Query Optimization Strategies and Implementation Based on Distributed Database

XUE Lin
College of Information
Linyi Normal University
Linyi, China
e-mail: xuelinem@163.com

***Abstract*—In this paper, on the basis of the research on the existing distributed query optimization, against the higher repetition rate, a new query processing program is presented, which completes the data dictionary, the overall model design, and then improves the query flow. Experiments shows that the strategies can improve the efficiency of the query, reduce the amount of middle data, and effectively reduce the total cost of network communication.**

Keywords-Distributed database; SDD_1; query optimization

I. INTRODUCTION

Distributed database system is physically distributed and logically centralized database system, is the product generated from the mutual penetration and organic integration of computer network technology and database technology. Physical dispersion refers to that the data composing the distributed database is distributed to the different computers in the network, and each site in the network has the ability to deal with and can implement local applications. Concentrated in logic refers to that each site is a logical whole which is managed by a distributed database management system, and each site implements the global application through the network communication subsystem.

In distributed database systems, data independence refers to data logical independence, physical independence and distributed independence which is also known as distribution transparency. Distribution transparency refers to that the users do not have to care about the logical sub-chip of the data, the details of data physical location distribution, the consistency of the duplicate copies, as well as do not have to care about what kind of data model the database in local field support. The distribution of transparency makes the system easier and more effective to be used. Distributed database architecture is shown in Figure 1 [5].

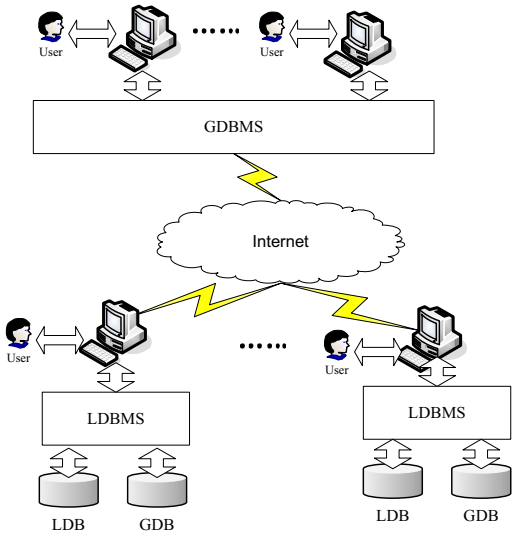

Figure 1. distributed database architecture

In recent years, with the extensive application of the distributed systems, distributed database has become an important part of information processing. Distributed data query processing is the interface of users and distributed database, and then query optimization has become a core problem of the relational database area.

II. DISTRIBUTED DATABASE QUERY OPTIMIZATION ANALYSIS

Query processing and optimization is very important in the relational database system, and also is one of the main research questions of the distributed database. Whether in centralized database or distributed database, a query processing strategy option is on the basis of performing queries expected expense.

A. Distributed database system mode structure

In a centralized database system, in order to enable users to abstractly and logically process the data without having to care about the specific expression way of the data in computer, the centralized database system uses a three-tier mode structure: inside mode, conceptual model and outside mode. And the distributed database is a logical aggregate of the centralized database based on computer network connection. Therefore, the mode structure of the distributed

978-1-4244-4520-2/09/$25.00 ©2009 IEEE

database system adds some new content on the basis of the centralized database system. Generally, in distributed database, the mode structure can be divided into global outside mode, global conception mode, piecewise mode, distribution mode, local conception mode, and local inside mode. The conversion between the various modes can be achieved through multiple images provided by the database management system, the specific structure as shown in figure 2 [3].

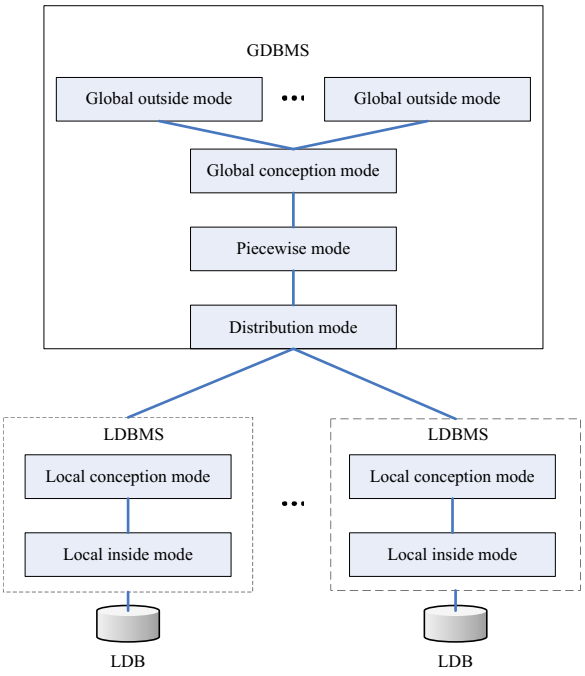

Figure 2. distributed database mode structure

B. *Distributed database query optimization analysis*

Distributed query technology is mainly to translate the global query request the user submitted into the local query request the various interrelated nodes can identify, and return the query results of the various nodes, which includes distributed query processing and distributed query optimization. Distributed query processing mainly researches on the whole distributed query processing process and strategy; Distributed query optimization mainly researches on the query strategy optimization, that is, how to select the program with the least query expense from a variety of programs [1].

In the distributed database, the data distribution to users is generally transparent, when users request for query, they do not have to care about whether or not the involved relationship has been parted, with or without a duplicate and where such issues can be, it can be the same as a centralized database which uses the relationship rather than fragments to express the inquiry. The distributed query processing flow is shown in figure 3 [2].

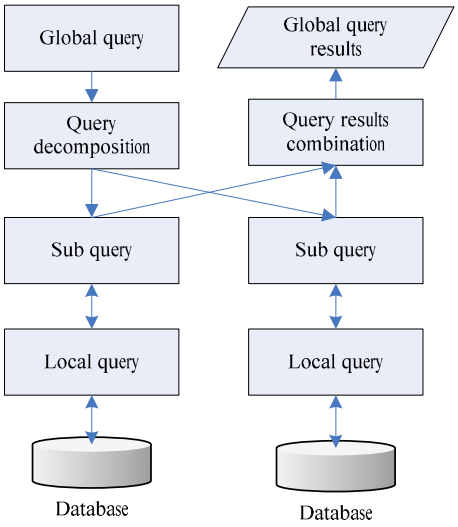

Figure 3. distributed query processing flow

In distributed database systems, two different objectives are often used to consider the query optimization. One objective uses the standard of minimum total cost, and in addition to considering CPU and I/O cost like the centralized database system, the total cost also includes the data network transmission cost [7]. Which is because that the data distribution and redundancy make the communication cost needed by transmitting data among the sites in the query processing should be considered, which will cause the total cost increased. Another objective uses the standard of minimum response time of each query, which has a great signification in the distributed database system. Which is because that the distributed database system is a system composed by multi computers, in which data distribution and redundancy has also increased the possibility of parallel processing, thus the response time of query processing can be reduced and the query processing speed will be speeded up.

In distributed query optimization the two standards are also often used at the same time, according to the difference of the system application, one as the main standard, and another as a complementary standard. For example, a implementation program with a minimum total cost may be firstly found, and then the program should be amended under the conditions of without increasing the total cost to achieve the shortest response time. In some cases, the optimization objectives of the query processing are to reduce communication costs and at the same time to reduce response time, then, the algorithm often requires making trade-offs between the two [4].

III. OPTIMIZATION DESIGN AND IMPLEMENTATION OF DISTRIBUTED DATABASE QUERY

A. *Optimization design of distributed database query*

Based on the above analysis, in this paper a distributed database query optimization process flow is given, as shown

in figure 4. Distributed data user module analyzes the user query request according to the query optimization criteria the user selected and thus the corresponding query processing method will be confirmed. Syntax analysis module analyzes the query sentence the user sent and generates the corresponding query tree. Query tree transformed module will convert the query tree received from the anterior stage into the inside expression which is another semantically equivalent and more conducive to optimizing module optimization. Optimization module will process the local query tree according to query optimization strategy to make the total cost smallest. Order processing module will distribute the mission to the corresponding server and return the server processing results to the user. If the operation related to global or local data dictionary, there will be corresponding operations on the data dictionary, in which information statistical operation will update the query sentence table in local data dictionary in accordance with the query frequency; information update operation will update the global data dictionary in accordance with the local data dictionary and the latest changes of the local server performance, and inform other sites through information broadcast operations.

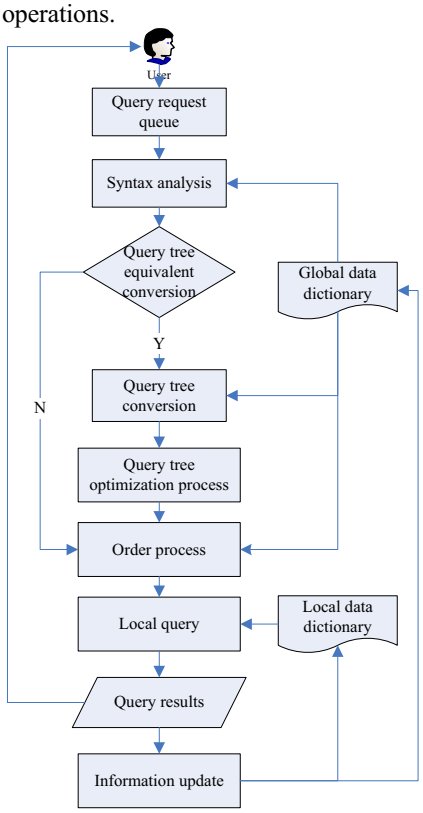

Figure 4. distributed database query optimization process flow

B. *Implementation of distributed database query optimization design*

1) *User module*

Users have a variety of the query results: some users want to get the exact query results, and always want the time of the last query results as short as possible, some users want to quickly get the first query result instead of concerning the time needed to complete the entire inquiries, so this optimization program set in the information query interface that different query result return ways can be selected: show all results, some of the results display first. If the user selects that some of the results display first, he will be provided a control mechanism which will control the current return status of the implementing query. Pre-setting the query optimization criteria can be selected in the information query interface will make the transmission cost lowest and the response time shortest; if the user selects the lowest transmission cost for the query optimization standard, the semi-join query algorithm will be adopted for query; if the user selects the shortest response time for query optimization standard, the pre-set parameters will be obtained and SDD-1 query algorithm will be adopted for query [6].

2) *Syntax analysis module*

The mission of syntax analysis module is to carry though lexical analysis and syntax analysis for the global query sentence, on the one hand, the global table of the data needed can be obtained, and on the other hand, the query sentence can be converted into syntax tree which will be done semantic analysis then. After syntax analysis module receives the user's query sentence, it will firstly send them to the lexical scanner which will separate them into the input composed by a number of words, convert into uniform internal expression and send to the syntax parser to deal with, including: organized sentence sequence input; spell words according to the rules and convert into binary form; delete spaces and useless symbols; find lexical wrong words. Syntax parser analyzes the syntax, obtains the corresponding Select data structure and transmits it to the semantic checker to check whether the involved tables, views, data arranges and other database objectives are effective, check whether the operations are the same type, complete the necessary type conversion work and transmit the Select data structure with type information to the query tree conversion module.

3) *Query tree conversion module*

The mission of query tree conversion module is to obtain the Select data structure from the syntax analysis module, convert it into the corresponding query tree, and rewrite the query tree for the equivalent but easier to be optimized query tree. In accordance with the Select data structure, the global query tree will be established and the algorithm to establish it as shown in figure 5.

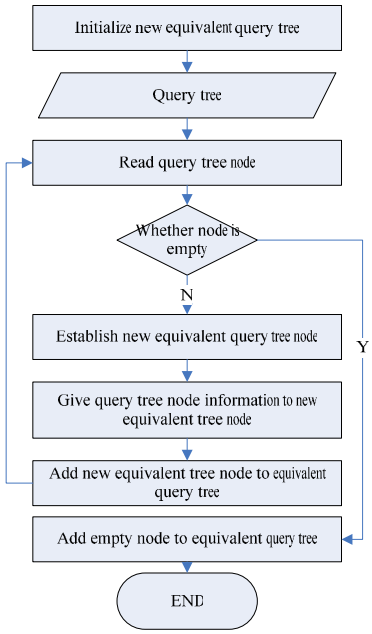

Figure 5. query tree establishment algorithm flow

4) Optimization module

After the query tree conversion module generated a relatively easy to be optimized query tree, the optimization module will receive the query tree, mapping it to a number of physical operator tree, and calculate the cost of the physical operator tree to find the physical operator tree with lowest implementation cost and transmit to the implementation module. Firstly the global query tree will be converted into segment query tree, the process as follows: to traverse the global query tree, when meet leaf node using UN operation to replace it, through the data dictionary to find all the segment relationship of this global relationship and regard these segment relationship as the sub nodes of UN node.

Then the relational algebra optimization of the segment query root will be done equivalent transformation. If the above tree has redundant copies, all the physical copies corresponding to the data segment should be found in accordance with the reflection structure table in the data dictionary, which needs load forecast for the different site server of the same information storage segment to judge the status of the database. According to the site status table in the global data dictionary, the query optimization program should use weight vector method to calculate the indicators of the site status table to obtain the site load performance. According to site capacity and network capacity index to complete decomposition positioning optimization; finally, the segment query tree will be done connection optimization, the optimization algorithm uses the existing SDD-1 and semi-join query optimization algorithm.

5) Data dictionary

Data dictionary is a set of tables and views contained database information, that is, the system directory. Global data dictionary is the basis of the distributed database system to realize centralized query, which essentially realizes a multi-level index sequence table, that is, first level index: database index, all the database in the distributed system are placed; second level index: table index, all the tables of each database are placed. Global data dictionary may be physically in the same network even in the same server, may also be another remote database server. In order to avoid the central node to become the "bottleneck" to reduce storage costs and maintenance costs, this system will store the global data dictionary in multiple servers [8]. In the global data dictionary adding site situation and global query table, and the site situation table is mainly used to feedback the status of all the sites in the system, the global query table is mainly used to show the node where the query results in. In order to ensure the global consistency of the table data, once data in the table has been revised the other nodes should be informed to update the table; and in order to improve the query speed to faster implement optimizing selection, the table must be permanently stored in memory. Adding local sentence table in the local data dictionary is mainly to count the usage of the query sentence.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Weather online monitoring is a system to monitor the weather, as the bottom data collection and management subsystem, embedded real-time database is responsible for regularly pre-processing the data monitored through the sensor and uploading to the upper central database server, and receive real-time commands such as query processing of the central server. To verify the validity of the above design strategy, in the monitoring embedded real-time database system, the above strategy and common strategy is separately tested, and the results as follows:

TABLE 1 MISSION SUCCESSFUL PROBABILITY COMPARING

system load	mission successful probability of common strategy	mission successful probability of improved strategy
0.90	89%	98%
0.92	83%	96%
0.96	75%	92%
0.98	67%	88%
1.00	60%	82%
1.04	52%	75%

TABLE 2 CPU UTILITY RATE COMPARING

system load	CPU utility rate of common strategy	CPU utility rate of improved strategy
0.90	12%	17%
0.92	18%	19%
0.96	23%	21%
1.00	29%	25%
1.05	35%	27%
1.20	43%	29%

On the basis of the research on the relevant knowledge of the distributed database query processing, against the higher repetition rate, a new query processing program was presented to complete the data dictionary, the overall model design, and then the query flow was improved. User module was added in the overall model, which makes user choose

different query optimization standard; query sentence table was added in the data dictionary, which makes the local query results frequently used be stored, thus the transmission of large amounts of data in query can be avoided and the query efficiency will be improved. The disadvantage is that the disposal of each node is more leisured when the system load is very low. At this time, the CPU processing time the optimization spent and the memory spending of the index sequence table are more prominent. The above disadvantage is caused due to the size of the table, the table bigger, the CPU search table spending greater, and the memory consumption greater. Therefore, to reduce the size of the memory table is a key issue.

V. CONCLUSION

In this paper, on the basis of the research on the existing distributed query optimization, against the higher repetition rate, a new query processing program was presented, which completed the data dictionary, the overall model design, and then improved the query flow. Query sentence table was added in the data dictionary, which made the local query results frequently used be stored, thus the transmission of large amounts of data in query can be avoided. Experiments showed that the strategies improved the efficiency of the query, reduced the amount of middle data, and effectively reduced the total cost of network communication.

REFERENCES

[1] Cyrus shahabi, et.al, A Probe-based Technique to Optimize Join Queries in Distribute Internet Databases, Knowledge and information system,2000(2),pp373～385.

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

[3] C.M.Chen, N.Roussopoulos, Adaptive Selectivity Estimation Using Query Feedback, SIGMOD, 2000.

[4] March S T, Rho S. Allocating data and operations to nodes in distributed database design. IEEE Trans, Know Data Eng.2005.7(2):305～316.

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

[6] D.Kossmann. The State of the Art in Distributed Query Pressing [J].ACM Computing Surveys,2000,32 (4):422～469.

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

[8] GuyM.Lohman,Sam5.Iightstone, SMART: Making DB2 (More) Autonomic http://www.almaden.ibm.eom/software/dm/SMART/