DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5731

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SURVEY ON IMPACT OF SEMI JOIN IN DISTRIBUTED QUERY PROCESSING

E. Padmalatha
CBIT
Hyderabad, Telangana, India

S.Sailekya
BVRITH
Hyderabad, Telangana, India

Abstract: Distributed system query processing is an essential factor in the presentation of a distributed data bases. Query processing in rapports of communication cost and processing cost should be minimum in the distributed data bases. As relations are partitioned based on horizontal or vertical partitions, a query is divided in to sub queries on the partitions that require operations at geographically separated databases. Query optimization is a difficult task in a distributed databases environment as data location becomes a foremost consideration. The query optimizer should select an efficient resultant for the given query, If a poor query execution plan is selected it will lead to a poor performance of the database system. Always the cost of execution of the query subjective function of the system resources needed to execute the query. System resources are like CPU time and the number of read, write operations on a relation. Realistic cost estimates of the optimizer need to evaluate the size of sub-queries. This will play a vital role in the selection of the join order of the relations. To approximation the dimensions of sub-queries, the optimizer needs to know the fussiness of the query basesThis paper briefly described join and semi join operation performance in the distributed data bases and analyzed with the practical application.

Keywords: semijoin ,query execution .query optimization.

1. INTRODUCTION

Data base management systems and the applications which involve with the large amount of data performance depend on the distributed and parallel processing. The user of the distributed data bases should feel as if they were in a single database [1]. In distributed computing systems they share input, output resources where as in DDBsystem the data and the operations on the data items are play the alike significant [2] .As per the name distributed databases in this system data is stored across geographical locations. The query processing in DDBS involves mostly three stages: local processing stage, reduction phase, and final processing phase [3]. The local processing phase consists of operations of selections and projections on the relations; the reduction stage apply a chain of reducers like semi joins and joins to reduce the size of relations; and the final processing phase sends all consequential relations to the assembly site where the final result of the query is constructed. This immature method like sending all relations directly to the third stage , to joins all relations, is hostile due to with huge transmission overhead and because less impact of parallelism .In distributed query processing, partition a relation into number of partitions, union of the partitions to form a entire relation, and transfer a relation or partition from one to another database are frequent operation .

2. METHODOLOGY OF PROCESSING DISTRIBUTED DATABASES

Distributed query processing methodology is described by P. Valduriez [1]and the pictorial representation is shown in Figure1.

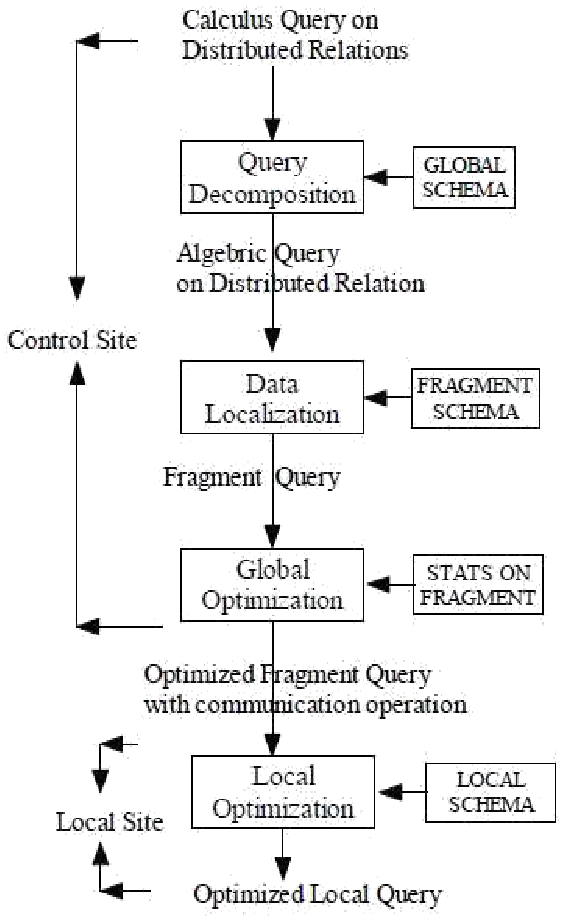


Figure1.Distributed Query Processing Methodology

The in the above Figure1 input query is internally uttered in relational calculus. There are different layers considered in processing distributed query as shown in Figur1. Main layers are involved in the distributed query optimizations. Each layer performs the operations of query decomposition, data localization, global query optimization, and local query optimization while mapping the distributed query on a local data base. At a central site the first three layers are considered. At local sites optimization is performed by the last layer shown in Figure 1.The cost of communication costs can be predictable around as [5] Transmission Cost (TC (x) =C0 +X *(C1)) X is for data transmission amount, unit of TC is b (bit) for computing. C0 is the original communication between two sites ,a constant, units are s (second). C1 is for the unit cost, and unit is s/b.

3. THE APPLICATION OF SEMI JOIN OPERATION

3.1 Communication cost in query processing

Semi join: It is used to diminish the communication cost in performing a join operation. Main idea is to trim down the size of a relation that needs to be transmitted and hence the communication cost.

The factors involved in Distributed query processing[7] are i)communication ii)performance cost. The two main approaches semi join and join sequence have been used to condense the amount of data transmission required for the phases of distributed query processing. Simple joining processing is one of most expensive operation.

In Site S1 student relation is considered with schema student (sno,sname,sex,sage,sdept). In Site S2 Course relation consist the schema of Course(cno,cname,credit). Studentcourse relationship is considered between Student relation and the Course relation .The of the relationship is with the schema of Studentcourse(sno,,cno,grade).

If the length of every attribute in the each relation is 30 bit, the speed of communication system is about 104 bit/s and the delay in time 1s. In above mentioned relations if the select sno ,sname has to be selected where elective credit is 3and the grade of the course is more than 75.Now the query statements can be expressed as:

```
SELECT Sno, SnameFROM Student, Course,
StudentCourse
WHERE Student.Sno=StudentCourse.Sno AND

Course.Cno=StudentCourse.Cno AND Credit =’3’
AND Grade>75
```

Here the assumption is for credit value 3 there are 2000 records are selected , for grade value more than 75 than 3000 records .For both credit value 3 and grade value more than 75 if 1000 records are considered. Now the real communication cost depends up on strategies of the semi-join operation. The specific steps and cost estimate are given below [6] whereas Csj represents cost of semi join and Cnj represents natural join.

STRATEGY 1:

1) Displaying the records whose value of credit is 3 relation Course in site S2, then sent them to site S1.

2)combine the results on the site S2 and the relation Student and StudentCourse, the communication cost is:

Csj $=2*1+(30*2000+1000*30*7)/10^4$ =29s------1

Strategy 2:

1)By sending cno from the relation Student and Student Course from site S1 to site S2.

2)Semi-join the results on the site S1 and the relation Course, the communication cost is:

Csj$=2*1+(30*3000+30*1000*3)/10^4$ =14s-------2

Directly if the relation Course sent from site S 2 to site S1, then make a connection on them. The cost of the communication obtained with this direct is like 31s

Cnj =1+30*10 /10 = 31s --------- 3.

3.2 Performance cost in query processing

The relations Book and Publisher are stored at different sites. Book relation is having 500 tuples and each tuple is 100 byte long, total size of Book is = 500 X 100, 50,000 bytes. Book relation is considered in site S1. 100 tuples are in relation Publisher and each tuple is 150 bytes long ,total size of Publisher is = 100X150 = 15,000 bytes. Publisher relation is considered in site S2 .The output of join is considered in site S3.

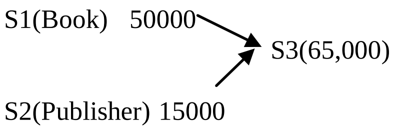
Projecting the ISBN from relation Book,Price,Pname,P_id from publisher_.

$$\boldsymbol{\pi}_{\text{Price}}\text{, P\_id,Pname (Book} \underset{\text{Book.P\_id=Publisher.P\_id}}{\bowtie} \text{Publisher)}$$

If join operation is performed on the both relation Book and Publisher ,by assuming ISBN size is 15,Price is 6,P_id is 4 and Pname is 50 bytes.After join, the resultant contain 500 tuples and each tuple will have is 75 bytes long because of the selected attributes in query (15+6+4+50).

Strategy1:

Transfer replicas of both relations to site S3 and process the query. Site S1(Book)with 50,000 bytes , Site S2(Publisher) with 15,000bytes. Moving S1 and S2 to S3to perform join operation. After performing simple join operation site S3will have 65,000 bytes.



Strategy2:

Transfer replica of publisher relation to site S1for processing at site S1 and then move result to S3.Move S2 (Publisher) to S1(Book) from SiteS2 15000 bytes move to S1. If the output considered at Site S1 that will be 37,5000 bytes(500*75). So totally 52,500 bytes moves to S3.

$$\begin{tikzcd}
\text{S1(Book) (o/p)} \arrow[r, "37{,}500"] & \text{S3(52,500 bytes)} \\
\text{S2 (Publisher)} \arrow[u, "15000"] &
\end{tikzcd}$$

Strategy3:

Transfer replica of Book relation to site S2 for processing at Site S2 and then ship result to Site S3.Now Site S1(Book) sends 50,000 to Site S2(Publisher) after processing at Site S2 37,5000 bytes move to Site S3.Now at Site S3 as resultant of join it will contain 87,5000 bytes.

$$\begin{tikzcd}
\text{S1(Book)} \arrow[d, "50{,}000"'] & \\
\text{S2(Publisher) (o/p)} \arrow[r, "37{,}500"] & \text{S3(87,500bytes)}
\end{tikzcd}$$

After considering the three strategies the strategy 2 is giving the less cost.

Semi join:

Book relation is considered in site S1, Publisher relation is considered in site S2 .Output is going to obtain at Site S2.Single strategy consists of three steps .

Strategy:

1. Project the join attribute (P_id) of Publisher at Site S2 and move it to S1.

   $$S2 \longrightarrow \pi_{P\text{-}id} \xrightarrow{100*4} S1 \quad [400 \text{ bytes}]$$

2. Join this transferred attribute P_ID with Book relation and transfer required attributes from result to S2 .

   $$S1 \xrightarrow{500*25} S2 \quad [12{,}500 \text{ bytes}]$$

3. Evaluate the expression by joining transferred attributes with Publisher at S2.

   S2[400 + 12,500= 12,900 bytes]

4. COMPARISONS OF JOIN AND SEMI JOIN

The values obtained in the section 3 for communication and performance cost indicates that in distributed query processing semi join will give us the less communication and performance cost as shown in Figure2.

Table1.Comparison of join operations

Cost / operation	Simple join	Semi join
Communication	31s	14s
Cost		
Performance	65,000bytes	12,900bytes
Cost		

From Table 1 it can be conclude that the most pertinent to use the semi-join operation to query only when Csj <Cnj. According to the comparison of the two costs Simple join processing is one of the most expensive operation in distributed query processing.

5. DEDUCTION

The main objective of the distributed query processing is to diminish the size of data and the cost of processing. The communication cost is considered as vital factors amongst them. Number methods can use on the same query to get the execution plan. Among all these execution plans, after relating the cost either in terms of communication or performance cost the best plan will be considered. If the best plan is not executed results between two different join operation methods can disturb the execution speed of the system candidly.

REFERENCES

[1] C. Liu and C. Yu, "Performance Issues in Distributed Query Processing," *IEEE,* vol. 4:8, pp. 889-905, 1993.

[2] S. Upadhyaya and S. Lata, "Task Allocation in Distributed Computing VS Distributed Database Systems: A Comparative Study," *IJCNS (International Journal of Computer Science and Network Security),*vol. 8:3, pp. 338-346, 2008.

[3] C. Wang and M.-S. Chen, "On the Complexity of Distributed Query Optimization," *IEEE* vol. 8, pp. 650-662, 1994.

[4] P. Valduriez and T. Ozsu, "Principle of Distributed Database Systems.," Prentice Hall, 1999.

[5] Lin Zhou,Yan Chen,Taoying Li " The Semi-join Query Optimization in Distributed Database System "CITCS2012.

[6] Bing Liu, Fuliang Guo. Query optimization Tactics in Distributed Database System. Computer & Digital Engineering, Vol 33. NO.12,pp.81-83,2005(In Chinese).

[7] B.M. Monjurul Alom, Frans Henskens and Michael Hannaford." Query Processing and Optimization in Distributed Database Systems" IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.9, September 2009 page no143.