

The Semi-join Query Optimization in Distributed Database System

Lin Zhou

Transportation Management College
Dalian Maritime University
Dalian, China
E-mail: zhoubusiness@foxmail.com

Taoying Li

Transportation Management College
Dalian Maritime University
Dalian, China
E-mail: ytaoli@126.com

Yan Chen

Transportation Management College
Dalian Maritime University
Dalian, China
E-mail: chenyan_dlm@163.com

Yingying Yu

Transportation Management College
Dalian Maritime University
Dalian, China
E-mail: uee870927@126.com

Abstract—In distributed database system, the distributed deposition and redundancy data brought convenient to fault recovery, but they make distributed query processing more complicated at the same time. Query optimization and processing is one of the key technologies in distributed database system. It generally uses semi-join operation to improve the time response performance of query and reduce communication cost. This paper briefly described the corresponding concepts and characteristics of distributed database system, summarized the goals of distributed database query optimization, and analyzed the query optimization process based on semi-join operation combined with the practical application. In addition, it introduced a classical algorithm which is used for multiple connection and query optimization based on the semi-join query optimization, the SDD-1 algorithm.

Keywords—distributed database; semi-join operation; query optimization

I. INTRODUCTION

The distributed database develops on the basis of centralized database and it is the collection of data which are related to each other logically and the connection of database and computer network.

Distribution and redundancy of data increases the price of the data transmission cost on the Internet, and makes query processing be more difficult and complex. Then, query optimization and processing become one of the key factors for improving query performance in distributed database. Query optimization and processing adopt reasonable algorithms and precisely reduce the transmission of information as far as possible, which increase the response time performance of the query, and reduce system overhead. The cost is different for different query processing method, which means that the query optimization and processing of distributed database become more and more important.

II. DISTRIBUTED DATABASE SYSTEM

Distributed database connects several databases which are scattered physically but centralized logically and composes database system. Compared with the centralized database, distributed database is thought to increase the system's reliability and availability by increasing the redundant data. The system can operate the copies of the data on another site when the bugs occur in one site. It won't cripple the entire system just owing to one fault. In addition, the user can operate the nearest data copies according the distance so that it can reduce communication cost and improve system performance.

Compared with the centralized database, distributed database system has several main characteristics as follows [1]:

1) Physical distributivity:

The data are not stored intensively in one certain site but dispersedly in multiple sites which are connected with computer network in distributed database system.

2) Logical integrity:

Although the data of distributed database are stored dispersedly on the site, it is a unified whole logically. It can be shared by all users and managed by the distributed database management system.

3) Site autonomy:

In the distributed database system, the data of every site are managed by the local database management system. Every site has the ability of autonomous processing and can perform the local application request.

Base on the basic characteristics mentioned above, we launch distributed database system of other features, such as data distribution transparency. Distribution transparency is the goal of distributed database, and it means that users don't need to pay attention on the fragmentation transparency, the copy transparency, the location transparency and things which data model the local site database support on. Then, the users can operate distributed database just like centralized database. In

addition, there are other characters such as the control mechanism combined with concentration and autonomy, appropriate data redundancy, the distribution of business management and so on.

III. THE GOALS OF QUERY OPTIMIZATION IN DISTRIBUTED DATABASE

In the centralized database system, there are many ways for executing the queries. Its expected cost is mainly the CPU cost and I/O price. It aims at making queries cost lowest.

In the distributed database system, the query optimization includes two parts: the query strategy optimization and local processing optimization. And the query strategy optimization is more important between them. There will be several strategies in the same query due to that the data are stored in different sites. The system resource and response time while each strategy costs is also different. So the expected cost should include corresponding communication cost besides CPU cost and I/O price. Namely the expected cost is the sum of CPU cost, I/O price and communication cost.

The formula of communication costs can be estimated roughly as follows:

$$TC(X) = C_0 + X * C_1 \quad (1)$$

Here, X stands for data transmission amount, usually its unit is b (bit) for computing. C₀ is short for the time which is going to take on the initial communication between two sites, which is determined by the communication system and almost a constant, its unit is s (second). C₁ is for the transmission cost of the unit cost (the reciprocal of data transmission speed), namely the unit data transfer times, and its unit is s/b.

In distributed database system, it usually considers the query optimization in two different targets: One is the minimum total cost, which makes the system resources use less and less and reduce the whole system cost. Another is the shortest response time spent in each query. The physical dispersion and data redundancy also increase the possibility of parallel query processing. Thus, it can shorten the response time and improve the query processing speed. So, it shows that the distributed query processing composes a lot of new contents compared with the centralized query and its complexity has also been shortened. In the practical application, these two standards can be used at the same time, one of which is the main standard and another as auxiliary standard is the specific application environment of the system. For example, the previous implementation makes the cost lower, and then corrects the implementation to make the response time shortest.

IV. THE APPLICATION BASED ON SEMI-JOIN QUERY OPTIMIZATION

If the data with complete relationship were transmitted in the network, it must bring redundancy. When one relationship is transmitted to other sites, not all the data participate in the join-operation or could be used. So, the data which don't participate in the join-operation or useless will not be in the network transmission. The basic principle of query optimization strategy based on semi-join operation just reduces

the data quantity in relationship operation and the data transmission among sites.

A. The definition of semi-join operation

Semi-join is the algebraic relationship operation derived from the connection and projection operation. Its definition is as follows [2]:

If there were two relationships named R and S on the site 1 and site 2 respectively, and property A and B are the two properties respectively on them. So the semi-join in public property R.A = S.B can be expressed as:

$$R \bowtie_{A=B} S = \Pi_R(R \bowtie_{A=B} S) \quad (2)$$

Using the equivalent transformation relationship of query, the formula (1) can be expressed as:

$$R \bowtie_{A=B} S = R \bowtie_{A=B} (\Pi_B(S)) \quad (3)$$

Attention: The semi-join operation is not symmetry, namely $R \bowtie_{A=B} S \neq S \bowtie_{A=B} R$.

The full connection operation can be expressed by the semi-join operation as the intermediate steps. With the semi-join operation, the full connection operation in public property R.A = S.B can be expressed as:

$$R \bowtie_{A=B} S = (R \bowtie_{A=B} S) \bowtie_{A=B} S = (R \bowtie_{A=B} (\Pi_B(S))) \bowtie_{A=B} S \quad (4)$$

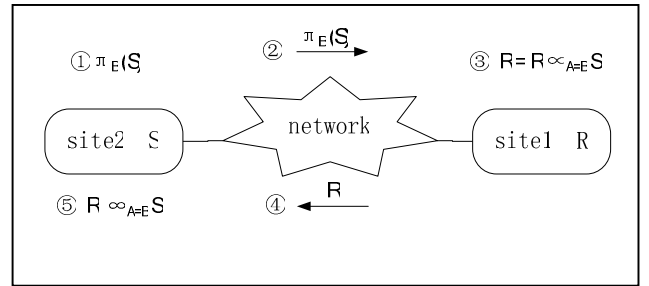


Figure 1. The connection process expressed by semi-join

B. The connection process and cost estimates using the semi-join optimization method

Specific operation process can be shown in Figure 1 and its cost estimates are as follows [3]:

1) With the public property B, $\Pi_B(S)$ can be projected between the relationship R and S on site 2.

2) Transmit the result of $\Pi_B(S)$ from site 2 to site 1, the cost is:

$$C_0 + C_1 * \text{size}(B) * \text{val}(B[S]) \quad (5)$$

Well, $\text{size}(B)$ stands for the length of property B, and $\text{val}(B[S])$ stands for the quantity of different values within property B in the relationship B.

3) Calculate the semi-join on site 1 and then set the results for R', then $R' = R \bowtie_{A=B} S$.

4) Transmit R' to site 2 from site 1, the cost is:

$$C_0 + C_1 * \text{size}(R) * \text{card}(R') \quad (6)$$

Well, $\text{card}(R')$ stands for the quantity of records in R' .

5) Do the connection operation on the site 2, then $R' \bowtie_{A=B} S$.

The total cost using the semi-join method for the query optimization is:

$$C_{sj} = 2C_0 + C_1(\text{size}(B) * \text{val}(B[S]) + \text{size}(R) * \text{card}(R')) \quad (7)$$

If we transmit the relationship R from site 1, then executive the connection operation with S on the site 2, the total cost is:

$$C_{nj} = C_0 + C_1 * \text{size}(R) * \text{card}(R) \quad (8)$$

In the process of query optimization, there may be several methods using the semi-join algorithm to optimize the connection query. Then we should select the best method after calculating the cost of each available semi-join scheme. Next, we can select the lowest site cost and calculate the full connection's cost. Finally determine the optimal method after comparing the two schemes.

C. The application of semi-join operation

Now let's take the Teaching database as an example and optimize the queries in the distributed database system based on semi-join operation [4].

The relationship distribution of site A and site B is shown in TABLE I. The E-R figure of teaching database is shown in Figure 2.

TABLE I. THE RELATIONSHIP DISTRIBUTION OF SITE A AND SITE B

Site	Relationship
A	Student (<u>Sno</u> , Sname, Ssex, Sage, Sdept)
	SC (<u>Sno</u> , Cno, Grade)
B	Course (<u>Cno</u> , Cname, Ccredit)

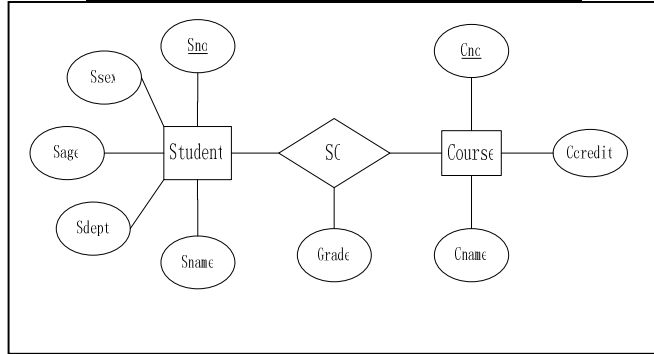


Figure 2. The E-R figure of teaching database

For table Student (Sno, Sname, Ssex, Sage, Sdept), there are 104 records. Sno stands for student's number, Ssex stands for student's sex, Sage stands for student's age, Sdept stands for student's major.

For table Course (Cno, Cname, Ccredit), there are 104 records. Cno stands for course's number, Cname stands for course's name, Ccredit stands for course's credit.

For table SC (Sno, Cno, Grade), there are 105 records. It's the connection of Table Student and Table Course. Sno stands for student's number, Cno stands for course's number, Grade stands for the student Sno's grade of course Cno.

We suppose that the length of each property in the each record Table 1 is 20 bit, the transmission speed of communication system is about 104 bit/s and the delay time of communication is 1s. According to the above table relationship, select students' number and name whose elective course's credit is 2 and the course's grade is more than 85. In the support of sliced transparent DDBMS, the SQL statements can be expressed as:

```
SELECT Sno, Sname
FROM Student, Course, SC
WHERE Student.Sno=SC.Sno AND
Course.Cno=SC.Cno AND
Ccredit='2' AND Grade>85
```

Suppose: There are 1000 course records whose credit's value is 2, 2000 records whose grade's value is more than 85, 500 records whose credit's value is 2 and grade's value is more than 85. The actual communication cost is different with diverse strategies of semi-join operation. The specific steps and cost estimate are given below [5]:

Strategy one:

1) Project the records whose credit's value is 2 in the Table Course on the site B, then sent them to site A.

2) Semi-join the results on the site B and the Table Student and SC, the communication cost is:

$$C_{sj} = 2 * 1 + (20 * 1000 + 500 * 20 * 7) / 10^4 = 11(s)$$

Strategy two:

1) Sent the courses' numbers from the Table Student and SC from site A to site B.

2) Semi-join the results on the site A and the Table Course, the communication cost is:

$$C_{sj} = 2 * 1 + (20 * 2000 + 20 * 500 * 3) / 10^4 = 9(s)$$

If the query operation using the full connection method directly, sent the Table Course from site B to site A, then connect them. The communication cost with this method is: $C_{nj} = 1 + 20 * 10^4 / 10^4 = 21s$. It can be seen easily that it's the most applicable to use the semi-join operation to query only when $C_{sj} < C_{nj}$ according to the comparison of the two ways above.

D. SDD-1 algorithm

In the SDD-1 algorithm, it uses the semi-join algorithm to handle the connection among the relationship and cut them. When all of the relationship is the maximum contract, then transfer it to a site where the query can be executed. The query operation is not always ended on this site [6].

SDD-1 algorithm has three important characteristics as follows:

- 1) It uses the semi-join operation to handle strategy.
- 2) The relationship of the whole sites is not repetitive and fragmented.
- 3) During price estimation of the whole algorithm, the transmission cost to the starting site is not calculated.

SDD-1 algorithm consists of two parts: the basic algorithm and the post-optimality. The basic algorithm evaluates the factors of execution strategy such as cost and profit according the reduced price formula. Then the program set cut by semi-join operation will be given. Finally, the most beneficial execution strategy will be decided. But the efficiency of this strategy may not be optimal. Post-optimality is the process of amending the implementation of basic algorithm to get more rational operation.

A query graph is given. The basic algorithm is described as follows:

- 1) Evaluate the income of all semi-join programs in the query graph.
- 2) Choose the semi-join with maximum income, execute it and recount the income of all the semi-join.
- 3) Execute step 2 circularly until all of the semi-join value of whose income is greater than 0 has been executed.
- 4) Select the site with minimum communication cost as the last executing site. Transmit all the relationship to the site and connect them to get the ultimate result.

The post-optimality can update the algorithm in two ways:

- 1) If the site where the last semi-join operation cut the relationship is the last executing location, the last semi-join operation can be canceled.
- 2) Correct the semi-join flow chart from the circulation of the basic algorithm. The cost of the original semi-join operation may be high, but the income evaluation is very large. If there is this kind of situation, the semi-join operation can be executed after the relationship is cut short. Then, the order of semi-join operation can be corrected.

Although the SDD-1 algorithm can cut short the relationship effectively and reduce the communication cost, there are some disadvantages, such as the complexity of the algorithm. When the number of records is very large, the cost

on query and search will increase rapidly. Moreover, the SDD-1 algorithm doesn't make full use of the distributivity of distributed database system. All of the semi-join operations are executed in order and it will increase the response time of query to a certain extent.

V. CONCLUSION

The goal of the distributed query optimization operations is to reduce the volume of data and the cost for process. The communication cost is one of the key factors among them. There may be many methods in same query. Finally, we compare the cost and results between two different semi-join operation methods and prove that the query methods can affect the execution speed of the system directly.

The query optimization is one of the most important research directions in whether the centralized database or distributed database. Because of the complex establish environment and rich technology content of distributed database, there are still aspects deserving further study. We believe that the query optimization technology in the distributed database will be more and more perfect.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central Universities (Grant No. 2012TD019) and Liaoning Provincial Department of Education Science Research Project (No.L2012173).

REFERENCES

- [1] Yan T, Iacobson M, Garcia-Mo Lina H, et al, Introduction of Query optimization of distributed database. Paris, FR: WAM Press, 1999.
- [2] Shan Wang, Shixian Sa. Theory & Application of Database System. Bei Jing: Higher Education Press, 2006 (In Chinese).
- [3] Xiuxia Yu, Yajuan Song. A Study of Optimized Algorithm for Distributed Database Half-join Query. Journal of Changchun University of Science and Technology, Vol 29. NO.4, pp.69-72, 2006 (In Chinese).
- [4] Zhengshu Jin, Hualiang Zi, Dongfang Lv. The Inquiry Optimization in The Distributed Data Base System. Journal Of Northeast China Institute Of Electric Power Engineering, Vol 23. No.4, pp. 12-15, 2003 (In Chinese).
- [5] Bing Liu, Fuliang Guo. Query optimization Tactics in Distributed Database System. Computer & Digital Engineering, Vol 33. NO.12, pp.81-83, 2005 (In Chinese).
- [6] Yang Zhang. Research on Data Query Optimization in Distributed Database. China University of Petroleum (East China), 2010 (In Chinese).