Henryk Josiński
Silesian Technical University
Institute of Computer Science
Akademicka 16, PL–44–101 Gliwice, Poland
phone: (48 32) 237 13 39, fax: (48 32) 237 27 33
H.Josinski@zti.iinf.polsl.gliwice.pl

Dynamic Query Optimization and Query Processing in Multidatabase Systems

1. Introduction

The multidatabase system (MDBS) approach, as a solution for integrated access to information distributed among diverse data sources, has gained a lot of attention in recent years. The multidatabase system is a database system which integrates pre–existing databases allowing the users to access simultaneously database systems (DBMSs) formulating a global query based on a global schema.

The component DBMSs are assumed to be heterogeneous and autonomous. Heterogeneity refers to different user interfaces, data models, query languages, and query optimization strategies [5]. Local autonomy means that each DBMS retains complete control over local data and processing. As result of this, its cost model may not be available to the global query optimizer.

When a global query is submitted, it is decomposed into two types of queries [1]:

- *subqueries*, operating on sharable data items from local databases,
- *assembling queries*, consisting of, among others, a set of *assembling operations* (e.g. join, outerjoin, union), that combine the partial results returned by subqueries forming the final answer. As a consequence of local autonomy only the assembling queries can constitute the subject of optimization.

If the global query language is different from the query language of a component DBMS, then the subquery must be translated. Partial results that are not in the common format need to be converted into it. In the end, data is assembled into the final result.

The significant problems related to query processing in MDBSs are the following:

- global schema integration,
- global query decomposition,
- aspects of global query optimization: selectivity estimation for various operations, cost models, and assembling operations scheduling,
- unexpected delays in the arrival of information from a component DBMS,
- parallelization of query processing.

Dynamic query optimization gives the possibility of multiple modification of an execution plan or suggests generating limited to the planning of the next elementary operation only, while processing a query. In this way, the course of processing will be adapted to the current circumstances. This approach seems to be the proper one for the efficient query processing in multidatabase environment.

2. Related Work

The dynamic query optimization scheme described in [2] has been implemented in the multidatabase system MIND [3] whose architecture is based on OMG's Object Management

Architecture. This approach uses the available partial results at run–time, instead of producing an execution plan based on the estimated execution time of subqueries.

The on–the–fly modification of the execution plan in response to unexpected delays in the arrival of partial results was described in [4].

The results of the other chosen researches into the query optimization in MDBSs are the following:

– a distributed query optimization algorithm was suggested in [5] and represents the idea of delegating the evaluation of the execution cost of the elementary steps in an execution plan to the DBMS where the computation is performed,
– methods such as statistical sampling [6] or synthetically created calibrating database [7] were proposed to deduce the coefficients in cost formula for the underlying DBMS,
– tree balancing algorithms were presented in [8] and introduce the idea of transformation of left–deep trees to bushy ones, in order to increase the concurrency and independent processing in query execution.

3. An Experimental Distributed Database Management System

For researching into problems connected with integrated access to distributed information, an experimental distributed database management system (EDDBMS) [10] was constructed. The system, the basic structure of which is depicted in Figure 1, forms an environment for developing and testing algorithms related to the following issues: global query processing, concurrency control, distributed transaction commitment, and maintenance of data replication. The system is assumed to run on a cluster of workstations connected by the Ethernet local network.

The EDDBMS consists of three fundamental components:

– *data processor*, managing data located on the local node,
– *application processor*, which coordinates the access to data items on various nodes, using metadata stored in the global dictionary,
– *communication software*, providing the communication primitives used by the application processor to transmit queries and data among various nodes.

The researches have focused on the implementation of the application processor, using the chosen database management systems (*Ingres*, *Postgres95*, *MS SQL Server*) as the data processors. The application processor access to the contents of a local database goes through the relevant data processor.

There are the following elements of the application processor:

– clients,
– local servers,
– replication manager,
– local clients.

Any number of client processes and one local server process for each data processor used, can simultaneously run in the system. Independent processes running on the network communicate through virtual shared memory (VSM) called *tuple space*, passing data units called *tuples*.

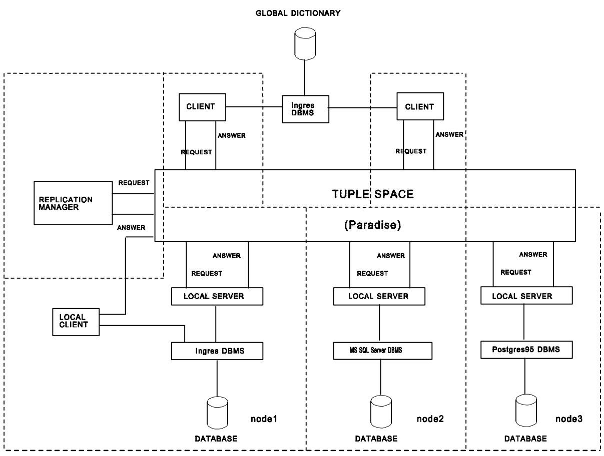

Figure 1. The basic structure of the EDDBMS.

The implementation is based on the distributed computing environment *Paradise*.
The *client* is responsible for:

- syntactical and semantic analysis of user queries,
- decomposition of the query into a set of subqueries and one assembling query,
- generating a preliminary execution plan,
- sending subqueries to local servers, supervising scheduling and execution of the assembling operations, reception of the final result and its presentation,
- control of the concurrent access to the data, including global deadlock detection and resolution, through an additional locking mechanism called *global logical locking* which is independent of the locks used by data processors,
- managing the distributed transactions.

The functions of the *local server* are defined as follows:

- translation of the SQL queries into the language characteristic for the data processor (if required),
- managing concurrent subtransactions in cooperation with the local DBMS.

The *replication manager* module maintains replicated data items. The *local client* module manages the local transactions operating on replicas. The communication between these two modules is achieved by mechanisms of active databases: triggers, database procedures, database events, and via tuple space.

New system components are intended to be added to the EDDBMS. The new module will perform an assembling operation ordered by the client supervising the processing of the global query. The selection of the given execution node will be the consequence of, among others, current network throughput and loads of the local nodes.

4. Dynamic Query Optimization in the EDDBMS

According to the generic framework of the dynamic query optimization [9] the following strategy is suggested for the EDDBMS. While decomposing a global query, a preliminary execution plan is produced by a client using the statically known metadata. The plan leaves some degree of freedom, related to assembling operations scheduling. After sending subqueries to local servers the first partial results are expected. The execution of the assembling operations proceeds in several phases interrupted by events that can cause modification of the plan. The arrival of the partial results and current messages about the state of the network elements are examples of such events. When a similar situation occurs, a planning phase starts. Its task is generating a plan for the assembling operations not executed yet, including the choice of their execution nodes according to the current knowledge of metadata.

5. Future Work

The research work will concentrate on the following topics:

- participation of the mobile agents in query processing,
- modification of the assembling operations scheduling algorithms, taking into consideration both selectivity estimation methods for various operations and cost formulas,
- modification of the assembling operations realization algorithms, taking into account parallelization possibilities given by the VSM model,
- design and implementation of the new EDDBMS components.

Comparative tests of the ideas being examined will be performed by means of simulation. Practical verification of the suggested solutions will take place after their implementation in the EDDBMS. Global query processing techniques are also intended to be tested in various distributed computing environments. The work will comprise the extension of the system as a result of object–oriented DBMSs' integration as well.

References

[1] C.T. Yu, W. Meng: *Principles of Database Query Processing for Advanced Applications*. Morgan Kaufmann Publishers, Inc., 1998.

[2] F. Ozcan, S. Nural, P. Koksal, C. Evrendilek, A. Dogac: *Dynamic Query Optimization in Multidatabases*. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 1997.

[3] C. Evrendilek, A. Dogac, S. Nural, F. Ozcan: *Multidatabase Query Optimization*. Distributed and Parallel Databases, 5, 1997, pp. 1–39.

[4] L. Amsaleg, M. Franklin, A. Tomasic, T. Urhan: *Scrambling Query Plans to Cope with Unexpected Delays*. Proc. of the Int. Conf. on Parallel and Distributed Information System, Miami Beach, USA, 1996.

[5] S. Salza, G. Barone, T. Morzy: *Distributed Query Optimization in Loosely Coupled Multidatabase Systems*. Proc. of the 5th Int. Conf. on Database Theory, Prague, Czech Republic, 1995, pp. 40–53.

[6] Q. Zhu, P–Å. Larsson: *A Query Sampling Method for Estimating Local Cost Parameters in a Multidatabase System*. Proc. of the 10th Int. Conf. on Data Engineering, Houston, USA, 1994, pp. 144–153.

[7] W. Du, R. Krishnamurthy, M–Ch. Shan: *Query Optimization in a Heterogeneous DBMS*. Proc. of the 18th VLDB Conf., Vancouver, Canada, 1992, pp. 277–291.

[8] W. Du, M–Ch. Shan, U. Dayal: *Reducing Multidatabase Query Response Time by Tree Balancing*. SIGMOD Int. Conf. on Management of Data, San Jose, USA, 1995, pp. 293–303.

[9] L. Bouganim, F. Fabret, P. Valduriez, O. Kapitskaia: *CARAVEL Project*. Via Internet: http://www-rodin.inria.fr/Eaction_dynamique.html

[10] P. Bajerski, M. Chłopek, K. Harężlak, H. Josiński: *Environment for Distributed Database Management Algorithms Evaluation Based on Virtual Shared Memory*. Proc. of the 16th IASTED Int. Conf. "Applied Informatics", Garmisch–Partenkirchen, Germany, 23–25.02.1998, pp. 184–186.