a challenging task due to the unpredictability of server performance and network traffic at the time of data shipment; this may result in the selection of an expensive query plan using a static query optimizer. We constructed an experimental setup consisting of two servers running the same DBMS connected via the Internet. Concentrating on join queries, we demonstrate how a static query optimizer might choose an expensive plan by mistake. This is due to the lack of a priori knowledge of the run-time environment, inaccurate statistical assumptions in size estimation, and neglecting the cost of remote method invocation. These shortcomings are addressed collectively by proposing a probing mechanism. Furthermore, we extend our mechanism with an adaptive technique that detects sub-optimality...
of a plan during query execution and attempts to switch to the cheapest plan
while avoiding redundant work and imposing little overhead. An
implementation of our run-time optimization technique for join queries was
constructed in the Java language and incorporated into an experimental
setup. The results demonstrate the superiority of our probe-based optimization
over a static optimization.

A distributed database is a collection of partially independent databases that
share a common schema, and coordinates processing of non-local transactions.
Processors communicate with one another through a communication network
(Silberschatz, Korth, and Sudarshan, 1997; Yu and Meng, 1998). We focus on
distributed database systems with sites running homogeneous software (i.e.,
database management system, DBMS) on heterogeneous hardware (e.g., PC and
Unix workstations) connected via the Internet. The Internet databases are appro-
priate for organizations consisting of a number of almost independent sub-
organizations such as a University with many departments or a bank with many
branches. The idea is to partition data across multiple geographically or adminis-
tratively distributed sites where each site runs an almost autonomous database
system.

In a distributed database system, some queries require the participation of
multiple sites, each processing part of the query as well as transferring data back
and forth among themselves. Since usually there is more than one plan to execute
such a query, it is crucial to obtain the cost of each plan, which highly depends on
the amount of participation by each site as well as the amount of data shipment
between the sites. Assuming a private/dedicated network and servers, this cost can
be computed a priori due to the predictability of servers and network conditions
and availability of effective network bandwidth. However, in the Internet environ-
ment, which is based on a best effort service, there are a number of unpredictable
factors that make the cost computation complicated (Paxson and Floyd; 1997).
A static query optimizer that does not consider the characteristics of the environ-
ment or only considers the a priori knowledge on the run-time parameters might end
up choosing expensive plans due to these unpredictable factors. In the following
paragraph, we explain some of these factors via simple examples.

Participating sites (or servers) of Internet database systems might have
different processing powers. One site might be a high-end multiprocessor system
while the other is a low-end PC running (say) Windows NT. In addition, since most
queries are I/O intensive, a site having faster disk drives might observe a better
performance. In an Internet-based environment these sites might be dedicated to
a single application or multiple simultaneous applications. For example, one site
might only run a database server while the other is a database server, a web server,
An Investigation of the Affect Heuristic in the Brazilian Stock Market  
[www.igi-global.com/article/an-investigation-of-the-affect-heuristic-in-the-brazilian-stock-market/121557?camid=4v1a](www.igi-global.com/article/an-investigation-of-the-affect-heuristic-in-the-brazilian-stock-market/121557?camid=4v1a)

Implication of Requirements Engineering in ICT4D Project Development  
[www.igi-global.com/article/implication-of-requirements-engineering-in-ict4d-project-development/135529?camid=4v1a](www.igi-global.com/article/implication-of-requirements-engineering-in-ict4d-project-development/135529?camid=4v1a)