Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improvement of the Performance of Multi Query Optimization Techniques in Distributed Database Management System

Ankita Ahuja, Ashok Kumar
CSE Department, DIT University, Dehradun,
Uttarakhand, India

Abstract: ***Query Optimization has been of great importance in distributed databases over last few years even due to the reason that distributed databases are vastly used now days. There has been a lot of work done on Query Optimization for distributed databases considering Semi join. In this paper we have worked on a new distributed Query Optimization algorithm in which we have taken into consideration joining i.e. Full join along with semi join. Later the Experimental analysis of the algorithm has been done that significantly shows that the algorithm effectively reduces the network communication cost with joining over the distributed databases.***

Keywords: Distributed Databases, Optimization Algorithm, Full Join, Left Join, Right Join.

I. INTRODUCTION

Since there has been a lot of advancement in technology over last few years, so centralized databases now has been replaced by distributed databases. Query Optimization then become of great importance since a variety of complex queries are executed in distributed databases.

II. DISTRIBUTED DATABASE OVERVIEW

A Distributed Databases is the collection of databases which are distributed over a computer network, so as to provide the easy access to the users. Computer network can include a large area or it can be a small area. The main objective of distributed database is that it should provide Location Transparency to the user i.e. user does not require to know the actual location of the data. The user at different locations may have right to access the databases distributed over the computer network.

Distributed Databases have been classified into following 2 types:-

1. Homogeneous Distributed Databases.
2. Heterogeneous Distributed Databases.

Query Optimization is an important step in distributed databases. The process of executing a given query effectively such that it reduces the minimum network communication cost required to process a query is known as query optimization.

Client/Server Distributed database architecture has been shown below, which includes a Server that contains the main database and the n number of clients that contains distributed databases which are connected with the Server via LAN/WAN.

All the processing in the Distributed Databases takes place through the Client/Server Architecture which is shown below.

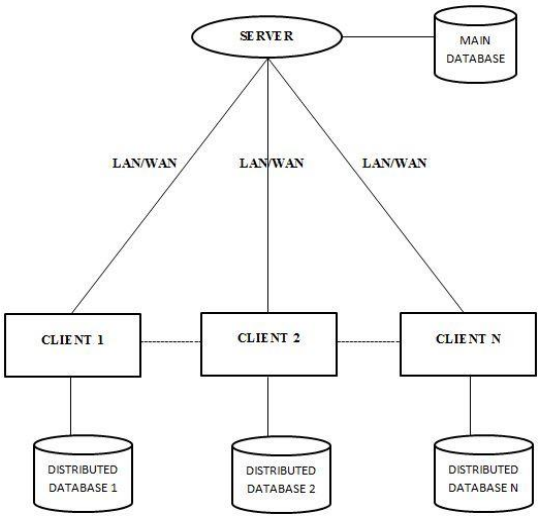


Fig 1 Client/Server Distributed Database Architecture.

III. RELATED WORK

Fan Yuanyuan, Mi Xifeng (2010), have designed the new semi connected database algorithm that is highly effective then the other algorithms which efficiently reduce network cost.

Preeti Tiwari, Swati V Chande (2013), have reviewed certain optimization strategies and Ant Colony Optimization algorithm is integrated with other optimization algorithms.

Song Lina (2013), have worked on the Query Optimization of Distributed Databases.

Seema, Parminder Kaur (2013), have designed a new query optimization algorithm based on Relational Algebra Equivalence Transformation.

Yasmeen R.M.Umar, Amit R.Walekar (2014), have reviewed the query optimization challenges in Distributed Databases and certain proposed system have also been reviewed.

IV. PROPOSED WORK

A Distributed Query Optimization Algorithm has been implemented using the concept of joining i.e. Full join along with Semi join. The algorithm is based on the concept that the Full join will only be implemented on the attributes that have constraints on them i.e. Primary key or Foreign key, whereas semi join will be implemented on all the attributes. The basic idea of the algorithm is to show that the network communication cost is reduced with full join as compared to semi join. There can be n number of clients in the network depending upon the availability of clients and there can be N number of multi join queries that can be executed on the client side.

The Network Communication Cost of Full join and Semi join is calculated by using the following formula:-

Total Cost=T_{cpu} * insts + $T_{I/O}$ * ops + C_0 + C_1+* X

Where,

T_{CPU} = CPU processing cost per instruction.

insts = Represents the total no of CPU instructions.

$T_{I/O}$ = Input/output processing cost per I/O operations.

ops =Represents the total no of Input/output Operation.

C_0 = Startup Cost for initiating transmission.

C_1 = Proportionality Constant.

X =Amount of data to be transmitted.

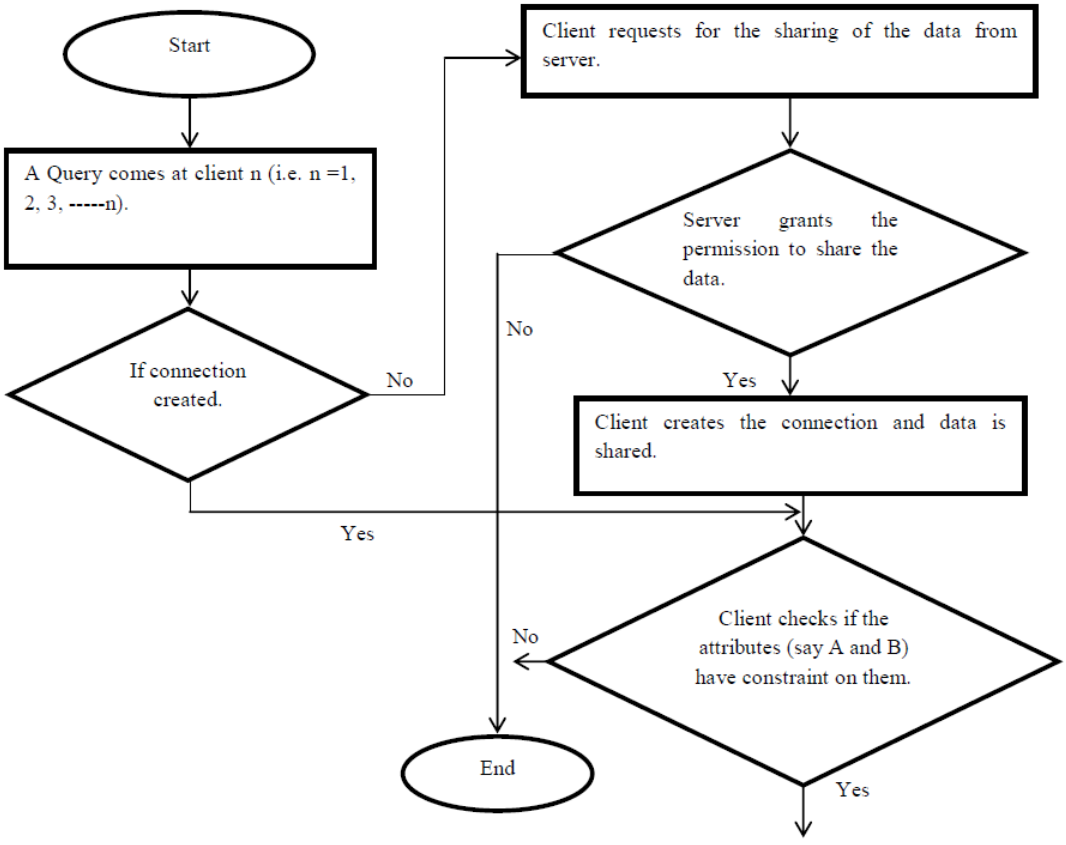
Parameters used in Algorithm are as follows:-

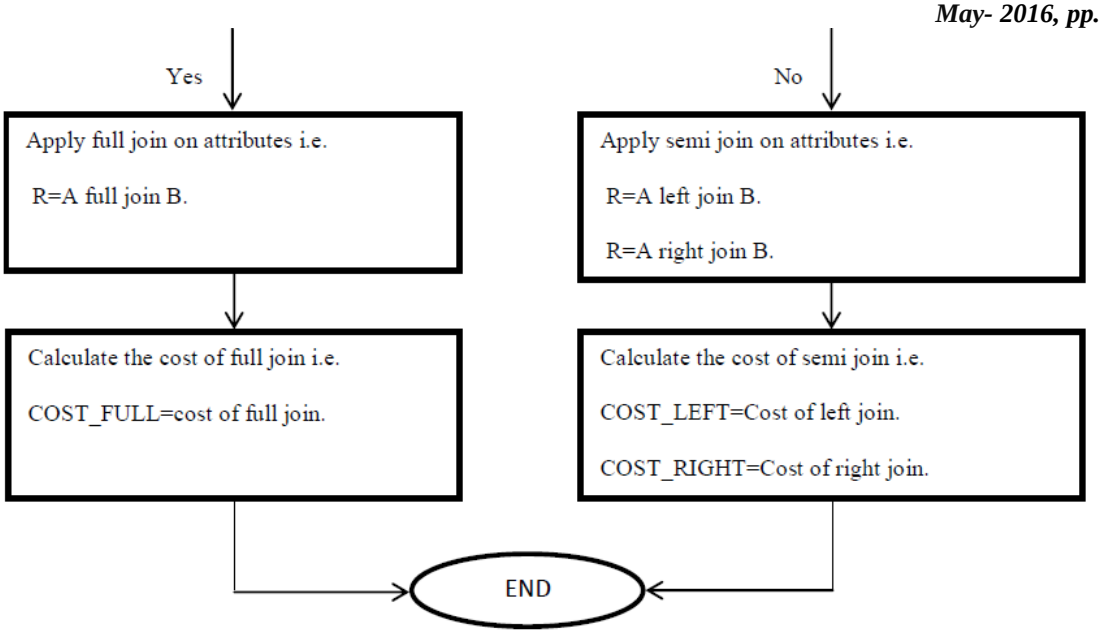
COST_FULL = Cost of Full join.

COST_LEFT =Cost of Left join.

COST_RIGHT =Cost of Right join.

The steps of Distributed Query Optimization algorithm are as follows:-





V. EXPERIMENTAL ANALYSIS

The proposed Distributed Query Optimization Algorithm is implemented on Oracle 11g Server and Clients. Initially, the College University Database is created on the Server and then shared with the clients via LAN by creating connections. Then the cost of Full Join and cost of Semi Join is calculated and later the comparison is done to conclude which of the join gives the minimum communication cost.

The database tables created on the Server consists of following tables:-

1. Students (Master) Table
   Students (StudentId, Program, FirstName, LastName, FathersName, MobileNo, EmailId, Batch);
2. Course Table
   Course (CourseId, DepartmentId, CourseName);
3. Departments Table.
   Departments (DepartmentId, DepartmentName);
4. Enrollments Table.
   Enrollments (StudentId, CourseId, StudentName, Semester);
5. Results Table.
   Results (StudentId, CourseId, StudentName, Semester, SGPA);

After creating the tables on Server, the algorithm is executed on the clients.

Client 1

Step 1: - Initially some query comes at client 1 for e.g.:-

Select * from Course, Departments where CourseName='BTech in Information Technology' and DepartmentName='Information Technology';

Step 2:- Since the connection was not created, so client1 creates the connection and the data of server is being shared.

Step 3:- Now client1 checks if the attributes i.e. CourseName and DepartmentName in the requested query have constraints on them. Since, the attributes have no constraint on them so semi join will be applied on attributes i.e. Left join query is as follows:-

Select Course. CourseId, Departments. DepartmentName from Course Left Join Departments on Course. CourseName=Departments. DepartmentName order by Course. CourseId;

The above query is executed in 0.00853096 seconds.

Whereas, if the above query is executed with right join then,

Select Course. CourseId, Departments. DepartmentName from Course Right Join Departments on Course. CourseName=Departments. DepartmentName order by Course. CourseId;

The above query is executed in 0.00784466 seconds.

Step 5:- Now calculating cost for left join and right join.

COST_LEFT $= T_{CPU} * insts + T_{I/O} * ops + C_0 + C_1 * X$
$= 0.000192357 * 20 + 0.000406236 * 21 + 0.02 + 1 * 0.001673$
$= 0.034075$ Mb/Sec.

COST_RIGHT $= T_{CPU} * insts + T_{I/O} * ops + C_0 + C_1 * X$
$= 0.000192357 * 20 + 0.000373555 + 0.02 + 1 * 0.001673$
$= 0.033364$ Mb/Sec.

Similarly, a set of 19 other Multi join queries were executed on client 1 and we got the following results:-

	CLIENT 1 LEFT JOIN	CLIENT 1 RIGHT JOIN	CLIENT 1 FULL JOIN
QUERY 1	0.034075		
QUERY 2		0.033364	
QUERY 3	0.042416		
QUERY 4		0.055474	
QUERY 5	0.039942		
QUERY 6		0.061015	
QUERY 7	0.093353		
QUERY 8		0.041185	
QUERY 9	0.053485		
QUERY 10		0.039154	
QUERY 11	0.047503		
QUERY 12		0.034737	
QUERY 13			0.05546
QUERY 14			0.045663
QUERY 15			0.067538
QUERY 16			0.037207
QUERY 17			0.033827
QUERY 18			0.035302
QUERY 19			0.029465
QUERY 20			0.039501

Fig 2 Execution results for Client 1.

Client 2

The Algorithm is now implemented on Client 2 with the set of other 20 Multi join queries, the result has been calculated in the same way as done for Client 1.The result after the execution is as follows:-

	CLIENT 2 LEFT JOIN	CLIENT 2 RIGHT JOIN	CLIENT 2 FULL JOIN
QUERY 1	0.075563		
QUERY 2		0.048267	
QUERY 3	0.047507		
QUERY 4		0.044849	
QUERY 5	0.044767		
QUERY 6		0.047473	
QUERY 7	0.048018		
QUERY 8		0.046777	
QUERY 9	0.039643		
QUERY 10		0.042595	
QUERY 11	0.046529		
QUERY 12		0.048744	
QUERY 13			0.039014
QUERY 14			0.046024
QUERY 15			0.049585
QUERY 16			0.044367
QUERY 17			0.037783
QUERY 18			0.039207
QUERY 19			0.049325
QUERY 20			0.044493

Fig 3 Execution results for Client 2.

Client 3

The Algorithm is now implemented on Client 3 with the set of other 20 Multi join queries; the result has been calculated in the same way as done for Client 1. The result after the execution is as follows:-

	CLIENT 3 LEFT JOIN	CLIENT 3 RIGHT JOIN	CLIENT 3 FULL JOIN
QUERY 1	0.088331		
QUERY 2		0.039263	
QUERY 3	0.038172		
QUERY 4		0.033644	
QUERY 5	0.055224		
QUERY 6		0.034246	
QUERY 7	0.040026		
QUERY 8		0.036221	
QUERY 9	0.03637		
QUERY 10		0.06638	
QUERY 11	0.034581		
QUERY 12		0.035849	
QUERY 13			0.034303
QUERY 14			0.039872
QUERY 15			0.044064
QUERY 16			0.033981
QUERY 17			0.044333
QUERY 18			0.034301
QUERY 19			0.038344
QUERY 20			0.0299

Fig 4 Execution results for Client 3.

Comparison of results of client 1, client 2, client 3.

A comparison have been done after executing the algorithm on 3 different clients in a distributed environment and certain results has been drawn by showing the certain graph representations that effectively proves that the full join gives the better result over semi join.

In the following table Full Join transaction has been compared with the individual transactions of Semi Join (i.e. Left Join and Right Join).

A	B	C	D	E	F	G	H	I	J
	CLIENT 1	CLIENT 1	CLIENT 1	CLIENT 2	CLIENT2	CLIENT 2	CLIENT 3	CLIENT 3	CLIENT 3
	LEFT JOIN	RIGHT JOIN	FULL JOIN	LEFT JOIN	RIGHT JOIN	FULL JOIN	LEFT JOIN	RIGHT JOIN	FULL JOIN
QUERY 1	0.034075			0.075563			0.088331		
QUERY 2		0.033364			0.048267			0.039263	
QUERY 3	0.042416			0.047507			0.038172		
QUERY 4		0.055474			0.044849			0.033644	
QUERY 5	0.039942			0.044767			0.055224		
QUERY 6		0.061015			0.047473			0.034246	
QUERY 7	0.093353			0.048018			0.040026		
QUERY 8		0.041185			0.046777			0.036221	
QUERY 9	0.053485			0.039643			0.03637		
QUERY 10		0.039154			0.042595			0.06638	
QUERY 11	0.047503			0.046529			0.034581		
QUERY 12		0.034737			0.048744			0.035849	
QUERY 13			0.05546			0.039014			0.034303
QUERY 14			0.045663			0.046024			0.039872
QUERY 15			0.067538			0.049585			0.044064
QUERY 16			0.037207			0.044367			0.033981
QUERY 17			0.033827			0.037783			0.044333
QUERY 18			0.035302			0.039207			0.034301
QUERY 19			0.029465			0.049325			0.038344
QUERY 20			0.039501			0.044493			0.0299

Fig 5 Comparison of Full Join with Individual Transaction of Semi Join.

Graph representation of above figure.

Client 1

Fig 6 Graph for Client 1.

Client 2

Fig 7 Graph for Client 2.

Client 3

Fig 8: - Graph for Client 3.

The above 3 graphs for Client 1, Client 2, Client 3 have been compared with each other. The conclusion made is that, in Client 1 the cost of full join gives the slight variations when compared with the cost of semi join, but the lowest value comes for full join only. In Client 2, queries of full join shows the better result when compared with semi join .In client 3, also the full join queries gives the better result when compared with Semi join. So, Full Join is considered as the better operation over the Semi join, as it gives the minimum Communication cost over Semi Join in distributed environment. We also concluded that in client 2 and Client 3,full join gives the better result over the client 1.So the network communication cost also depends on the speed of network and the execution time taken to process a query.

Comparisons of full join transaction with the overall transaction of semi join.

	CLIENT 1 SEMI JOIN	CLIENT 1 FULL JOIN	CLIENT 2 SEMI JOIN	CLIENT 2 FULL JOIN	CLIENT 3 SEMI JOIN	CLIENT 3 FULL JOIN
QUERY 1	0.067439		0.12383		0.127594	
QUERY 2	0.09789		0.092356		0.071816	
QUERY 3	0.100957		0.09224		0.08947	
QUERY 4	0.134538		0.094795		0.076247	
QUERY 5	0.092639		0.082238		0.10275	
QUERY 6	0.08224		0.095273		0.07043	
QUERY 7		0.05546		0.039014		0.034303
QUERY 8		0.045663		0.046024		0.039872
QUERY 9		0.067538		0.049585		0.044064
QUERY 10		0.037207		0.044367		0.033981
QUERY 11		0.033827		0.037783		0.044333
QUERY 12		0.035302		0.039207		0.034301
QUERY 13		0.029465		0.049325		0.038344
QUERY 14		0.039501		0.044493		0.0299

Fig 9 Comparison of Total Executions.

In the above figure, the transaction of Full join has been compared with the overall transaction of semi join. The graph representation of the above figure is as follows:-

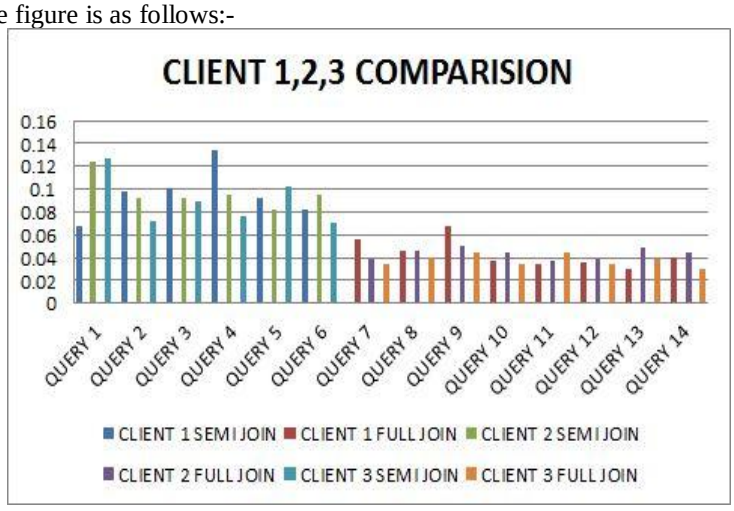


Fig 10 Graph Representation of Total Comparison

The above graph shows that the Full Join is a better operation than Semi join, because it successfully reduces the Network communication cost in a distributed database environment.

VI. CONCLUSION

In this paper, new Query optimizations Algorithm for Distributed Databases have been proposed using the concept of joining i.e. Full Join and Semi Join. The basic idea of joining is to use Full Join only on the attributes that have constraint on them i.e. Primary key or Foreign Key and Semi Join can be used on all the attributes. The basic objective of the algorithm is to show that the Full Join reduces the network communication cost in Distributed Databases. Later the Experimental Analysis of the algorithm was done in which the algorithm was executed in real time environment on 3 different clients. Different Multi join queries were executed and accordingly the result was calculated for 3 different clients, then the comparison between the results were done accordingly and by certain graph representation it was shown that Full join significantly reduces the network communication cost and also proves the efficiency of the new designed algorithm.

ACKNOWLEDGEMENT

I would like to thank my guide Dr. Ashok Kumar, who helped me in my research work throughout the year, by providing me the depth knowledge of the subject. I would also like to thanks my parents who always supported me in completing my research work.

REFERENCES

[1] Fan Yuanyuan and Mi Xifeng, *Distributed Database System Query Optimization Algorithm Research*,978-1-4244-5539-3/10/$26.00 ©2010 IEEE.

[2] Ms. Preeti Tiwari, Swati V.Chande.*Query Optimization Strategies in Distributed Databases.* International Journal of Advances in E ngineering Sciences, Vol.3 (3), July, 2013 e-ISSN: 2231-0347 Print-ISSN: 2231-2013.

[3] Song Lina, *Research on Query Optimization Algorithm in Distributed Database. International Journal of Digital Content and its Applications (JDCTA)*, Volume 7, Number 6, March 2013,doi:10.4156/jdeta.vol 7 issue 6.8.

[4] Seema, Parminder Kaur. *Query Optimization Algorithm Based On Relational Algebra Equivalence Transformation*.I.J.E.M.S. VOL.4 (3) 2013:326-311.

[5] Yasmeen R.M. Umar and Amit R. Welekar, *Query Optimization in Distributed Database: A Review.* International Journal of Current Engineering and Technology, Vol. 4, No.6 (Dec. 2014).

[6] Jyoti Mor, Indu Kashyap, R.K.Rathy. *Analysis of Query Optimization Techniques in Database,*. International Journal of Computer Applications (0975-888).Volume 47-No.15, June 2012.

[7] Pankti Doshi, Vijay Raisinghani, *Review of Dynamic Query Optimization Strategies in Distributed Database*, 978-1-4244-8679-3/11/$26.00 © 2011 IEEE.

[8] Morteza Nasiraghdam, Shahriar Lotfi, Reza Rashidy. *Query Optimization in Distributed Database Using Hybrid Evolutionary Algorithm.* 978-1-4244-5651-2/10/$26.00 ©2010 IEEE.

