Distributed Database System Query Optimization Algorithm Research

Fan Yuanyuan
Department of Computer and Information Engineering，
Jiaozuo Teachers College
weniss@163.com

Mi Xifeng
Department of Computer and Information Engineering，
Jiaozuo Teachers College
mixifeng@126.com

***Abstract*-Query optimization is an important part of database management system. In this paper, through the research on query optimization technology, based on a number of optimization algorithms commonly used in distributed query, a new algorithm is designed, and experiments show that this algorithm can significantly reduce the amount of intermediate result data, effectively reduce the network communication cost, to improve the optimization efficiency.**

Keywords-distributed database; query optimization; optimization algorithms

I. INTRODUCTION

In recent years, with the development of computer network and database technology, distributed database is more and more widely used; with the expanding application, data queries are increasingly complex, the efficiency requests are increasingly high, so query processing is a key issue of the distributed database system.

II. DISTRIBUTED DATABASE SYSTEM OVERVIEW

Distributed database system is a physically dispersed but logically centralized database system, is a combination of computer network and database system[1]. Physically dispersed, referring to that the data composed of distributed database is distributed in different computers, and each site in the network has the ability of independent processing, to perform local application. Logically centralized, referring to that the site is a logical whole and managed by a distributed database management system, each node performs the overall application through the network communication subsystem.

Distributed database system includes not only a distributed database management system and distributed database, but also contains more actual content. It can be run and stores, maintains the data by way of distributed database, as well as provides data and information to the applied network environment systems. The system architecture is shown as in figure 1.

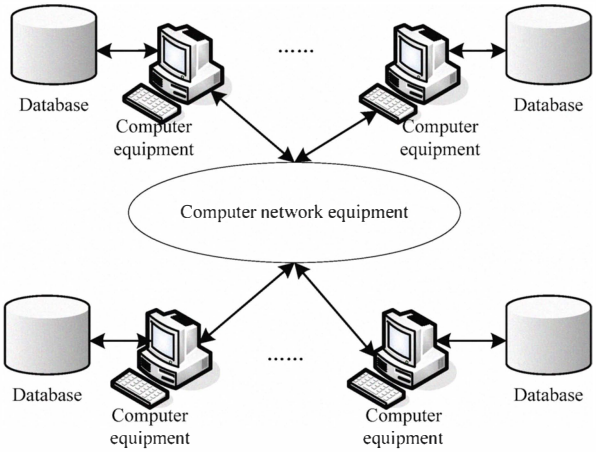

Figure 1. Distributed database system architecture

Distributed database system consists of the following elements:

1) Multiple computer equipment, connected by computer network.

2) Computer network equipment, a set of software of network communications.

3) Distributed database management system includes GDBMS, LDBMS and CM, and in addition to a global user interface connected by the GDBMS[4]*, also have the autonomy site user interface connected by the site DBMS, has a separate site directory/dictionary.*

4) Distributed database includes the global database, local database and autonomous site database.

5) Distributed database manager can be divided into two, one for the global database manager, and the other for the local or autonomous site database managers, which are collectively known as the local database manager.

6) Distributed database system software document, this is a set of software documents matched with the software, as well as a variety of system instructions and documents.

According to the principle to principle the distributed database system architecture, it can be divided into two categories: isomorphic distributed database management systems and heterogeneous distributed database management systems. These two can meet different application requirements.

Distributed database management system should be able to support the three basic functions of the distributed database system[3]. The first is the remote database operations of the

978-1-4244-5540-9/10/$26.00 ©2010 IEEE

application, the second is the full or partial transparence of it, and the third is the management and control of the distributed databases.

Therefore, in addition to the functions of a centralized database management system, it must also provide maps of each site database for centralized management and control.

III. QUERY PROCESSING AND OPTIMIZATION

A. *Distributed query processing and optimization process*

Query optimization has been one of the research focuses of database area, although many researchers have done a lot of work, but the not commensurate with the successful application of relational database technology in data processing is multi-join query optimization has been a problem not well resolved in relational database systems.

Distributed system first checks whether there is such a local database after the arrival of the user queries, if there is, for a local implementation; if there is no, then the global query processing module is to select a optimal node to process this query according to the table information, that is, select a database node which has the database and has the minimum query cost of the controlled table, and to establish a connection with the optimization node to send the query command to it for implementation, while returning its IP to the client. The client will immediately re-establish the connection with the new IP after receiving the feedback information. When the new server node completes the query, it will return the results to the client.

The structure in figure 2 shows the various stages of the distributed query processing, that is, the query begins from the user input of SQL statement[7], then the parser translates and optimizes to make it become a query plan which can be performed, and then the line resource generator will execute this plan, finally obtains the query results.

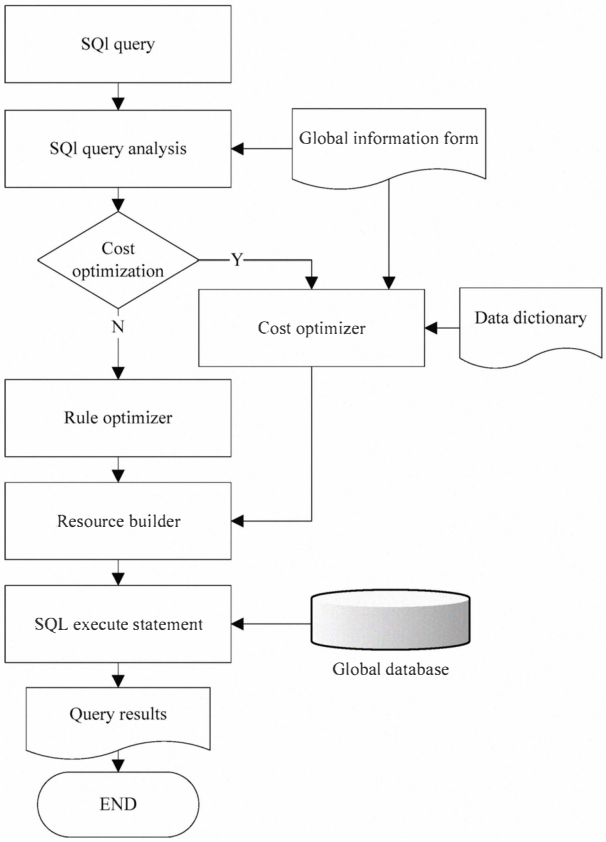

Figure 2. Distributed query optimization processing

1) Parser. At this stage, the query statement is parsed into an internal inquiry expression for later stages of processing.

2) Rule-based optimizer. It is to delete some redundant information according to the rule, simplify the expression, sub-queries and views, Statement in this step is carried out a simple optimization[2]*. In the distributed database system, query rewriting device can also select those database tables to participate in the query.*

3) Cost-based query optimizer. It is responsible for the major optimization work, in accordance with the statistical information of the database to select the operation order involved in the implementation of query statement, how to allocate memory for each operation.

4) Dictionary table. It is the statistical description of the database, stores the database model information, such as table definitions, views, user-defined types and functions, constraints and so on.

5) Implementation plan. It is to describe how a query request to be specifically executed, almost all database systems use the same method to express the query plan-syntax tree. Each tree leaf node is the relationship needs to operate; tree middle node is the operator for each operation in the query plan, and each operator represents a particular operation.

6) Global information table. It is to store all the information required by the distributed points to rewrite all stages by using analysis. It stores the database distribution information, global segmentation, reconstruction rules and form physical storage location.

7) Line resource generator. It is to access to the database according to the finally determined implementation plan to obtain query results and return to the user.

B. Distributed query optimization algorithms

1) Optimization algorithm based on relational algebra equivalence transformation

It is to use heuristic optimization method to optimize the relational algebra expression. And in the relational algebra expression, the most time and space operation is Cartesian product and join operations, so, the implementation of selection and projection operations as early as possible, to avoid the direct Cartesian product operation, then combines a series of selection and projection before and after it together to reduce the size of intermediate relations, thus to achieve optimization.

The basic idea of this algorithm is: to convert the query problem into relational algebra expression, analyze the obtained query syntax tree, and optimize according to equivalence rules. The algorithm first uses relational algebra equivalence transformation to raise the connecting and merging operations in the query tree as much as possible, while moving the selection and projection operations down to the fragment definition [6]. Then to determine the horizontal or vertical slice, if the horizontal slice, to compare with the slice and the selection conditions to remove the contradictory fragments, if only a fragment left then to remove one parallel operation. f the vertical slice, then to compare with the fragment property set and the attribute set involved in projection operation to remove all the unrelated fragments. If only one vertical segment left then to remove one connection operation, thus to achieve the purpose of optimization query.

2) Optimization algorithm based on semi-connection operations

The basic idea of this algorithm is: data in the network is transmitted as the entire relationship or a fragment, which is obviously a redundant way. When a relationship transferred to another venue, not every data is involved in connection operation or useful. Therefore, the data is not involved in the connection or useless data needs not to be transmitted circularly in the network [5]. The basic principle of this optimization strategy is to use semi-connection operation to only transmit the data involved in the connection in the network as far as possible. The steps of using semi-connection algorithm to optimize the connection query as shown in figure 3.

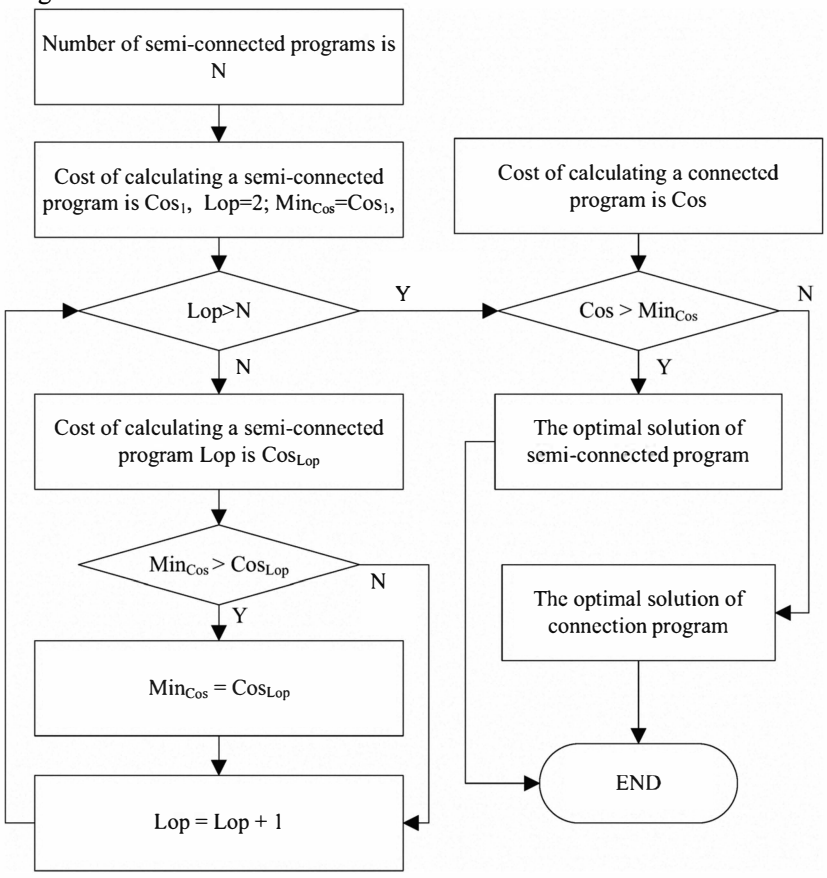

Figure 3. Steps of using semi-connection algorithm to optimize connection query

IV. SEMI-CONNECTION QUERY OPTIMIZATION ALGORITHM IMPROVEMENT AND EXPERIMENT

A. *Semi-connection query optimization algorithm improvement*

General semi-connection query method is to perform semi-connection in accordance with the connection order between relationships, its advantage is simple to process, and deficiency is not considered how to optimize the semi-connection order of the sub-query to further reduce the network communication costs [8] . The algorithm in this paper is presented according to the deficiency of general algorithm, that is, through the cost estimate and semi-connection to improve on how to further reduce the network cost of sub-query. In this paper, the data of the intermediate results generated by performing all the sub-queries is regarded as the decisive factor of the network cost, and the optimization benefits of the multi relationship semi-connection query optimization algorithm is agreed to the relative benefits derived from the comparison with the ordinary algorithms.

B. *Experimental results*

This experiment uses sets of PC memory distributed databases, the operating system with Windows 2000 Server, database management system with Microsoft SQL Server 6.0, and the experimental data is the goods data information of a certain large-scale supermarket. In the simulation of distributed environment, a comparison between the traditional method and the semi-connection method based on repeat query is carried out, the experimental results as shown in table 1.

TABLE I. EXPERIMENTAL RESULTS

Query scale	Query efficiency of traditional algorithm	Query efficiency of optimized algorithm
1000	0.47	0.48
2000	0.47	0.49
5000	0.45	0.52
8000	0.43	0.56

The semi-connection algorithm based on repeat query is mainly to use the middle connection results redundantly stored in the repeat query namely the first step of semi-join operations to reduce computation and transmission, so that to improve query efficiency. Of course, the second step is also can be kept, or even the final results of the connection, but this would significantly increase storage space, while reducing the number of re-use. Through simulation experiments, the correctness of the algorithm is verified, and the efficiency is improved.

V. CONCLUSION

General semi-connection query method is to perform semi-connection in accordance with the connection order between relationships, its advantage is simple to process, and deficiency is not considered how to optimize the semi-connection order of the sub-query to further reduce the network communication costs. This paper designs a new semi-connected database query algorithm, which has the data of the intermediate results generated from the implementation of all sub-queries as the decisive factor of network cost, and defines a function to determine the optimization benefits of this algorithm. Experimental results show that in the distributed query optimization, and comparing with the general semi-connected algorithms, the improved semi-connection query optimization algorithm has higher optimization efficiency, significantly reduces the amount of intermediate result data, and effectively reduces the total cost of the network communications.

REFERENCES

[1]. Cyrus Shahabi, Latifur Khan, Dennis Mcleod. A probe based technique to optimize join queries in distributed internet bases, Knowledge and Information Systems.2000, 2.

[2]. Syam Menon. Allocating fragments in distributed databases [J]. IEEE Transactions on Parallel and Distributed Systems, 2005, 16: 577-585.

[3]. D.Kossmann.T he State of the Art in Distributed Query Processing [J].ACM Computing Surveys, 2000, 32(4):422-469.

[4]. D.Kossmann,K.Stocker. Iterative Dynamic Programming: A New Class of Query Optimization Algorithms [J].ACM Transactions on Database Systems, 2000, 25(1):43-82.

[5]. Peng-Yeng Yin,Shiuh-Sheng Yu,Pei-Pei Wang and Yi-Te Wang. A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems [J].Computer Standards & Interfaces, Volume 28, Issue 4, April 2006, 441-450.

[6]. LEEE Chiang, CHIH Chi-sheng, CHEN Yaw-huei. Optimizing large join queries using a graph-based approach, IEEE Trans on Knowledge and Data Eng.2001, 13(2):298-315

[7]. TSAIPSM, CHENALP, Optimizing queries with foreign function in a distributed environment, IEEE Trans on Knowledge and Data Eng, 2002, 14(2):809-824.

[8]. S. Pramanik, D. Vineyard. Optimization Join Queries in Distribute Database. IEEE TSE.2004.14 (9):1319-1326.