

Cost Estimation for Queries Experiencing Multiple Contention States in Dynamic Multidatabase Environments

Qiang Zhu, Satyanarayana Motheramgari, Yu Sun

Department of Computer and Information Science, The University of Michigan – Dearborn,
Dearborn, MI, USA

Abstract. Accurate query cost estimation is crucial to query optimization in a multidatabase system. Several estimation techniques for a static environment have been suggested in the literature. To develop a cost model for a dynamic environment, we recently introduced a multistate query-sampling method. It has been shown that this technique is promising in estimating the cost of a query run in any given contention state for a dynamic environment. In this paper, we study a new problem on how to estimate the cost of a large query that may experience multiple contention states. Following the discussion of limitations for two simple approaches, i.e., single state analysis and average cost analysis, we propose two novel techniques to tackle this challenge. The first one, called fractional analysis, is suitable for a gradually and smoothly changing environment, while the second one, called the probabilistic approach, is developed for a rapidly and randomly changing environment. The former estimates a query cost by analyzing its fractions, and the latter estimates a query cost based on Markov chain theory. The related issues including cost formula development, error analysis, and comparison among different approaches are discussed. Experiments demonstrate that the proposed techniques are quite promising in solving the new problem.

Keywords: Cost estimation; Dynamic environment; Multidatabase; Query optimization

1. Introduction

To meet users' increasing needs to access data from multiple pre-existing databases managed by heterogeneous database management systems (DBMS), multidatabase systems (MDBS) have been studied by many database researchers in

recent years (Litwin et al., 1990; Sheth and Larson, 1990; Lu et al., 1992; Lee and Chen, 1997). An MDBS is a global system built on top of multiple local (component) DBMSs and provides users with a uniform interface to access local databases. A key feature of an MDBS is local autonomy that each local database system retains to serve existing local applications. The global system can only interact with local DBMSs at their external user interfaces.

A global query issued on an MDBS is decomposed into a set of local queries executed at local database systems during query processing. The results from local queries are integrated into the final query result returned to its user. However, the way to decompose a global query is not unique. Different decomposition strategies may yield significantly different performance in the distributed environment. Choosing a good decomposition and integration strategy for a given global query is the task of global query optimization. To perform global query optimization, cost information for local queries to be performed on local database systems is required. However, such information is unavailable to the global query optimizer since the internal implementation details of a local DBMS is unknown to the MDBS. Estimating the costs of local queries at the global level in an MDBS is a major challenge for global query optimization in the system.

To tackle this challenge, a number of techniques have been proposed in the literature. Du et al. (1992) proposed a calibration method that makes use of the observed costs of some special queries run against a special synthetic calibrating database to deduce necessary local cost parameters. Gardarin et al. (1996) extended Du et al.'s method so as to calibrate cost models for object-oriented local database systems in an MDBS. Zhu and Larson (1998, 1996, 1994) proposed a query-sampling method that develops regression cost models for local query classes based on observed costs of sample queries run against actual user databases. Zhu and Larson (1997) also introduced a fuzzy method based on fuzzy set theory to derive fuzzy cost models in an MDBS. Naacke et al. (1998) suggested an approach to combining a generic cost model with specific cost information exported by wrappers for local database systems. Adali et al. (1996) suggested maintaining a cost vector database to record cost information for every query issued to a local database system. Roth et al. (1999) introduced a framework for costing in the *Garlic* federated system.

All the above techniques considered only a static environment, i.e., assuming that the environment does not change significantly over time. However, such an assumption may not be true in reality since many factors such as the number of concurrent processes in a multidatabase system environment may change significantly. The cost of a query can be dramatically different at different times in a dynamic system environment. For example, in one of our experiments, the cost of a sample query¹ performed on Oracle 8 in a dynamical environment varied from 2.58 s to 127.05 s (49 times!) when we had 1–30 concurrent user processes in the environment. Hence query cost estimates obtained for a static environment cannot be used in a dynamic environment.

In fact, the environmental factors in a local database system in an MDBS can be classified into three types based on their changing frequencies: (1) *frequently changing factors*, such as CPU load, number of I/Os per second, and size of memory space being used; (2) *occasionally changing factors*, such as local DBMS

¹ The query was `SELECT a1, a5, a7 FROM R WHERE a3 > 300 and a8 < 2000` on table `R(a1, a2, ..., a9)` with 50,000 tuples of random data.

configuration parameters, local database physical schemas, and local database conceptual schemas; and (3) *steady factors*, such as local DBMS type, local database location, and local CPU speed.

Clearly, the steady factors usually do not cause a problem for a query cost model. For the occasionally changing factors, a simple and effective approach to capture them in a cost model is to rebuild the cost model periodically or whenever a significant change for the factors occurs by, for example, invoking the query-sampling method (Zhu and Larson, 1998). Since such factors do not change very often, rebuilding cost models from time to time to capture them is acceptable. However, this approach cannot be used for the frequently changing factors because rebuilding cost models frequently would significantly increase the system load and maintenance overhead for cost models.

To capture the frequently changing factors in a cost model, we recently proposed a qualitative approach (Zhu et al., 2000). This approach extends our previous query sampling method (Zhu and Larson, 1998) and develops regression cost models using qualitative variables to indicate system contention states. Each contention state reflects a combined effect of dynamic factors on the system. Although such a cost model can be used to estimate the costs of queries for any contention state in the dynamic environment, each query is assumed to be run in a single contention state. The qualitative approach cannot directly solve the problem of estimating the cost of a large query run in multiple contention states.

To estimate the costs of large queries experiencing multiple contention states in a dynamic multidatabase environment, we develop two new techniques in this paper. The first technique, called fractional analysis approach, is to estimate query costs in a dynamic environment in which the system contention states change gradually and smoothly. The idea is to analyze and integrate the fractions of a query cost for multiple experienced contention states. The second technique, called the probabilistic approach, is to estimate query costs in a dynamic environment in which the system contention states change rapidly and randomly. The idea is to make use of the theory of Markov chains to derive a cost formula to estimate the query costs in such an environment. These two techniques together with our qualitative approach provide a suite of techniques to estimate the costs of queries for different cases in a dynamic multidatabase environment.

The rest of this paper is organized as follows. Section 2 outlines our qualitative approach to developing cost models with qualitative variables for dynamic multidatabase environments. Section 3 presents the fractional analysis approach to estimating query costs. Section 4 discusses the probabilistic approach to estimating query costs. Section 5 shows some experimental results. The last section summarizes the conclusions.

2. Dynamic Cost Models with Qualitative Variables

To incorporate the dynamic factors in a multidatabase system into a cost model, we proposed an effective qualitative approach (Zhu et al., 2000). In this approach, we consider the combined effect of all the factors on a query cost together rather than individually. Although the dynamic factors change differently in terms of changing frequency and level, they all contribute to the contention level of the underlying system environment, which represents their net effect. Notice that the cost of a query increases as the contention level. The system contention level can be divided into a number of discrete states (categories) such as ‘High Contention’

(S_H), ‘Medium Contention’ (S_M), ‘Low Contention’ (S_L), and ‘No Contention’ (S_N). A qualitative variable is used to indicate the contention states. This qualitative variable, therefore, reflects the combined effect of the dynamic environmental factors. A cost model including such a qualitative variable can capture the dynamic factors to a certain degree.

Since, for a given query, its cost increases with the system contention level, we can use the cost of a probing query to gauge the contention level and classify the contention states for the dynamic system environment. To obtain an appropriate classification of system contention states, we first partition the range of a probing query cost in the given dynamic environment into subranges (intervals) with an equal size. Each subrange represents a contention state. If some neighbor contention states are found to have a similar effect on the derived cost model, they are merged into one state. Such a uniform partition with merging adjustment procedure for a classification of contention states has been proven to be very effective in practice (Zhu et al., 2000).

A qualitative variable X with M possible system contention states S_1, S_2, \dots, S_M can be represented by a set of $M - 1$ indicator (binary) variables Z_1, Z_2, \dots, Z_{M-1} . That is, $X = S_i$ ($1 \leq i \leq M - 1$) is represented by $Z_i = 1$ and $Z_j = 0$ (for any $j \neq i$); and $X = S_M$ is represented by $Z_k = 0$ (for any $1 \leq k \leq M - 1$). Including qualitative variable X in a cost model is equivalent to including indicator variables Z_1, Z_2, \dots, Z_{M-1} in the cost model.

To develop a cost model including the indicator variables, we extend our previous query sampling method (Zhu and Larson, 1998). In other words, we use observed costs of sample queries to build a regression cost model with indicator variables as follows:

$$Y = \underbrace{(B_0^0 + \sum_{j=1}^{M-1} B_0^j Z_j)}_{intercepts} + \sum_{i=1}^n \underbrace{(B_i^0 + \sum_{j=1}^{M-1} B_i^j Z_j)}_{slopes} X_i, \quad (1)$$

where Y is the query cost, X_i 's are explanatory variables, Z_j 's are indicator variables, and B_i^j 's are the regression coefficients. The intercepts and slopes of equation (1) change from one contention state to another, indicated by the values of Z_i 's. Since the above qualitative approach is obtained by introducing multiple contention states into our previous query sampling method, it is also called the multistate query sampling method. For more details of this method, please refer to Zhu et al. (2000).

3. Fractional Analysis Approach

One assumption made by the qualitative approach discussed in the last section is that the contention state does not change during the execution of a query although different executions of queries can be run in different contention states. This assumption is usually valid for small (cost) queries. For large (cost) queries, they may experience multiple contention states during their executions. How to estimate the cost of a query when it experiences multiple states during its execution is the issue to be discussed in this and the following sections.

3.1. Single-State Analysis

There are two simple approaches to estimating the cost of a query experiencing multiple states. One is called *single-state analysis*. The idea is to ignore the changes in contention states during the execution of a query and use the dynamic cost model with a qualitative variable discussed in Section 2 together with one prevailing contention state to estimate the query cost. The prevailing contention state can be (1) the initial state in which the query is to start; (2) the median state among all states; or (3) a random state from all states. Unlike the initial state, the median and random states may not actually be experienced by the query at all. Hence the initial state may be superior in most cases for the single-state analysis approach. The advantage of this approach is that one step application of the dynamic cost model is sufficient to give a cost estimate. However, the resulting estimate may be inaccurate since not all experienced contention states are considered.

3.2. Average Cost Analysis

Another simple approach is called the *average cost analysis*. The idea is to use the average of costs for all the states in the environment as an estimate for the cost $C(Q)$ of query Q ; that is, using

$$\bar{C}(Q) = \sum_{i=1}^M C(Q, S_i) / M \quad (2)$$

to estimate $C(Q)$, where $C(Q, S_i)$ denotes the cost for Q in state S_i and S_1, S_2, \dots, S_M are all possible states in the given environment. However, $C(Q, S_i)$ typically cannot be known exactly. Its estimate $C_e(Q, S_i)$ has to be used in (2) in place of $C(Q, S_i)$, resulting in an estimated average cost $\bar{C}_e(Q)$. Thus $\bar{C}_e(Q)$ is actually used to estimate $C(Q)$ in the average cost analysis.

Although the average cost estimate is usually better than the single-state cost estimates, it may still be quite rough due to the fact that some contention states may never be experienced, while other contention states may be experienced with various durations for the given query. One might suggest considering only those contention states that the given query experiences in the average cost analysis. However, the difficulty is that which contention states are to be experienced by a query are unknown in advance.

3.3. Fractional Analysis

In the following discussion, we are going to introduce a better cost estimation via a finer analysis, called *fractional analysis*. The key idea is to analyze a query cost by fractionalizing it according to the contention states to be experienced.

We notice that the system load in a particular application environment often demonstrates a certain pattern. Figure 1 shows the load for a system environment observed in a real-world company on different days. Clearly, the loads follow a similar pattern during every observed day in the company. The loads are minimum off working hours. The loads start to grow in the morning when the working hours begin and decline when the working hours are close to the end

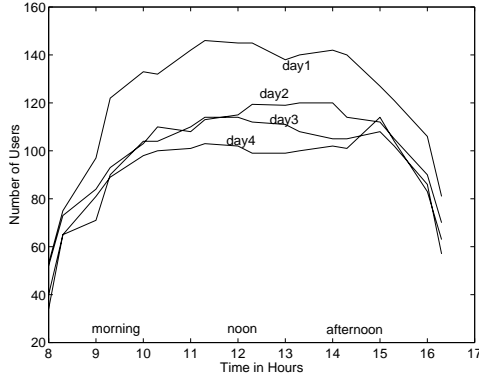


Fig. 1. System loads in a company.

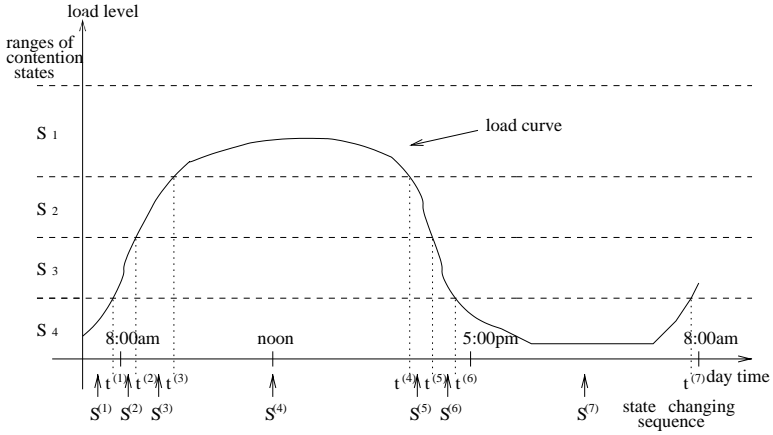


Fig. 2. A typical load curve.

of the day. A curve depicting such a pattern in which the system load changes over time in an application environment is called a *load curve*.² Such a load curve can be obtained via calibrating the application environment under consideration. One assumption made in the following discussion is that the load curve for the given application environment is known in advance.

As suggested in Section 2, the load (contention) level is divided into a number of discrete contention states (see Fig. 2), where the load level is measured by a probing query cost. Let $\Delta = \{ S_1, S_2, \dots, S_M \}$ be the set of all possible contention states; $\{ S^{(i)}, i = 1, 2, \dots \}$ be the sequence of contention states occurring along the load curve in the given application environment, where $S^{(i)} \in \Delta$; $t^{(i-1)}$ and $t^{(i)}$ be the starting and ending times for state $S^{(i)}$ ($i = 1, 2, \dots$).

Consider query Q starting its execution at time t_o^s ($t^{(k-1)} \leq t_o^s < t^{(k)}$) in state $S^{(k)}$. Let $C(Q)$ be the cost of query Q , which may experience multiple states. Let $C(Q, S^{(i)})$ ($i = k, k + 1, \dots$) be the cost of query Q if the query is executed entirely

² Without loss of generality, we assume that the load curve is periodic. The period can be a day, a week, a month, a year, or any other reasonable duration.

in state $S^{(i)}$. Let $T^{(k)} = \min\{ C(Q, S^{(k)}), (t^{(k)} - t_0^s) \}$ and $T^{(i)} = (t^{(i)} - t^{(i-1)})$ for $i \geq k + 1$.

If $C(Q, S^{(k)}) \leq (t^{(k)} - t_0^s)$, query Q is to experience only one contention state $S^{(k)}$. Hence $C(Q, S^{(k)})$ is the cost for query Q , i.e., $C(Q) = C(Q, S^{(k)})$.

If $C(Q, S^{(k)}) > (t^{(k)} - t_0^s)$, query Q is to experience more than one contention state. Then $T^{(k)}/C(Q, S^{(k)})$ (clearly, < 1 in this case) is the fraction of work done for Q in state $S^{(k)}$. The remaining fraction $[1 - T^{(k)}/C(Q, S^{(k)})]$ of work for Q is to be done in the subsequent contention states. If $[1 - T^{(k)}/C(Q, S^{(k)})] * C(Q, S^{(k+1)}) \leq (t^{(k+1)} - t^{(k)})$, all remaining work of Q can be done in state $S^{(k+1)}$. Thus the cost of Q is: $C(Q) = T^{(k)} + [1 - T^{(k)}/C(Q, S^{(k)})] * C(Q, S^{(k+1)})$.

If $[1 - T^{(k)}/C(Q, S^{(k)})] * C(Q, S^{(k+1)}) > (t^{(k+1)} - t^{(k)})$, query Q is to experience more than two contention states. Then $T^{(k+1)}/C(Q, S^{(k+1)})$ is the fraction of work done for Q in state $S^{(k+1)}$, and $T^{(k)}/C(Q, S^{(k)}) + T^{(k+1)}/C(Q, S^{(k+1)})$ is the fraction of work done so far (in both states $S^{(k)}$ and $S^{(k+1)}$). The remaining fraction $[1 - T^{(k)}/C(Q, S^{(k)}) - T^{(k+1)}/C(Q, S^{(k+1)})]$ of work for Q is to be done in the subsequent contention states. If $[1 - T^{(k)}/C(Q, S^{(k)}) - T^{(k+1)}/C(Q, S^{(k+1)})] * C(Q, S^{(k+2)}) \leq (t^{(k+2)} - t^{(k+1)})$, all remaining work of Q can be done in state $S^{(k+2)}$. Hence the cost of Q is: $C(Q) = T^{(k)} + T^{(k+1)} + [1 - T^{(k)}/C(Q, S^{(k)}) - T^{(k+1)}/C(Q, S^{(k+1)})] * C(Q, S^{(k+2)})$.

In general,

$$C(Q) = \sum_{i=k}^m T^{(i)} + [1 - \sum_{i=k}^m T^{(i)}/C(Q, S^{(i)})] * C(Q, S^{(m+1)}) \quad (3)$$

where m is the minimum integer such that $m \geq k$ and $[1 - \sum_{i=k}^m T^{(i)}/C(Q, S^{(i)})] * C(Q, S^{(m+1)}) \leq T^{(m+1)}$. Since, as mentioned before, $C(Q, S^{(j)})$ ($k \leq j \leq m + 1$) cannot be known exactly, its estimate $C_e(Q, S^{(j)})$ has to be used in the above procedure and formula (3), resulting in a cost estimate $C_e(Q)$ for query Q .

Note that m cannot be determined in advance. It has to be determined during the fractional analysis. The following algorithm describes the fractional analysis procedure.

Algorithm 3.1. Fractional Analysis (FA)

Input: The load curve including the contention states changing sequence $\{ S^{(i)}, i = 1, 2, \dots \}$ and the starting time $t^{(i-1)}$ and ending time $t^{(i)}$ for each state $S^{(i)}$ ($i = 1, 2, \dots$); the starting time t_0^s of query Q ; the cost (estimation) model $C_e(Q, S)$ for estimating the cost of query Q in any contention state S .

Output: Cost estimate $C_e(Q)$ for query Q from the fractional analysis.

Method:

1. **begin**
2. Find the initial state $S^{(k)}$ for Q such that $t^{(k-1)} \leq t_0^s < t^{(k)}$;
3. Let $F := 0$; $C := 0$; $T := t^{(k)} - t_0^s$; $m := k - 1$;
4. **while** $(1 - F) * C_e(Q, S^{(m+1)}) > T$ **do**
5. $C := C + T$;
6. $F := F + T/C_e(Q, S^{(m+1)})$;
7. $m := m + 1$;
8. $T := t^{(m+1)} - t^{(m)}$;
9. **end**;
10. $C := C + (1 - F) * C_e(Q, S^{(m+1)})$;
11. **return** C ;
12. **end**.

3.4. Properties and Error Analysis

The estimation error of $C_e(Q)$ from the fractional analysis is mainly caused by the errors of parameter estimates $C_e(Q, S^{(j)})$'s ($k \leq j \leq m+1$) used in the analysis.³ By the total differential of $C(Q)$ from (3):⁴

$$\begin{aligned} d C(Q) &= \sum_{i=k}^{m+1} \frac{\partial C(Q)}{\partial C(Q, S^{(i)})} d C(Q, S^{(i)}) \\ &= \sum_{i=k}^m \{ [C(Q, S^{(m+1)}) * T^{(i)}] / C(Q, S^{(i)})^2 \} d C(Q, S^{(i)}) \\ &\quad + [1 - \sum_{i=k}^m T^{(i)} / C(Q, S^{(i)})] d C(Q, S^{(m+1)}) \end{aligned}$$

we have the (absolute) estimation error of $C_e(Q)$ as follows:

$$\begin{aligned} \varepsilon(C(Q)) &\approx \sum_{i=k}^m \{ [C(Q, S^{(m+1)}) * T^{(i)}] / C(Q, S^{(i)})^2 \} \varepsilon(C(Q, S^{(i)})) \\ &\quad + [1 - \sum_{i=k}^m T^{(i)} / C(Q, S^{(i)})] \varepsilon(C(Q, S^{(m+1)})) \end{aligned} \quad (4)$$

where $\varepsilon(C(Q)) = C(Q) - C_e(Q)$ and $\varepsilon(C(Q, S^{(j)})) = C(Q, S^{(j)}) - C_e(Q, S^{(j)})$ ($k \leq j \leq m+1$).

From (4), we can see that the estimation error $\varepsilon(C(Q))$ of cost estimate $C_e(Q)$ for query Q from the fractional analysis is proportional to the estimation error $\varepsilon(C(Q, S^{(j)}))$ of cost estimate $C_e(Q, S^{(j)})$ for Q in each state $S^{(j)}$ and the fraction of work done in each contention state (i.e., $T^{(i)} / C(Q, S^{(i)})$ for $S^{(i)}$ ($k \leq i \leq m$) and $[1 - \sum_{i=k}^m T^{(i)} / C(Q, S^{(i)})]$ for $S^{(m+1)}$). In other words, the more accurate cost estimates $C_e(Q, S^{(j)})$'s we use in the fractional analysis and the more fraction of work for query Q is done in those states $S^{(t)}$'s with better estimates $C_e(Q, S^{(t)})$'s ($t \in \{k, k+1, \dots, m+1\}$), the better cost estimate for Q we can get from the fractional analysis.

Comparing fractional analysis with single-state analysis, we notice that, when query Q is expected to complete its execution entirely in its initial contention state (i.e., $C_e(Q, S^{(k)}) \leq (t^{(k)} - t_o^s)$), the cost estimates for Q from the fractional analysis and the initial single-state analysis are identical. If such an initial contention state also happens to be the median (or randomly selected) contention state, the query cost estimates from the fractional analysis and the median (or random) single-state analysis are identical. However, in general, the execution of a large query may experience more than one contention state. Single-state analysis does not adopt a correct cost model in such a case. Thus a model error is added into its estimation error. On the other hand, fractional analysis typically utilizes the correct cost model by considering multiple contention states that the query experiences. Hence it usually gives a better cost estimate than a single-state analysis.

³ Assume that the load curve including its associated contention states changing sequence is correct. Otherwise some error may also be caused by an incorrect load curve.

⁴ Assume that $T^{(k)} < C(Q, S^{(k)})$. Otherwise it is trivial to get $d C(Q) = d C(Q, S^{(k)})$.

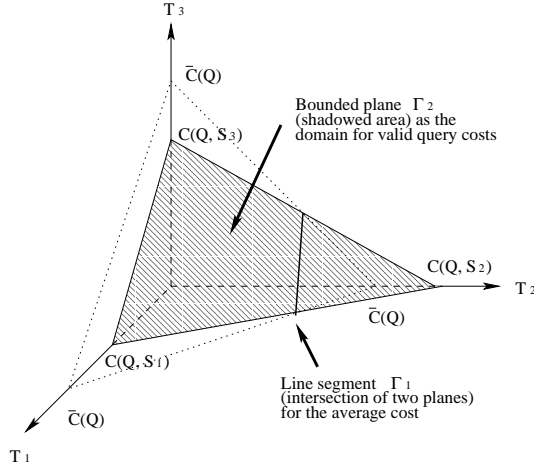


Fig. 3. Domain for valid query costs.

Comparing fractional analysis with average cost analysis, we notice that average cost analysis does not adopt the correct cost model in most cases, although it may be better than single-state analysis. To demonstrate this claim, let us consider the sequence of contention states $S^{(k)}, S^{(k+1)}, \dots, S^{(m)}, S^{(m+1)}$ experienced by query Q . Let I_n be the set of all indexes u 's such that $S^{(u)}$ is in the sequence and $S^{(u)} = S_n \in \Delta$ for $n \in \{1, 2, \dots, M\}$. Let $T_n = \sum_{u \in I_n} T^{(u)}$ (i.e., the accumulated duration for Q in state S_n). Assume that $T_n = 0$ if $I_n = \emptyset$ (empty set). Then formula (3) becomes:⁵

$$C(Q) = T_1 + T_2 + \dots + T_M, \quad (5)$$

where

$$\begin{cases} 0 \leq T_n \leq C(Q, S_n) & (1 \leq n \leq M) \\ \sum_{n=1}^M T_n / C(Q, S_n) = 1 \end{cases} \quad (6)$$

Geometrically, domain (6) for valid query costs can be represented by a bounded $(M-1)$ -dimensional linear surface⁶ (i.e., plane) Γ_{M-1} in the M -dimensional space with T_n 's as axes. Figure 3 shows such a Γ_{M-1} for $M=3$ (i.e., the shadowed plane region).

For each point (T_1, T_2, \dots, T_M) on Γ_{M-1} , formula (5) gives a valid query cost.

However, not every valid query cost $C(Q)$ equals the average cost $\bar{C}(Q)$ from (2). $C(Q) = \bar{C}(Q)$ only for the set of points on Γ_{M-1} satisfying:

$$\begin{cases} 0 \leq T_n \leq C(Q, S_n) & (1 \leq n \leq M), \\ \sum_{n=1}^M T_n / C(Q, S_n) = 1, \\ \sum_{n=1}^M T_n = \lceil \sum_{n=1}^M C(Q, S_n) \rceil / M \end{cases} \quad (7)$$

Unless $M=1$, condition (7) specifies a limited subset of points on Γ_{M-1} . Geometrically, this subset can be represented by a bounded $(M-2)$ -dimensional

⁵ Note that formula (5) cannot directly be used to estimate $C(Q)$ since T_n 's have to be determined via fractional analysis. It is only used to study the properties of the cost formula.

⁶ When $M=2$ or 1, Γ_{M-1} is degraded to a line segment or a point, respectively.

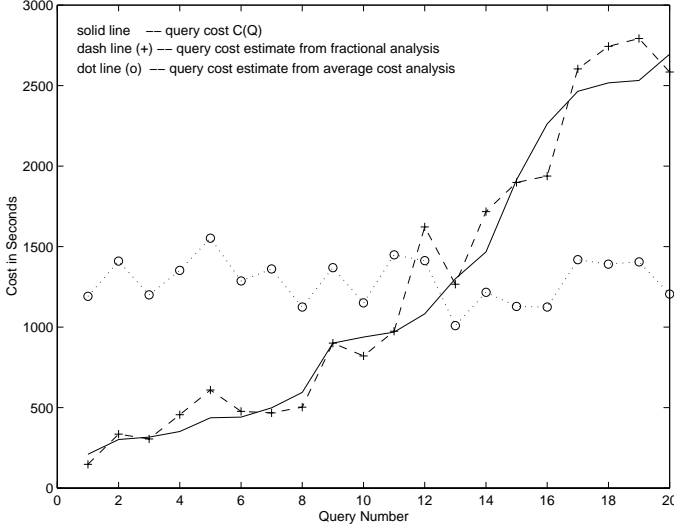


Fig. 4. Comparison of cost estimates from fractional analysis and average cost analysis via simulation.

linear surface⁷ Γ_{M-2} in the M -dimensional space. The line segment lying in the shadowed plane region in Fig. 3 shows such a Γ_{M-2} for $M = 3$.

Clearly, $C(Q) \neq \bar{C}(Q)$ in most cases. In other words, average cost analysis often adds a model error (i.e., $C(Q) - \bar{C}(Q)$) into its estimation error. Thus cost estimate $\bar{C}_e(Q)$ obtained from average cost analysis often suffers both the model error and the parameter estimation errors (i.e., $C(Q, S_n) - C_e(Q, S_n)$ for $1 \leq n \leq M$). On the other hand, fractional analysis typically does not introduce a model error. Therefore, fractional analysis can usually give better cost estimates. This observation is demonstrated by a comparison of cost estimates obtained from the two approaches via simulation in Fig. 4. In the simulation, a set of given $C(Q, S_n)$'s ($1 \leq n \leq 5$) were considered. T_n 's were randomly generated within the domain, and $C_e(Q, S_n)$'s were randomly generated with relative errors within 30% with respect to their corresponding $C(Q, S_n)$'s.

Although condition (7) gives a necessary and sufficient condition for $C(Q) = \bar{C}(Q)$, it is very general and not intuitive. The following propositions consider the relationships between $C(Q)$ and $\bar{C}(Q)$ in two interesting special cases.

Proposition 3.1. Assume that $C(Q, S_j) \neq C(Q, S_i)$ for some $j \neq i$. If $T_1 = T_2 = \dots = T_M$, then $C(Q) < \bar{C}(Q)$, where $C(Q)$ and $\bar{C}(Q)$ are from (3) and (2), respectively.

Proof. Let $T = T_j$ ($1 \leq j \leq M$). Since $\sum_{j=1}^M T_j / C(Q, S_j) = 1$, we have $T = 1 / [\sum_{j=1}^M 1 / C(Q, S_j)]$. From (5), we have

$$C(Q) = M / \left[\sum_{j=1}^M 1 / C(Q, S_j) \right] \tag{8}$$

⁷ When $M = 2$, Γ_{M-2} is degraded to a point.

On the other hand, we have

$$\begin{aligned} & \left[\sum_{j=1}^M 1/C(Q, S_j) \right] * \left[\sum_{i=1}^M C(Q, S_i) \right] = \sum_{j=1}^M \sum_{i=1}^M C(Q, S_i)/C(Q, S_j) \\ & = M + \sum_{j=1}^{M-1} \sum_{i=j+1}^M [C(Q, S_i)/C(Q, S_j) + C(Q, S_j)/C(Q, S_i)] \end{aligned} \quad (9)$$

Notice that $\frac{a}{b} + \frac{b}{a} \geq 2$ for $a, b > 0$ and the equality is true only if $a = b$. Since $C(Q, S_j) \neq C(Q, S_i)$ for some $j \neq i$ is assumed, i.e., $C(Q, S_i)/C(Q, S_j) + C(Q, S_j)/C(Q, S_i) > 2$ for some $j \neq i$. Hence, from (9)

$$\sum_{j=1}^M 1/C(Q, S_j) * \sum_{i=1}^M C(Q, S_i) > M + \sum_{j=1}^{M-1} \sum_{i=j+1}^M 2 = M^2 \quad (10)$$

From (8), (10) and (2), we have

$$\frac{\bar{C}(Q)}{C(Q)} = \left[\sum_{j=1}^M 1/C(Q, S_j) \right] * \left[\sum_{i=1}^M C(Q, S_i) \right] / M^2 > \frac{M^2}{M^2} = 1$$

Therefore, $C(Q) < \bar{C}(Q)$. \square

One might think that a query cost would equal to the average cost for all contention states if the query spends the same amount of time in every contention state. However, Proposition 3.1 states that the average cost is larger than the actual cost when query Q spends an equal amount of time in every contention state. The reason for this phenomenon is that Q runs in different contention states with different working rates. The higher the contention level is, the slower the working rate. Therefore, with the same amount of time, the query will complete less work in a contention state for higher contention level. If all contention states spend the same amount of time on Q , most work of Q will be done in the contention states for lower contention level. The actual cost of Q will be smaller than the average cost in such a case.

Proposition 3.2. If $T_1/C(Q, S_1) = T_2/C(Q, S_2) = \dots = T_M/C(Q, S_M) = 1/M$ (i.e., $T_n = C(Q, S_n)/M$ for $1 \leq n \leq M$), then $C(Q) = \bar{C}(Q)$, where $C(Q)$ and $\bar{C}(Q)$ are from (3) and (2), respectively.

Proof. Without loss of generality, we assume $S^{(m+1)} = S_M$. From (3), we have

$$\begin{aligned} C(Q) &= \sum_{j=1}^{M-1} \sum_{u \in I_j} T^{(u)} + \sum_{u \in I_M \wedge u \neq m+1} T^{(u)} + \left[1 - \sum_{j=1}^{M-1} \sum_{u \in I_j} T^{(u)}/C(Q, S_j) \right. \\ &\quad \left. - \sum_{u \in I_M \wedge u \neq m+1} T^{(u)}/C(Q, S_M) \right] * C(Q, S_M) \\ &= \sum_{j=1}^{M-1} T_j + T_M - T^{(m+1)} + \left[1 - \sum_{j=1}^{M-1} T_j/C(Q, S_j) - T_M/C(Q, S_M) \right. \\ &\quad \left. + T^{(m+1)}/C(Q, S_M) \right] * C(Q, S_M) \end{aligned}$$

$$\begin{aligned}
 &= \sum_{j=1}^{M-1} C(Q, S_j)/M + C(Q, S_M)/M - T^{(m+1)} + [1 - (M - 1)/M \\
 &\quad - 1/M + T^{(m+1)}/C(Q, S_M)] * C(Q, S_M) \\
 &= \sum_{j=1}^M C(Q, S_j)/M = \bar{C}(Q)
 \end{aligned}$$

□

Note that $T_j/C(Q, S_j)$ is the fraction of work done for Q in contention state S_j (during its total stay in the contention state if there are several visits). Proposition 3.2 states that the average cost for all contention states equals the actual cost when query Q experiences every possible contention state in Δ at least once and completes an equal fraction ($1/M$) of work in each contention state.

In the above discussion, accurate costs $C(Q, S_n)$'s are used. However, the conclusions are still valid for estimated costs. For example, if each $C(Q, S_n)$ is replaced by its estimate $C_e(Q, S_n)$ in (7), the condition becomes a necessary and sufficient condition for cost estimate $C_e(Q)$ from fractional analysis to equal cost estimate $\bar{C}_e(Q)$ from average cost analysis. Similarly, if each $C(Q, S_n)$ is replaced by $C_e(Q, S_n)$ in Propositions 3.1 and 3.2, similar relationships between $C_e(Q)$ and $\bar{C}_e(Q)$ are obtained. More specifically, Proposition 3.1 would state that $C_e(Q) < \bar{C}_e(Q)$ when query Q spends an equal amount of time in each contention state, and Proposition 3.2 would state that $C_e(Q) = \bar{C}_e(Q)$ if the estimated fraction of work (i.e., $T_n/C_e(Q, S_n)$) in each contention state is the same.

4. Probabilistic Approach

Although the fractional analysis approach in the last section can estimate costs for queries experiencing multiple contention states during their executions, one assumption made is that the load curve in the given dynamic environment is known in advance and the load changes gradually. To deal with cases with rapidly and randomly changing loads in a dynamic environment, we develop a probabilistic approach in this section.

4.1. Cost Formula Development

Note that a rapidly and randomly changing load in a dynamic environment causes frequent changes in its contention states. The occurrence of a contention state is a random phenomenon and governed by laws of probability.

Let $\Delta = \{ S_1, S_2, \dots, S_M \}$ be the set of all possible contention states in a dynamic environment. We consider a sequence of occurrences of contention states $\{ X^{(n)}, n = 0, 1, 2, \dots \}$ in the given environment as a stochastic process, where $X^{(n)}$ is a random variable taking values from Δ . $X^{(n)} = S_i$ indicates that the environment is in contention state S_i at time $t_n = t_0 + n * \delta$, where δ is the observing time interval. We notice that the probability for the next contention state $X^{(n+1)}$ taking a particular value usually depends only on the value of the present contention state $X^{(n)}$ and is independent of values of past contention states $X^{(0)}, X^{(1)}, \dots, X^{(n-1)}$. For example, if the present contention state is 'very busy', the next contention state is most likely to be 'quite busy' or 'extremely busy'

regardless of past contention states. In other words, the conditional distribution of any future state $X^{(n+1)}$ satisfies

$$\begin{aligned} P(X^{(n+1)} = S_j \mid X^{(n)} = S_i, X^{(n-1)} = S_{k_{n-1}}, \dots, X^{(0)} = S_{k_0}) \\ = P(X^{(n+1)} = S_j \mid X^{(n)} = S_i) = P_{ij}, \end{aligned}$$

for any $n \geq 0$, and $S_j, S_i, S_{k_{n-1}}, \dots, S_{k_0} \in \Delta$

P_{ij} ($1 \leq i, j \leq M$) denotes the (one-step) transition probability for the system contention state changing from S_i to S_j in the next time interval. Clearly, $P_{ij} \geq 0$ and $\sum_{j=1}^M P_{ij} = 1$ since the system has to be in one of the contention states in Δ in the next time interval. Such a stochastic process is known as a (finite) Markov chain (Parzen, 1962).

The next issue is how to establish the transition probabilities in the Markov chain for a dynamic environment. Note that the contention state in a dynamic environment after each time interval can either remain in the same state or change to other states. However, the probability for the contention state changing to a far-away state is less than that for it changing to a neighbor state.

Let the system contention state at time t_0 be S_i , and the system contention state at next time t_1 be S_j . Recall that a contention state reflects a set of close contention levels which are measured by probing query costs. Let L_k ($1 \leq k \leq M$) be the center of gravity of the contention levels for contention state $S_k \in \Delta$. Let d_{ij} be the distance between L_i and L_j .

If the probability for the system contention state remaining in the same state S_i is q_i , the probability for the system contention state changing to other states from S_i will be $(1 - q_i)$. Among other states, a reasonable assumption is that the probabilities are inversely proportional to their distances to S_i . Hence,

$$P_{ii} = q_i; \quad P_{ij} = [(1 - q_i)/d_{ij}] / [\sum_{j=1}^M 1/d_{ij}], \quad \text{for } 1 \leq i, j \leq M$$

Parameter q_i can be calibrated via experiments. Matrix $(P_{ij})_{M \times M}$ lists all one-step transition probabilities for the Markov chain.

The probability $P_{ij}(n)$ for a contention state S_i changing to another contention state S_j after n time intervals is called an n -step transition probability for the Markov chain. For a finite Markov chain, the limit $\pi_j = \lim_{n \rightarrow \infty} P_{ij}(n)$ exists (Parzen, 1962) and is called the limit probability of state S_j . Two interesting properties (Parzen, 1962) of a limit probability are: (1) it is independent of the initial state (i.e., S_i), and (2) it not only represents the probability of a contention state in a Markov chain after a sufficiently large number of transitions but also represents the long-run portion of time for the Markov chain being in the state.

The limit probabilities for a finite Markov chain satisfy the following system of linear equations (Parzen, 1962):

$$\pi_j = \sum_{i=1}^M \pi_i P_{ij}, \quad \text{for } j = 1, 2, \dots, M, \quad \text{subject to } \sum_{j=1}^M \pi_j = 1$$

which can be used to determine π_j ($1 \leq j \leq M$). All limit probabilities $\{\pi_j \mid j = 1, 2, \dots, M\}$ comprise a so-called long-run distribution for the Markov chain.

Since we consider, in this section, the situation in which Q is a large (cost) query and the contention states in the dynamic environment change frequently, it is expected that there are many transitions during the execution of Q . Since π_i

represents the long-run portion of time for the Markov chain being in S_i , $\pi_i * C(Q)$ is the amount of cost incurred in state S_i . Hence $(\pi_i * C(Q))/C(Q, S_i)$ is the portion of work done for Q in S_i . Clearly, the portions of work done for Q in all states should add to 1, i.e., $\sum_{i=1}^M \frac{\pi_i * C(Q)}{C(Q, S_i)} = 1$. Solving this equation, we have

$$C(Q) = 1 / \left[\sum_{i=1}^M \frac{\pi_i}{C(Q, S_i)} \right] \quad (11)$$

Similar to fractional analysis, parameters $C(Q, S_i)$'s have to be estimated (i.e., using $C_e(Q, S_i)$) by applying the dynamic cost models with qualitative variables discussed in Section 2, resulting in an estimated cost $C_e(Q)$ for query Q .

4.2. Properties and Error Analysis

Similar to fractional analysis, the estimation error of cost estimate $C_e(Q)$ for query Q from the probabilistic approach is mainly caused by the errors of parameter estimates $C_e(Q, S_i)$'s used in the approach.⁸ By the total differential of $C(Q)$ from (11), we get the estimation error for $C_e(Q)$ as follows:

$$\varepsilon(C(Q)) \approx \left\{ 1 / \left[\sum_{i=1}^M \pi_i / C(Q, S_i) \right]^2 \right\} * \left\{ \sum_{i=1}^M [\pi_i / C(Q, S_i)^2] * \varepsilon(C(Q, S_i)) \right\} \quad (12)$$

From (12), we can see that the estimation error $\varepsilon(C(Q))$ of cost estimate $C_e(Q)$ for query Q is proportional to the estimation error $\varepsilon(C(Q, S_i))$ of cost estimate $C_e(Q, S_i)$ of Q in each contention state S_i and the long-run portion (i.e., π_i) of time for the Markov chain being in each contention state. In other words, the more accurate the cost estimates $C_e(Q, S_i)$'s we use in the probabilistic approach and the longer the query Q stays in those contention states S_i 's with better estimates $C_e(Q, S_i)$'s ($t \in \{1, 2, \dots, M\}$), the better cost estimate $C_e(Q)$ for Q we can get from the probabilistic approach.

Comparing the above probabilistic approach with the single-state analysis approach, we notice that the cost estimate for a query from the single-state analysis approach is identical to that from the probabilistic approach when the limit probability for the selected contention state is 1 (i.e., the limit probabilities for other contention states are 0). However, in general, more than one contention state has a non-zero limit probability. Hence single-state analysis may not adopt a correct model for query cost estimation in such a rapidly changing environment.

Comparing the probabilistic approach with the average cost analysis approach, similar to the situation with fractional analysis, we find that average cost analysis does not adopt the correct cost model in most cases. The domain for valid query costs for formula (11) is

$$\begin{cases} 0 \leq \pi_i \leq 1 & (1 \leq i \leq M), \\ \sum_{i=1}^M \pi_i = 1 \end{cases} \quad (13)$$

⁸ Assume that limit probabilities π_i 's are accurate. Otherwise some error may also be caused by inaccurate π_i 's.

while the subset of points in the domain for which $C(Q) = \bar{C}(Q)$ is

$$\begin{cases} 0 \leq \pi_i \leq 1 & (1 \leq i \leq M), \\ \sum_{i=1}^M \pi_i = 1, \\ \{[\sum_{i=1}^M C(Q, S_i)]/M\} * \{\sum_{i=1}^M \pi_i/C(Q, S_i)\} = 1 \end{cases} \quad (14)$$

Clearly, similar to a gradually changing environment, average cost analysis suffers both model error and parameter estimation errors in most cases for a rapidly changing dynamic environment. Hence the probabilistic approach, which suffers only parameter estimation errors, can usually give better query cost estimates in such a dynamic environment.

The following propositions consider the two interesting special cases corresponding to those for Propositions 3.1 and 3.2.

Proposition 4.1. Assume that $C(Q, S_j) \neq C(Q, S_i)$ for some $j \neq i$. If $\pi_1 = \pi_2 = \dots = \pi_M = 1/M$, then $C(Q) < \bar{C}(Q)$, where $C(Q)$ and $\bar{C}(Q)$ are from (11) and (2), respectively.

Proof. This is similar to the proof of Proposition 3.1. \square

Proposition 4.1 states that the average cost for all contention states is larger than the actual cost when the limit probabilities for all states are the same in a rapidly changing dynamic environment. This phenomenon can also be explained by the different working rates in the contention states, like the explanation given for Proposition 3.1.

Proposition 4.2. If $\pi_1/C(Q, S_1) = \pi_2/C(Q, S_2) = \dots = \pi_M/C(Q, S_M)$, then $C(Q) = \bar{C}(Q)$, where $C(Q)$ and $\bar{C}(Q)$ are from (11) and (2), respectively.

Proof. Since $\pi_i/C(Q, S_i) = \pi_j/C(Q, S_j)$, we have $\pi_i = \pi_j * C(Q, S_i)/C(Q, S_j)$. From (11) and (2), we have

$$\begin{aligned} \frac{\bar{C}(Q)}{C(Q)} &= \frac{1}{M} \sum_{i=1}^M C(Q, S_i) \sum_{j=1}^M \pi_j/C(Q, S_j) \\ &= \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^M C(Q, S_i) * \pi_j/C(Q, S_j) \\ &= \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^M \pi_i = \frac{1}{M} \sum_{i=1}^M M * \pi_i = \sum_{i=1}^M \pi_i = 1 \end{aligned}$$

Therefore, $C(Q) = \bar{C}(Q)$. \square

Although $\pi_i/C(Q, S_i)$ does not have any physical meaning, $\pi_i * C(Q)/C(Q, S_i)$ is the portion of work done for Q in state S_i . Therefore, the assumption in Proposition 4.2 implies that each state completes an equal portion of work for Q . In such a case, the average cost for all contention states equals the actual cost in the rapidly changing dynamic environment.

Similar to fractional analysis, if each accurate $C(Q, S_i)$ is replaced by its estimate $C_e(Q, S_i)$ in the above discussion, the conclusions including condition (14) and Propositions 4.1 and 4.2 are also valid for estimated cost $C_e(Q)$ from the probabilistic approach and estimated cost $\bar{C}_e(Q)$ from the average cost analysis.

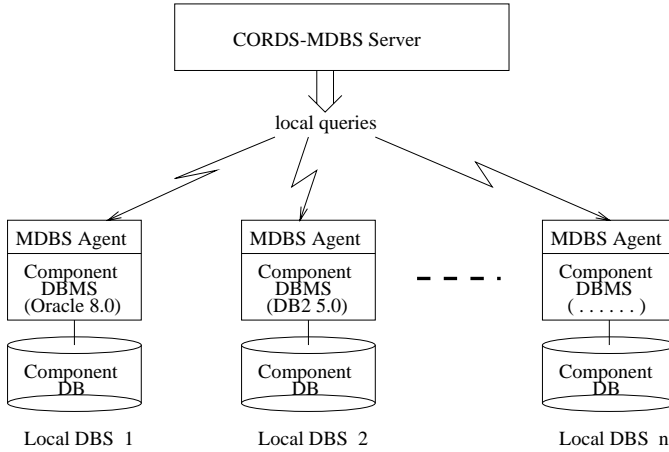


Fig. 5. Experimental environment.

5. Experimental Results

To validate the cost estimation techniques proposed in the previous sections, experiments were conducted using a multidatabase system prototype, named *CORDS-MDBS* (Attaluri et al., 1995). Two commercial DBMSs, i.e., Oracle 8.0 and DB2 5.0, were used as component database systems running under Solaris 5.1 on two SUN UltraSparc 2 workstations. Figure 5 shows the experimental environment. To evaluate the techniques in various dynamic environments, we developed a load builder which can generate dynamic system loads to simulate various dynamic environments following different load curves (for the fractional analysis approach) or retention probability distributions (for the probabilistic approach) for contention states.

The table schemas in the component databases used in the experiments were the same as those⁹ in Zhu and Larson (1998). More specifically, each component database contains 12 tables $R_i(a_1, a_2, \dots, a_n)$ ($1 \leq i \leq 12; 1 \leq n \leq 13$) with all integer columns. The data in the tables are randomly generated using different ranges for different columns to achieve various selectivities. The table cardinalities range from 3000 to 250,000. Each table has some indexed columns. For more details of the test database, please refer to Zhu and Larson (1998).

Since the costs of unary queries are usually not large and the techniques in this paper are for large-cost queries, we chose a join query class for our experiments. Following the multistate query sampling method in Section 2, we drew a sample of queries from the query class and executed them in a dynamic multidatabase environment. Based on the observed costs of sample queries, our cost model building tool automatically selects significant variables and uses them together with a qualitative variable (represented by a set of indicator variables) indicating system contention states to develop a cost model for the query class. Our tool also applies some statistical measures to validate the significance of the cost model. Table 1 shows the cost models developed for the query class on the two component DBMSs, i.e., Oracle and DB2. The coefficient of total

⁹ The size of each table is 10 times larger than that in Zhu and Larson (1998).

determination indicates that both cost models can capture over 99% variations in the query costs. The F -test also shows that both cost models are useful. Between the two models, the one for Oracle is even better. One reason for this may be that many dynamic query-processing strategies are adopted within DB2, while the query-processing strategies within Oracle are relatively static.

To further validate the cost models, test queries were used in our experiments. They were randomly generated from the following forms:

$$\pi_{\alpha}(R_i \overset{\bowtie}{\underset{\text{qualification}}{R_j}})$$

where:

$$\begin{aligned} \text{qualification} &::= [(\text{simple_left} \mid \text{disjunct_left}) \wedge] \text{key_join} \\ & \quad [\wedge (\text{simple_right} \mid \text{disjunct_right})] [\wedge \text{additional_join}] \\ \text{key_join} &::= R_i.a_n = R_j.a_m \\ \text{simple_left} &::= R_i.a_p \omega_1 C_1 \\ \text{simple_right} &::= R_j.a_q \omega_2 C_2 \\ \text{disjunct_left} &::= R_i.a_u \omega_3 C_3 \vee R_i.a_v \omega_4 C_4 \\ \text{disjunct_right} &::= R_j.a_s \omega_5 C_5 \vee R_j.a_t \omega_6 C_6 \\ \text{additional_join} &::= R_i.a_x = R_j.a_y \end{aligned}$$

where $\omega_1 \sim \omega_6 \in \{=, \neq, <, \leq, >, \geq\}$; the other parameters such as the project list (α), referenced columns ($R_i.a_n, R_j.a_q, \dots$) and constants (C_1, C_2, \dots, C_6) are randomly selected from their domains.

We first ran some test queries in the dynamic environment under the restriction that each query only experiences a (random) single contention state. We applied the dynamic cost models in Table 1 to estimate the costs of the test queries and compared the estimated costs with their observed costs as well as the estimated costs using a static cost model.¹⁰ The comparison results are shown in Figs 6 and 7. From the figures we can see that the estimated costs given by the dynamic cost model are much better than those given by the static cost model.

However, assuming a query to experience one state may only be valid for small (cost) queries. For large (cost) queries, the execution of a query may experience more than one contention state. The techniques presented in Sections 3 and 4 should be applied to estimate the cost of such a query.

Notice that a proper classification of system contention states is determined during the development of a cost model with a qualitative variable. For example, the cost model for Oracle in Table 1 uses five states, while the one for DB2 uses four states. To evaluate the effectiveness of the fractional analysis technique, the ‘shape’ of the load curve in Fig. 1 is assumed. However: (1) the ‘noon’ contention state S_1 is split into two state occurrences to simplify the experimental program, i.e., contention state sequences $S_5, S_4, S_3, S_2, S_1, S_1, S_2, S_3, S_4, S_5$ and $S_4, S_3, S_2, S_1, S_1, S_2, S_3, S_4$ were used for experiments on Oracle and DB2, respectively; and (2) the load curve repeats its pattern periodically so that the queries that cannot be finished within the current cycle can be completed within the following cycle(s). The initial starting state for each test query was randomly

¹⁰ The static cost model was developed by using our static sampling query method in Zhu and Larson (1998), i.e., assuming that the environment has only one (static) state.

Table 1. Cost models for a join query class in a dynamic multidatabase environment

(Dynamic) cost model with qualitative variable				
Component DBMS 1 (Oracle)	$(0.7419e+1 - 0.1169e+3 * Z_4 + 0.2748e+2 * Z_3 - 0.3963e+2 * Z_2 + 0.3626e+2 * Z_1) + (0.1131e-2 + 0.2212e-2 * Z_4 + 0.4383e-2 * Z_3 + 0.5517e-2 * Z_2 + 0.6588e-2 * Z_1) * RN_J + (0.4952e-3 - 0.5211e-3 * Z_4 - 0.2522e-3 * Z_3 + 0.3308e-3 * Z_2 + 0.2617e-2 * Z_1) * TN_{J1} + (-0.4691e-3 + 0.6974e-3 * Z_4 + 0.9641e-3 * Z_3 + 0.1900e-2 * Z_2 + 0.3194e-2 * Z_1) * TN_{J2} + (-0.1121e+1 + 0.5489e+1 * Z_4 + 0.1419e+2 * Z_3 + 0.7953e+1 * Z_2 + 0.1452e+2 * Z_1) * RL_J + (-0.1995e+2 + 0.3754e+2 * Z_4 + 0.2510e+2 * Z_3 + 0.3139e+2 * Z_2 + 0.8524e+1 * Z_1) * L_{J1} + (0.1647e+2 - 0.2275e+2 * Z_4 - 0.3715e+2 * Z_3 - 0.2881e+2 * Z_2 - 0.2503e+2 * Z_1) * L_{J2}$			
	Coeff. of multi. determination	Std. error of estimation	Avg. sample cost (sec.)	F-statistics (critical value at $\alpha = 0.01$)
	0.9992	0.1708e+3	0.1178e+4	16166.13 (> 1.25)
(Dynamic) cost model with qualitative variable				
Component DBMS 2 (DB2)	$(-0.1529e+2 - 0.1923e+1 * Z_3 + 0.1178e+2 * Z_2 + 0.3510e+1 * Z_1) + (0.4171e-7 - 0.1697e-7 * Z_3 + 0.6500e-7 * Z_2 + 0.9597e-6 * Z_1) * TN_{J12} + (0.8934e-3 + 0.1697e-2 * Z_3 + 0.2478e-2 * Z_2 + 0.3554e-2 * Z_1) * RN_J + (0.2763e-3 - 0.1106e-2 * Z_3 - 0.4020e-3 * Z_2 + 0.7719e-2 * Z_1) * TN_{J1} + (0.1234e-4 + 0.4639e-4 * Z_3 + 0.2241e-4 * Z_2 + 0.3693e-3 * Z_1) * TZ_{J1} + (0.3684e-4 + 0.6292e-4 * Z_3 + 0.1345e-4 * Z_2 - 0.1768e-3 * Z_1) * TZ_{J2}$			
	Coeff. of multi. determination	Std. error of estimation	Avg. sample cost (sec.)	F-statistics (critical value at $\alpha = 0.01$)
	0.9963	0.2436e+3	0.7860e+3	5108.04 (> 1.41)

RN_J , result table size; TN_{J1} , TN_{J2} , 1st and 2nd intermediate table sizes; TN_{J12} , $TN_{J1} * TN_{J2}$; RL_J , result table tuple length; L_{J1} , L_{J2} , 1st and 2nd operand table tuple lengths; TZ_{J1} , $N_{J1} * L_{J1}$; TZ_{J2} , $N_{J2} * L_{J2}$; N_{J1} , N_{J2} , 1st and 2nd operand table sizes.

selected from the contention states sequence. Different changing patterns ('increase', 'decrease', 'equal', and 'random') of time durations for the contention state occurrences along the curve were tested in our experiments. Figures 8 and 9 show the experimental results for the 'random' case, which well represents all other cases. The following observations can be obtained from our experiments:

- The fractional analysis technique can give good cost estimates for the test queries in a gradually and smoothly changing environment. Most cost estimates in the experiments have relative errors within 30%.
- The cost estimates given by the fractional analysis are clearly better than those given by the average cost analysis or the initial single-state analysis for most cases.
- There are some cases (i.e., queries completed entirely within one state) in which the cost estimates from the fractional analysis and the initial single-state analysis are the same, which is consistent with our theoretical analysis.
- The average costs are larger than the estimated costs given by fractional analysis when a query spent an equal (accumulated) amount of time in every state, which is consistent with Proposition 3.1, which we showed in Section 3.
- The accuracy of the cost model with a qualitative variable used to estimate $C(Q, S_i)$ of query Q for state S_i is important to the fractional analysis, as indicated in the error analysis. Since we obtained a better cost model for

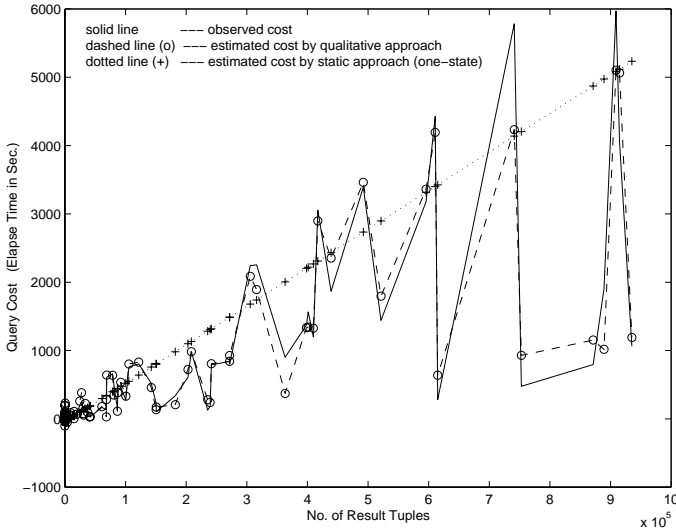


Fig. 6. Estimated costs for test queries via qualitative approach on Oracle 8.0.

Oracle, its fractional analysis results are overall more accurate than those for DB2.

In the experiments to evaluate the effectiveness of the probabilistic approach, we tested various dynamic environments with different retention probability changing patterns ('increase', 'decrease', 'equal', and 'random'). Since an environment with an equal retention probability for all contention states may not yield an equal limit probability for every contention state, we also tested the 'equal' limit probability case. The experimental results for the 'random' case, which well represents other cases, are shown in Figs 10 and 11. From the experimental results, we have the following observations:

- The probabilistic approach gives good cost estimates for a rapidly and randomly changing environment. The majority of cost estimates in the experiments have relative errors within 30%.
- The cost estimates from the probabilistic approach are usually more accurate than the average costs.
- The average costs are larger than the cost estimates from the probabilistic approach when the limit probabilities are the same for all states, as predicted by Proposition 4.1.
- The accuracy of the cost model with a qualitative variable used to estimate $C(Q, S_i)$ of query Q for state S_i is important to the probabilistic approach, as indicated in the error analysis.

In summary, our experimental results demonstrate that the qualitative approach, fractional analysis approach and probabilistic approach comprise a promising suite of techniques in estimating local query costs for a dynamic multidatabase environment.

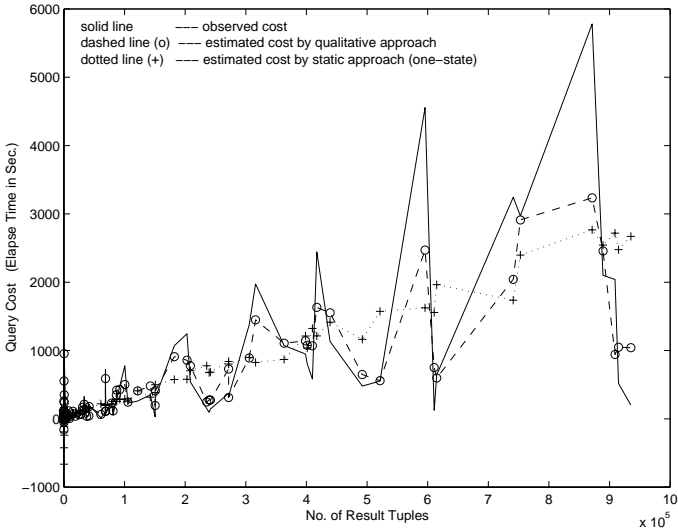


Fig. 7. Estimated costs for test queries via qualitative approach on DB2 5.0.

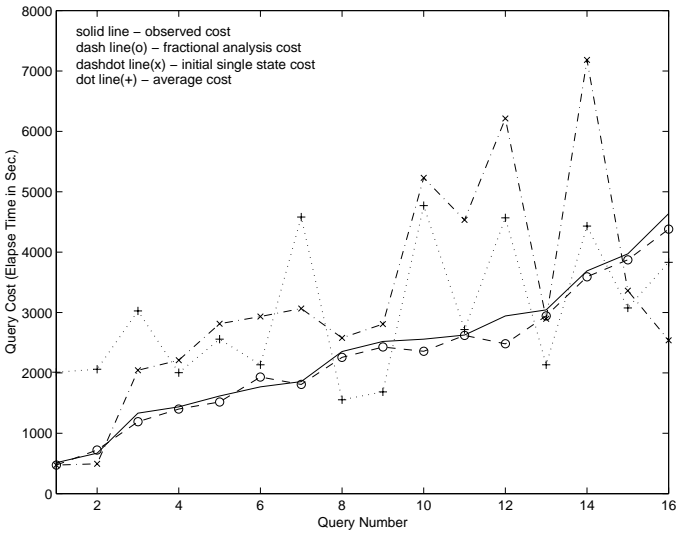


Fig. 8. Estimated costs for test queries via fractional analysis on Oracle 8.0.

6. Conclusion

A major challenge for performing global query optimization in an MDBS is that some local cost information may be unavailable at the global level. Most techniques suggested so far in the literature considered only static system environments, namely, assuming the environment never changes. However, the cost of a query changes dramatically in a realistic dynamic environment.

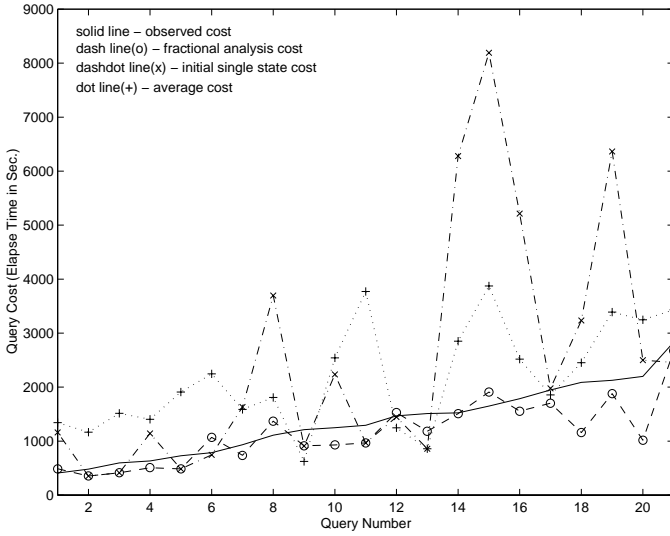


Fig. 9. Estimated costs for test queries via fractional analysis on DB2 5.0.

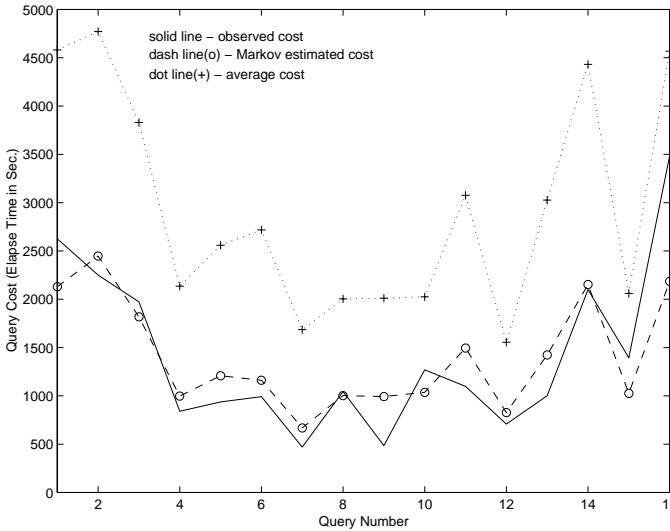


Fig. 10. Estimated costs for test queries via probabilistic approach on Oracle 8.0.

To solve the problem, we have investigated several techniques. In our recent work (Zhu et al., 2000), we employed a multistate query sampling method to develop cost models with qualitative variables indicating the system contention states. For queries experiencing a single state, such developed cost models can be used to directly estimate their costs. To estimate the costs of (large) queries experiencing multiple states, we propose two new techniques, i.e., fractional analysis and probabilistic approach, to estimate their costs in this paper. The first is

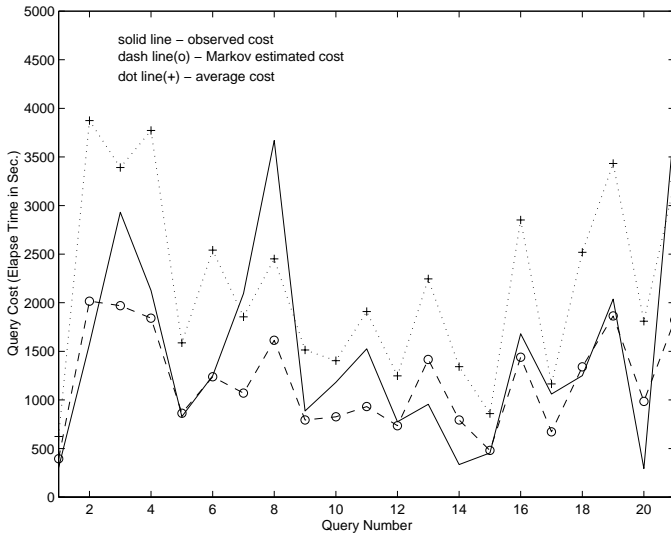


Fig. 11. Estimated costs for test queries via probabilistic approach on DB2 5.0.

suitable for a gradually and smoothly changing environment, while the latter is suitable for a rapidly and randomly changing environment. The cost formulas in both cases are derived. Their estimation errors and properties are analyzed. Note that although the two new techniques make use of the cost models developed by the multistate query sampling method, it is not required; in other words, any method that can estimate the cost of a query in each contention state can be used together with the two techniques proposed in this paper. To validate the effectiveness of the new techniques, we conducted extensive experiments. Our experimental results demonstrate that the presented techniques are quite promising in estimating query costs in a dynamic multidatabase environment. Their derived cost estimates for most queries have relative errors within 30%.

Dynamic environmental factors were ignored in existing cost models for database systems owing to lack of appropriate techniques. The work reported in this paper has shown some promise in solving the problem. However, our work is just the beginning of work needed to be done in the future in order to completely solve all related issues.

Acknowledgements. We would like to thank Thanikachalam Ponnusamy for work on the dynamic environments for real-world application systems. We are grateful to Amira Rahal-Arabi for her help in running some experiments. We would also like to thank Per-Åke (Paul) Larson, Guy M. Lohman, and Wen-Chi Hou for their valuable suggestions and comments on the work reported in this paper. This research was supported by the US National Science Foundation under Grant # IIS-9811980 and the University of Michigan under OPR and UMD grants.

References

Adali S, Candan KS, Papakonstantinou Y, Subrahmanian VS (1996) Query caching and optimization in distributed mediator systems. In Proceedings of the ACM SIGMOD international conference on management of data, pp 137–148, Montreal, Quebec, Canada. ACM Press

- Attaluri G, Bradshaw D, Coburn N, Larson PA, Martin P, Silberschatz A, Slonim J, Zhu A (1995) The CORDS multidatabase project. *IBM Systems Journal* 34(1): 39–62
- Du W, Krishnamurthy R, Shan MC (1992) Query optimization in heterogeneous DBMS. In Proceedings of the 18th VLDB conference, pp 277–291, Vancouver, Canada. Morgan Kaufmann
- Du W, Shan MC, Dayal U (1995) Reducing multidatabase query response time by tree balancing. In Proceedings of the ACM SIGMOD international conference on management of data, pp 293–303, San Jose, California. ACM Press
- Evrendilek C, Dogac A, Nural S, Ozcan F (1997) Multidatabase query optimization. *Distributed and Parallel Databases* 5(1): 77–114
- Gardarin G, Sha F, Tang ZH (1996) Calibrating the query optimizer cost model of IRO-DB, an object-oriented federated database system. In Proceedings of the 22nd VLDB conference, pp 378–389, Mumbai, India. Morgan Kaufmann
- Lee C, Chen CJ (1997) Query optimization in multidatabase systems considering schema conflicts. *IEEE Transactions on Knowledge and Data Engineering* 9(6): 941–955
- Litwin W, Mark L, Roussopoulos N (1990) Interoperability of multiple autonomous databases. *ACM Computing Surveys* 22(3): 267–293
- Lu H, Ooi BC, Goh CH (1992) On global multidatabase query optimization. *SIGMOD Record* 21(4): 6–11
- Naacke H, Gardarin G, Tomic A (1998) Leveraging mediator cost models with heterogeneous data sources. In Proceedings of the 14th international conference on data engineering pp 351–360, Orlando, Florida. IEEE Computer Society
- Parzen E (1962) *Stochastic processes*. Holden-Day, Oakland, CA
- Roth MT, Ozcan F, Haas LM (1999) Cost models DO matter: providing cost information for diverse data sources in a federated system. In Proceedings of the 25th VLDB conference, pp 599–610, Edinburgh, Scotland, UK. Morgan Kaufmann
- Sheth AP, Larson JA (1990) Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys* 22(3): 183–236
- Zhu Q, Larson PA (1994) A query sampling method for estimating local cost parameters in a multidatabase system. In Proceedings of the 10th IEEE international conference on data engineering, pp 144–153, Houston, Texas. IEEE Computer Society
- Zhu Q, Larson PA (1996) Building regression cost models for multidatabase systems. In Proceedings of the 4th IEEE international conference on parallel and distributed information systems, pp 220–231, Miami Beach, Florida. IEEE Computer Society
- Zhu Q, Larson PA (1997) A fuzzy query optimization approach for multidatabase systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 5(6): 701–722
- Zhu Q, Larson PA (1998) Solving local cost estimation problem for global query optimization in multidatabase systems. *Distributed and Parallel Databases* 6(4): 373–420
- Zhu Q, Sun Y, Motheramgari S (2000) Developing cost models with qualitative variables for dynamic multidatabase environments. In Proceedings of the 16th IEEE international conference on data engineering, pp 413–424, San Diego, California. IEEE Computer Society

Author Biographies



Qiang Zhu is an Associate Professor in the Department of Computer and Information Science at the University of Michigan – Dearborn. He received his Ph.D. from the University of Waterloo in Canada in 1995. He also holds an M.Sc. from the McMaster University in Canada, and an M.Eng. and a B.Sc. both from the Southeast University in China. He was a faculty member in Computer Science at the Southeast University and a Visiting Scientist at the IBM Toronto Lab. His research has been funded by competitive sources including the US National Science Foundation and IBM. He has published numerous papers in various journals and conference proceedings. Some of his research results have been included in several well-known database books. Dr Zhu has served as a program/organizing committee member and workshop/session chair for many international conferences. His current research interests include query processing and optimization, multidatabase systems, data mining, and Web database technology.



Satyanarayana Motheramgari is a graduate student in the Department of Computer and Information Science at the University of Michigan – Dearborn. He also works as a software engineer in the Advanced Research Group at Yazaki-Na Inc., Canton, Michigan. He obtained his Bachelor degree with distinction from Osmania University, India, in 1983. He also received a Masters degree from the Indian Institute of Technology, Kanpur, India, in 1985. He was a faculty member at the Regional Engineering College, Warangal, India, from 1987 to 1995. His research interests include stochastic analysis of complex systems, database query cost modeling and optimization.



Yu Sun is a Research Associate in the Department of Computer and Information Science at the University of Michigan – Dearborn. He received a B.Sc. degree in Mathematics in 1982 and an M.Eng degree in Computer Science and Engineering in 1988, both from the Southeast University, China. He was a faculty member in Computer Science at the Southeast University from 1982 to 1998. He has participated in numerous projects of database research and applications including multidatabase systems, (traditional) distributed database systems, object-oriented database systems, and computer-integrated manufacturing systems. His current research interests include query processing and optimization, database statistical techniques, system architecture and prototyping.

Correspondence and offprint requests to: Qiang Zhu, Department of Computer and Information Science, The University of Michigan - Dearborn, Dearborn, MI 48128, USA. Email: qzhu@umich.edu