Responsive

This paper presents the advantages of e-Learning education and focus on query processing in distributed databases. For this, we present the factors that influence distributed query optimization and examine the optimization strategies of query execution. The goal of this paper is to present strategies of optimizing query execution in e-learning systems used mainly by universities with geographically distributed locations, in order to obtain a system response time as low as possible and to minimize the total cost of implementation.

Keywords: e-Learning; distributed databases; queries; factors; strategies

1. Introduction

The evolution of information technology generated also a tremendous evolution of database systems and in this context distributed database technology has changed the centralized point of view by offering major advantages.

A distributed database (DDB) is a collection of shared data, logically interrelated, which are geographically distributed in a computer network (Iacob, 2010).

The distributed database management systems (DDBMS) is the software that ensures working with DDB and supplies an access mechanism which makes their distribution transparent to all users.

A distributed database system is a database system which is fragmented or replicated on the various configurations of hardware and software, located usually at different geographical sites within an organisation (Beynon-Davies, 2004).

The evolution of modern information technologies and the Internet have changed traditional education and training methods resulting in reorganization and transformation of those in order to provide opportunity to offer professional training to students, irrespective of age, sex or geographical area. Using new technologies, any student can communicate with specialists in their area of interest or other colleagues for exchanging information, knowledge and experience.

E-learning is the interaction between teaching and learning, information technology and communications, covering a wide spectrum of activities from computer-aided education to education conducted entirely in an online manner.
Modern learning programs and distance learning are supported and distributed through the widespread use of e-Learning environment, learning based on information and communication technology that covers a wide spectrum of educational activities, from traditional learning methods, to practical combination of traditional and electronic learning, to completely on-line learning.

E-Learning environment is practically a virtual learning environment that uses digital technology and the Internet in order to give to students concerned about their professional development a set of tools for teaching, learning, assessment and communication. The platform offers various resources for managing groups of students, educational programs and also provides flexible ways to measure students’ knowledge.

2. Distributed queries

A user interacts with the database by providing read and writes requests in the database. These read and write operations are grouped into transactions. In a computer network it is very important to not allow the unauthorized persons to gain access to the information sent between computers (Defta, 2010).

The global users working with DDB would work as with a centralized database:
- make requests (global transactions);
- requests are assessed, decomposed and delivered to distributed execution supervisor;
- the supervisor sends the requests to nodes in which there are local databases that will be queried;
- queried databases send replies to the supervisor.

A methodology of distributed query processing is shown in Figure 1, (Ozsu & Valduriez, 2011).

The input is a query on distributed data expressed in relational calculus. Four main layers are involved to map the distributed query into an optimized sequence of local operations, each of them acting on a local database. These layers perform the functions of query decomposition, data localization, global query optimization, and local query optimization. Query decomposition and data localization correspond to query rewriting. The first three layers are performed by a central site and use global information.

Local optimization is done by the local sites.

![Figure 1. Distributed Query Processing Methodology](image)

Distributed query processing contains four stages which are query decomposition, data localization, global optimization and local optimization.
• **Query decomposition**: in this stage we are giving Calculus Query as an input and we are getting output as Algebraic Query. This stage is again divided in four stages they are Normalization, Analysis, Simplification and Restructuring.

• **Data localization**: in this stage Algebraic query on distributed relations is input and fragment query is output. In this stage fragment involvement is determined.

• **Global optimization**: in this stage Fragment Query is input and optimized fragment query is output. Finding best global schedule is done in this stage.

• **Local optimization**: Best global execution schedule is input and localized optimization queries are output in this stage. it contain two sub stages they are Select the best access path, Use the centralized optimization techniques.

### 2.1. Distributed query optimization

Distributed query optimization is defined as finding efficient execution strategy path in distributed networks. A query running against a distributed database environment (DDBE) will have to go through two types of optimization. The first type of optimization is done at the global level, where communication cost is an important factor. The second type of optimization is done at the local level. This is what each local DBE performs on the fragments that are stored at the local site, where the local CPU and the disk input/output (I/O) time have the main importance. Almost all global optimization alternatives ignore the local processing time. When these alternatives were being developed, it was believed that the communication cost was a more dominant factor than the local processing cost. Now, it is believed that both the local query cost and the global communication cost are important to query optimization (Rahimi & Haug, 2010).

The problem of optimizing execution of distributed queries must be addressed in order to obtain a system response time as low as possible in terms of minimizing the total cost of implementation.

**The total cost of a query execution** in a distributed system, formula (1), includes:

- CPU – the cost of CPU processing for the execution of data operations in main memory,
- I/O - the costs of accessing (input/output) physical data on the respective support,
- Communication costs between nodes associated with sending and receiving messages between nodes (CMSP) and the transfer of data between nodes (CTR).

$$ \text{Total\_cost}=\text{CCPU}\times \text{no\_instructions}+\text{CI/O}\times \text{no\_I/O}+\text{CMSP}\times \text{no\_messages}+\text{CTR}\times \text{data\_quantity} \quad (1) $$

**The response time of the distributed system**, formula (2), is calculated from the time the query starts until it receives response from the system.

$$ \text{Response\_time}=\text{TCPU}\times \text{no\_instructions}+\text{TI/O}\times \text{no\_I/O}+\text{TMSP}\times \text{no\_messages}+\text{TTR}\times \text{data\_quantity} \quad (2) $$

The response time minimization is achieved by increasing the parallelism degree of distributed query execution. This does not necessarily imply that the total cost will be minimized. Contrary, the total cost may increase when trying to increase the degree of parallelism of the execution and transmission. On the other hand, minimizing the total cost involves better use of resources in the detriment of response time which will increase. In practice, it is desirable to do a compromise between the two objectives.

### 2.2. Strategies to optimize the execution of queries

The query optimization refers to that process that finds the best query execution strategy for a variety of alternatives. An execution strategy for distributed query can be described on relational algebra operations and communication primitives (send / receive operations) necessary to transfer of data. In a distributed model, between query decomposition and their optimization interfere two additional steps: **data location** and **global query optimization**.

**The data location** is made easily in the model - using information from the catalog database. Depending on the location of the required fragments, in order to to solve a read query q received by node N, we have two situations:
Case 1. All fragments \( \{ d_i, \mid d_i \rightarrow q, i=1, \ldots, k \} \), are stored on the node N.

Case 2. One or more fragments are not stored on the node N, they are stored in other nodes.

Query optimization uses three components: a search space, a cost model (defined in terms of units of time generally refers to disk space, the number of operations I/O to buffer space, CPU cost, communication cost, etc.) and a search strategy.

We consider a distributed database system with a set of sites \( S = \{ s_1, \ldots, s_n \} \). A query Q is represented as an ordered sequence of subqueries \( Q = \{ q_1, \ldots, q_m \} \). Each subquery \( q_i \) is the maximum processing unit that accesses a single base relation and communicates with its neighboring subqueries. Each site \( s_i \) has a load, noted by load \( (s_i) \), which reflects the number of queries currently submitted. The load can be expressed in different ways, e.g. as the number of I/O bound and CPU bound queries at the site. The average load of the system is defined as:

\[
\text{Avg\_load}(S) = \frac{\sum_{i=1}^{n} \text{load}(s_i)}{n}.
\]

The balance of the system for a given allocation of subqueries to sites can be measured as the variance of the site loads using the following unbalance factor:

\[
\text{UF}(S) = \frac{1}{n} \sum_{i=1}^{n} (\text{load}(s_i) - \text{Avg\_load}(S))^2.
\]

Given:
1. a set of sites \( S = \{ s_1, \ldots, s_n \} \), with the load of each site;
2. a query \( Q = \{ q_1, \ldots, q_m \} \); and
3. for each subquery \( q_i \) in Q, a feasible allocation set of sites \( S_q = \{ s_1, \ldots, s_k \} \), where each site stores a copy of the relation involved in \( q_i \);

the objective is to find an optimal allocation on Q to S such that:
- UF(S) is minimized, and
- the total communication cost is minimized.

The algorithm that finds near-optimal solutions in a reasonable amount of time consists of the following steps. The first heuristic (step 1) is to start by allocating subqueries with least allocation flexibility, i.e. with the smaller feasible allocation sets of sites. Thus, subqueries with a few candidate sites are allocated earlier. Another heuristic (step 2) is to consider the sites with least load and best benefit. The benefit of a site is defined as the number of subqueries already allocated to the site and measures the communication cost savings from allocating the subquery to the site. Finally, in step 3 of the algorithm, the load information of any unallocated subquery that has a selected site in its feasible allocation set is calculated again.

Algorithm:

**Input:** Q: \( q_1, \ldots, q_m \);
Feasible allocation sets: \( S_q_1, \ldots, S_q_m \);
Loads: \( \text{load}(S_1), \ldots, \text{load}(S_m) \);

**Output:** an allocation of Q to S

**Begin**

for each \( q \) in Q do

\( \text{compute}(\text{load}(S_q)) \)

endfor

while Q not empty do

\( a \in q \in Q \) with least allocation flexibility; \{select subquery a for allocation\} (1)

\( b \in s \in S_a \) with least load best benefit; \{select best site b for a\} (2)

Q\( \leftarrow Q-a \);

**End**
\{\text{calculate loads of remaining feasible allocation sets}\} (3)

\textbf{for} each \( q \in Q \) where \( b \in S_q \) \textbf{do}

\text{compute}(\text{load}(S_q))

\textbf{endfor}

\textbf{endwhile}

\textbf{end}

2.3. \textit{Factors that influence the optimization of distributed queries}

- \textbf{The system security} and implications of data access authorization: what data someone can access and on what node the authorization routines should be implemented. The intercalation of the authorization routines in the execution flux of the operations in the system will increase its response time to queries.

- \textbf{The availability of resources}. At some point one of the nodes may not be available for querying, or the network may be down. This increases the response time and also the total cost, or even worse, may cause the query execution to fail if the fragment containing the desired data is not replicated at another node available at that time.

- \textbf{The integrity constraints} declared in the global database level help optimizing the queries addressed to the system. In this way it is possible to remove some branches from the query tree because of these integrity constraints.

- \textbf{The implementation and the usage of communication network}. Network traffic can vary from day to day, even from hour to hour, knowing that there are times of the day with network load peaks and also low traffic periods. Moreover, as the network undergoes changes of any kind, optimization solutions must be found.

Those are some aspects with significant impact on response time and hence the total cost of implementation of a distributed query. Also, while the number of factors that influence the performance of applications increases, the optimization of the system becomes very complex.

3. \textit{Conclusion}

Processing queries ensures the conversion of user transactions, minimizes the cost of data transmission and the cost of local processing those data and optimize these operations.

The optimization of queries in distributed systems is a complex activity that depends on many factors. In a certain percent it is performed by the DBMS, but there are situations when the user applications must contain algorithms for the query optimization. If fragmentation of the database is done correctly, most queries will run at local level.

Assuming that the future of education will be based on the freedom to choose an education form according to the needs of every person that will remove physical barriers, the development of \textit{e-learning} (distance learning) will become certainly a priority to current modern education process.

The main objective of improving the e-Learning technology consists in implementation of educational applications that will offer greater flexibility, better reliability, a new set of features and also increased communications security over Internet.

\textbf{References}


