Stochastic Analysis of DSS Queries for a Distributed Database Design

Manik Sharma
Ph.D. Research Scholar,
Punjab Technical University, Jalandhar

Gurvinder Singh
Professor & Head, DCSE, GNDU
Amritsar

Rajinder Singh
Associate Professor, DCSE, GNDU
Amritsar

Gurdev Singh
Professor (IT), Gurukul Vidyapeeth,
Punjab Technical University, Jalandhar

ABSTRACT
Optimization of query in distributed database system is one of the dominant subjects in the field of database theory. Depending upon the placement of data a query can be described as centralized or distributed query. The processing of distributed query is entirely different from the centralized query as in the former case the data is distributed over number of sites. Decision Support System Query (DSSQ) is one of the decisive types of distributed query. DSS queries are complex and time consuming in nature. Due to the decentralization of data and the complexity of query, it becomes mandatory to optimize the DSS query in distributed database system. In this work an effort is made to find an optimal DSS sub query allocation plan in distributed environment stochastically using Genetic Algorithm. The queries are designed on the basis of one of the benchmark of DSS query as given by TPC-DS. The DSS queries are optimized on the basis of Total Cost. The use of Genetic Algorithm has significantly expedited the process of DSS query optimization. The effect of varying communication cost over Total Cost of system resources is also observed.

Keywords
DSS Query; Distributed Database; Genetic Algorithm, Sub-query Allocation Plan.

1. INTRODUCTION
All the database operations are performed by using the concept of a query. Query is a statement that performs the specific function on the database. The successful implementation of a query either selects the data from database or will change the state of data in database. At the basic level a query can be categorized as data retrieval query or an action query (Write query, update query, deletion query etc.)[5] . Each query is performed by using different set of statements. In general all the data retrieval queries are performed by using select statement followed by where, group by or having clause. Similarly the Write and update queries are performed by using insert and update statement respectively. Depending upon the location of data a query can be defined as centralized or distributed query. Distributed queries are more complex to handle and optimize as compare to centralized database queries. Distributed Database is one of the major progresses in the field of database theory. Distributed database system [1] [2] [4] [7] [8] [19] is the compilation of logically interrelated data disseminated over several sites. Technically DDB system [3][4][13] is the convergence of network and database technologies. In distributed database the queries can be categorized as OLTP and DSS Queries.

DSS queries are long running, complex queries normally affect large amount of data as compare to OLTP queries. DSS queries are normally used to retrieve data from huge database. Due to complexity, DSS queries consume significant amount of system resources and can easily saturate CPU or memory of the query server.

One of the important characteristics of DSS queries is that their running time is normally unpredictable as compare to OLTP queries. Due to dissemination of data over several sites one of the major issues in the optimization of DSS distributed query is the placement of data and program across the different sites available in distributed database system. The performance of the DSS query is heavily dependent upon data allocation and operation allocation. A DSS query normally consist of 3 to 15 joins operations and deals with table of having size in MB, GB or even in TB. In DSS queries the data access operation is normally associated with read operation.

In this case DSS query is optimized by using cost based optimization model. Total Cost of System Resources is considered as an optimization parameter. In distributed database system cost is associated with each basic operation (I/O, Processing & Communication) to generate Input Output Cost, Processing Cost and Communication Cost. To determine the Total Cost of system resources of a query is computed by finding the sum of input output cost, processing cost and communication cost.

Genetic Algorithm is used to compute the Total Cost of DSS query based on the Input-Output Cost, Processing Cost and Communication Cost. All the basic operations of GA like selection, crossover and mutation are implemented to find an effective DSS query allocation plan (based on optimal Total Cost) in distributed database system. The use of genetic algorithm helps in finding an effective query allocation plan in a flash as compare to significant time taken in the traditional query optimization technique. An effort is made to find the effect of the varying communication cost w.r.t. I/O cost on the Total Cost of a query allocation plan.

2. FORMULATION OF PROBLEM
Query optimization [5][6]is one of the consequential concepts in regard to DSS Queries. The optimization process in DSS queries is much more complex as compare to OLTP Queries, because in the former case one has to optimize both total cost of system resources and response time to optimize a query.

In this paper prime focus is given on the optimization of DSS query by considering Total Cost of system resources in mind. An effort is also made to reduce the response time by

executing some of the sub operation of a DSS query in parallel. Beside this the effect of varying communication cost with respect to Input-Output cost on the Total Cost is also analyzed.

The Genetic Algorithm is used to expedite the process of query optimization to a great extent.

Structure of Distributed Database

For analyzing the DSS queries the TPC-DS benchmark [18] is referred and based on the its database the set of DSS queries is designed and analyzed. The structure of selected DSS queries with distributed database is given as below:

Store (Storeid, Sname, Manager, Market, Address, Company, City, State: Varchar; No_of_Emp:Number; S_date: Date;)

Customer (Custid, Cust_Fname, Cust_Lname, DOB, Contact, Email: Varchar;)

Cust_Address (Custid, HouseNo, Street, Street2, City, State, Country: Varchar)

Items (Itemcode, Name, Brand, Type, Size, Colour, Description, Ware_no: Varchar; Price: Number)

Sales(Saleid, Storeid, Store_name, City, Warehouse_id: Varchar; Item_code, Qty, Unit_price, Tax, Discount,Net_price:Number;)

Callcentre (CCID, Cen_name, Manager, Cen_Address: Varchar; No_ofEmp, Area_in_SQFT:Number;)

Webstore (Website, Web_id, Web_mkt_mgr, Nature: Varchar)

Warehouse (Wareh_id, Wname, Wmanager, Address, Company, City, State: Varchar; No_of_Emp, Ware_size: Number; S_date: date;)

Marketing (Mark_id, Mark_item, Mark_promo_name, Mark_Manager, Warehid: Varchar; Expenditure: Number; Mark_sdate, Mark_edate: Date)

Shipping (Ship_id, Ship_mode, Ship_item, ship_address, Ship_cont_person: Varchar; Ship_date: Date; Ship_item_units: Number)

Case 1: Find the first and last name of customer who purchased item having marketing promo 'supersaving' is going with for all warehouses situated in Punjab.

Relational Algebra Query

$$(\pi_{c1}(\sigma_{p1})B1) :X: (\pi_{c2}(\sigma_{p2})B2) :X: (\pi_{c3}(\sigma_{p3})B3) :X: (\pi_{c4}(\sigma_{p4})B4) \quad (1.1)$$

Case II: Find the address of customer who has purchased an item from the warehouse in Punjab. The marketing expenditure on the purchased item should be less than 4000. Item should be shipped by courier mode and must be available at webstore under the web store manager 'Khan'.

Relational Algebra Query

$$(\pi_{c1}(\sigma_{p1})B1) :X: (\pi_{c2}(\sigma_{p2})B2) :X: (\pi_{c3}(\sigma_{p3})B3) :X: (\pi_{c4}(\sigma_{p4})B4) :X: (\pi_{c5}(\sigma_{p5})B5) :X: (\pi_{c6}(\sigma_{p6})B6) :X: (\pi_{c7}(\sigma_{p7})B7). \quad (1.2)$$

Case III: Find the address of all customers

(i) Who have purchased an time from the warehouse located in their own city.

(ii) The marketing expenditure on the purchased item should be greater than average expenditure of marketing.

(iii) The purchased item should be shipped by courier mode.

(iv) The item purchased must be available at webstore under the web store manager 'Karan'.

(v) The type of item purchased should be electronics

(vi) The item purchased should be attended in the call centre called 'Galaxy'.

Relational Algebra Query

$$(\pi_{c1}(\sigma_{p1})B1) :X: (\pi_{c2}(\sigma_{p2})B2) :X: (\pi_{c3}(\sigma_{p3})B3) :X: (\pi_{c4}(\sigma_{p4})B4) :X: (\pi_{c5}(\sigma_{p5})B5) :X: (\pi_{c6}(\sigma_{p6})B6) :X: (\pi_{c7}(\sigma_{p7})B7) :X: (\pi_{c8}(\sigma_{p8})B1) :X: (\pi_{c9}(\sigma_{p9})B2) :X: (\pi_{c10}(\sigma_{p10})B3) X: (\pi_{c11}(\sigma_{p11})B4) \quad (1.3)$$

3. STATISTICS OF DSS QUERIES IN DISTRIBUTED DATABASE SYSTEM

The various attributes associated with set of selected DSS queries used in the nomenclature of distributed database system is depicted in the following figure.

The Figure 1 shows pictorial representation of the different attributes (# of Sites, # of Base Relations, # of Operations, # of Internal Fragments, # of Joins etc.) of selected DSS queries.

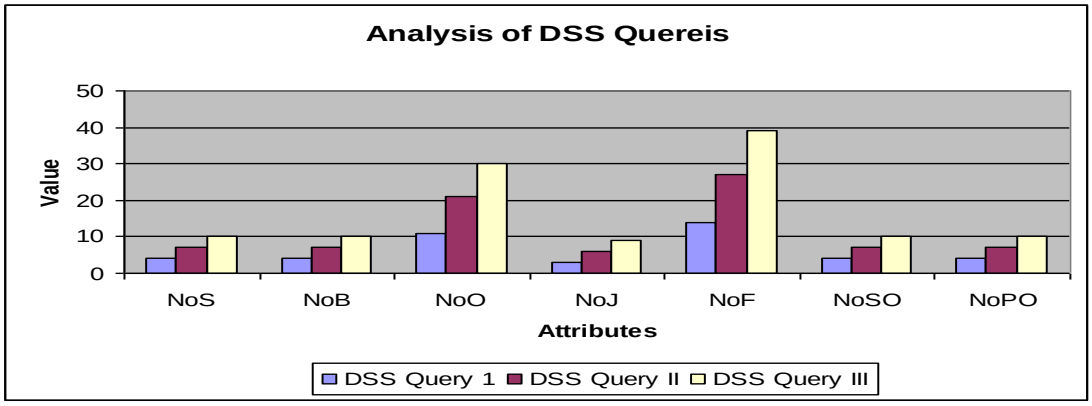

Figure 1: Analysis of DSS Queries in Distributed Database System

4. DSS QUERY OPTIMIZATION PROCESS

One of the major criticisms of distributed database system is lack of efficiency in handling data access queries. The concept of query optimization is used to solve the above said problem. Economic theory states that the optimization process should tries to maximize the benefit and minimize the loss.

In general the objective of query optimization [9] [12] [22] is to provide a cost effective query execution plan based on either Total Cost or Response Time. In database theory a query can be optimized by manipulating the order of sub operations, by minimizing the amount of data movement across different sites of a distributed system or by executing the query with the help of some algorithm.

In this case selected DSS queries are optimized by manipulating the order of sub operation using Genetic Algorithm. A DSS query represented by a Tree data structure is converted into a text file with various input factors (I/O Cost, Communication Cost, Processing Cost, Data Allocation Matrix etc) acts as an input to the simulator. Each node represents the sub operation like selection, projection or join of a query. Root node represents the final operation of a query [23].

The study has shown that [11] the existing manufacturer of database system provides optimization support for OLTP queries only, they have almost neglected optimization for DSS Queries. It is found that even a simple DSS query takes significant processing time and resources on commercially available database architectures.

The DSS Query allocation plan [10] is composed of two phases called OSP (Operation Sequence Problem) and OAP (Operation Allocation Problem). The role of OSP phase is to decompose a DSS query operation into number of sub operations (Selection, Projection, Joins, and Semi-joins) and generate the sequence in which sub operations are executed. OAP is responsible for allocating the decomposed DSS Query sub operations to different sites available in distributed database system.

A DSS query in distributed database system can be executed in number of ways [13][14] leading to number of alternate query allocation plan with different performance metrics like execution time, resource usage etc. Research shows that independent of query type (OLTP & DSS) the optimization was performed by using different deterministic and randomized techniques [16]. Research has shown that the one of widely used deterministic technique i.e. exhaustive enumeration gives best solution, but it takes days, month of even years in finding an optimal DSS query allocation plan hence it is not feasible to use exhaustive approach in case of optimization of a DSS query. Earlier dynamic programming was also used dominantly for optimization of a query.

5. ANALYSIS OF DSS QUERIES USING GENETIC ALGORITHM

The DSS queries designed on the basis of TPC-DS benchmark (Eq. 1.1, 1.2 & 1.3) are optimized and analysed by using one of the most widely used evolutionary approach that is Genetic Algorithm. Genetic algorithms are dominant research methodology used in versatile fields for providing an optimized result. In simple terms Genetic Algorithm [24] commonly abbreviated as GA is defined as a search algorithm that works on the principle of natural selection and natural genetics. One of the foremost objectives of Genetic Algorithm [26] is to provide best or nearly optimal solution from fixed and finite sized population. It is a probabilistic [20] technique based on the concept of natural selection. In simple words the concept of Genetic Algorithm is derived from nature and was first proposed by John Holland [22][24]. The use of GA [18] helps in finding the best optimized solution from the search space of complex problem. Genetic Algorithm [21] works in two phases in the first phase selection is performed from the population and in the second phase selected generation is manipulated to create new generation. The working of Genetic algorithms is mainly based on three operations called selection, crossover and mutation. Selection is first phase operation that is used to select better parent. Crossover and Mutation are second phase operations used to manipulate the selected parents to generate effective better offspring for next generations. The pseudo-code used for finding an optimal DSS query allocation plan using genetic algorithm is given as below:

Step1: Input Data

Step 1.1 Select the DSS query based upon TPC-DS benchmark database.

Step 1.2 Decompose the DSS query into sub queries based upon different operation like selection, projection and join.

Step 1.3 Input various parameter like NoS, NoB, NoO, NoJ, NoF, NoSO, NoPO, IOC, CP, Comm, POPSIZE, Z, MaxGenr.

Step2: Initial Population

Step 2.1 Design Chromosome having length one less than the number of operations.

Step 2.2 Initialize the population of size Z with above said chromosome

Step3: Analyze the fitness

Step 3.1 Compute the fitness value of each chromosome from initial population based upon total cost of system resources.

Step4: Selection Operation

Step 4.1 Select best two chromosome that acts as parent based upon the fitness value.

Step5: Crossover Operation

Step 5.1 Apply one point crossover operation over two selected parents.

Step6: Mutation Operation

Step 6.1 Apply mutation operation on the resultant of crossover operation

Step 7: Termination

Step 7.1 generate DSS query allocation plan

Step 7.2 Goto step3 until MaxGenr.

Fitness Function

One of the major components of Genetic Algorithm is the fitness function. In simple terminology the fitness function [26] is defined as the objective of the problem that should be optimized. The fitness function considered for optimization in this case is the total cost associated with DSS query.

Fitness Function (TCDSSdq)= $\mathrm{LPC}_{dy}^{dq} + \mathrm{COMM}_{dy}^{dq}$ (1.4)

For providing an optimal result, Genetic Algorithm starts with fixed sized initial population. Every member from fixed size population is called chromosome [25]. In this case the chromosome is designed by considering the number of sites and the operation involved in a DSS query.

Genetic algorithm will provides us number of different DSS query allocation plans. The chromosome is designed in such a way that the length of chromosome is N-1; where N is number of operation involved in a DSS query. The design of chromosome for the selected DSS query is as given below:

Table1: Design of Chromosome

# of Operations	# of Sites	Design of Chromosome
11	4	2 3 1 3 4 2 1 2 3 1
21	10	2 3 5 7 3 4 5 3 5 7 8 2 1 4 5 6 4 3 4 3 5 6
30	10	6 5 3 7 3 9 5 3 5 7 8 2 1 4 5 6 4 3 4 3 5 6 7 5 3 7 9 1 4

The use of Genetic approach in the optimization process of DSS query reduces the total time involved in finding an optimal query allocation plan significantly, on the other hand GA compromises on the fitness value of q DSS query, i.e. the optimized plans produced by traditional exhaustive enumerative approach will be more optimal as compare to the query plan provided by GA. This work is an effort to analyse how Genetic Algorithm expedite the process of DSS query optimization in distributed database system. Genetic Algorithm is used because in complex DSS queries the exhaustive enumeration techniques failed to give query plan in feasible time. The simulation study shows that enumeration technique will take hours, days or even weeks for computing the optimal query allocation plan for complex DSS query in distributed database system. The analysis is performed by selecting three different DSS queries based on TPC-DS benchmark to know the effectiveness of Genetic Algorithm in the optimization process of DSS queries in distributed database system. A simulator is designed in MATLAB for analyzing the selected DSS Queries on dual core machine having 2GB of RAM.

6. ASSUMPTION AND RESULT OF SIMULATION

The simulator developed in MATLAB is used to find the effectiveness of genetic approach in optimizing the DSS queries in a distributed database system. A simulator takes number of input parameters like number of base relations, Number of Operations, Number of Intermediate Fragments, I/O speed coefficients, CPU Coefficients, Communication Coefficient, Number of Joins etc. Simulator takes a text file as an input and produces another text file with number of query execution plan for DSS query. The default ratio between Input Output Cost and Communication cost is assumed to be 1:1.6. It is assumed that the cost associated with communication is ten times than input output cost. All the base tables are assumed to be of same size. The block size of relation is used while computing the total cost of system resources. One of the important factors while simulating is the allocation of data on different sites. In this case it is assumed that this model places a base relation on two different sites. The above said DSS queries are analyzed by using the block structure of the concerned base relation. The following figure shows the optimal cost (Total Cost of System Resources) of selected DSS queries as computed by simulator.

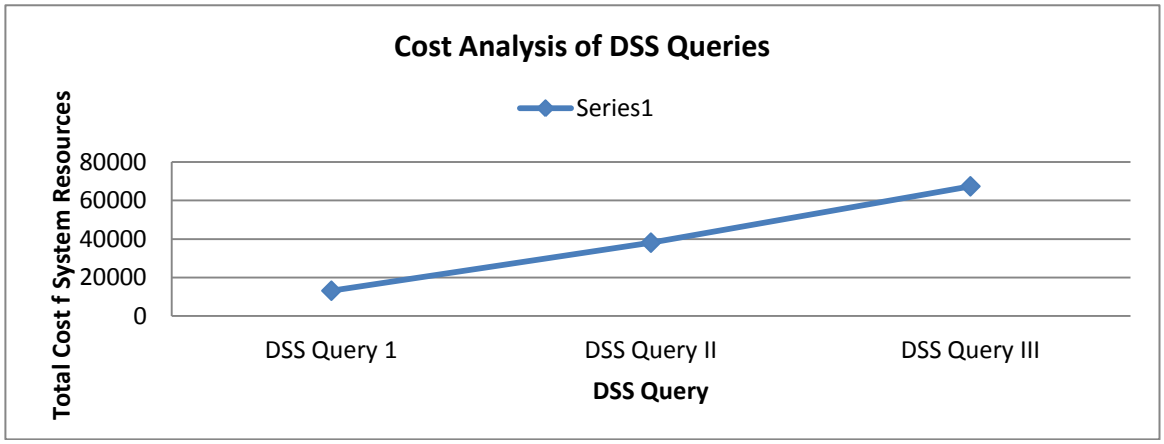

Figure 2: Cost Analysis of DSS Queries

The interesting factor is that where Exhaustive Enumeration technique is unable to find the optimal query allocation plan, the Genetic Algorithm has given the nearly optimal result in a flash as compare to Exhaustive Enumeration Technique.

To analyze the effect of varying communication cost with respect to input output cost over total cost of system resources, let us consider the following Table 2 that provides the total cost of system resources by varying communication cost.

The graphical representation of the data computed and analyzed in Table 2 is shown in the figure 3 (Analysis of DSS Queries).

Table2: Analysis of Total Cost of System Resources

Fitness Function	Input Output to Communication Cost Ratio					
	1:0.8	**1:1.6**	**1:2**	**1:3**	**1:4**	**1:5**
DSS Query 1	11243	13146	15054.6	16980.6	18920.5	21050.6
DSS Query II	34768.3	38148	40698	48638	53695	58210
DSS Query III	61234	67452	70313	79145	84761	104396

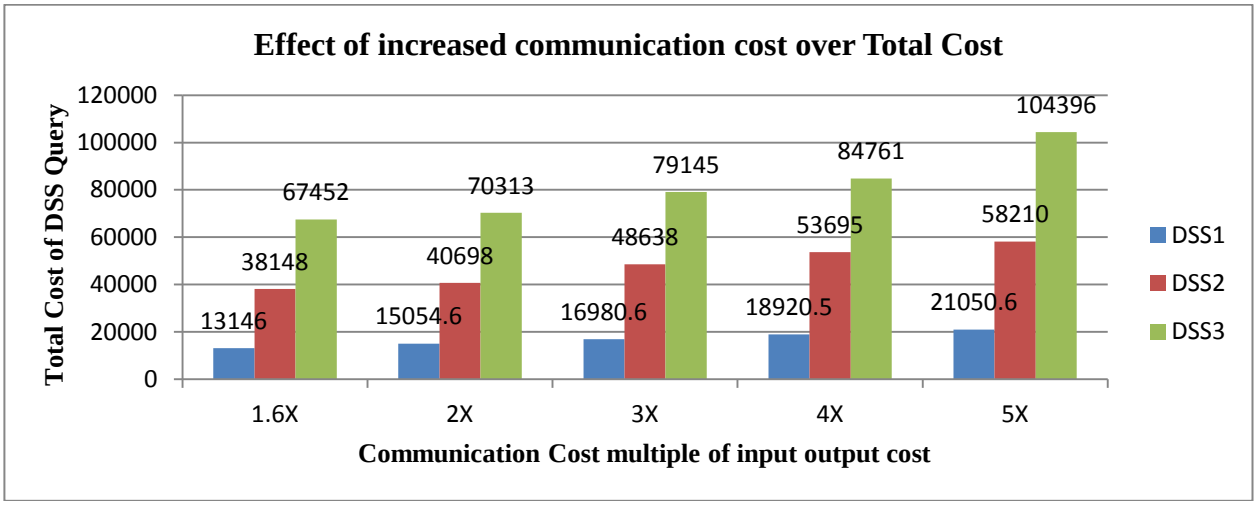

Figure 3: Analysis of DSS Queries

The following Table3 shows how total cost of system resources for selected set of DSS queries is increased by increasing the communication cost.

Table3: Gain in Total Communication Cost

DSS Query	Input Output to Communication Cost Ratio				
	1:1.6->1:0.8	**1:1.6->1:2**	**1:1.6->1:3**	**1:1.6->1:4**	**1:1.6->1:5**
DSS Query 1	1903	1908	3834.6	5774.5	7904
DSS Query II	2379.6	2550	10490	15547	20062
DSS Query III	2413.5	2861	11693	17309	36944

The following Figure 4 shows the percentage how Total Cost of System Resources (%) is increased by increasing the communication cost from 1.6 times of input output cost to 2,3,4, & 5 times respectively. The effect is observed by considering 1:1.6 (I/O Cost: Communication Cost) as base.

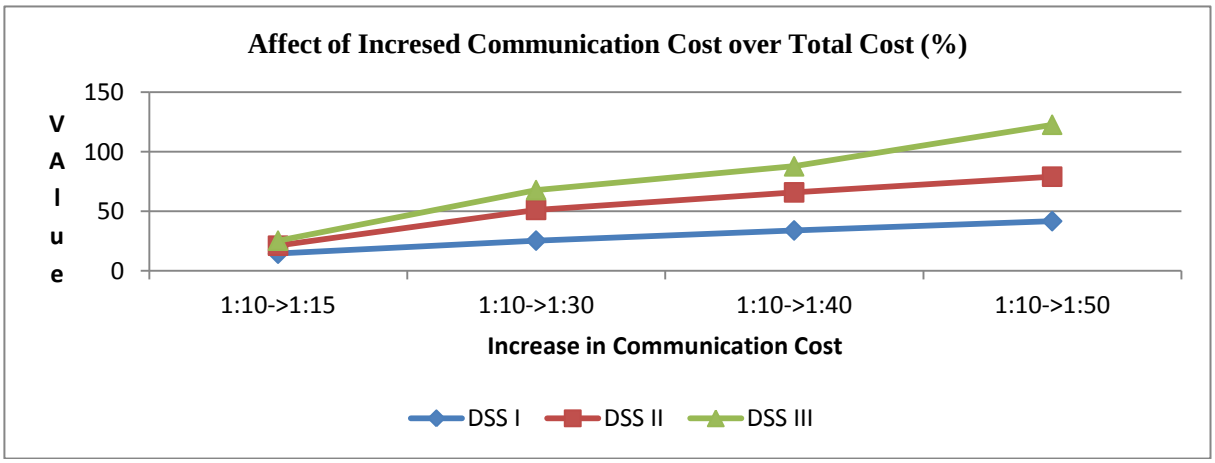

Figure 4: Analysis of gain in communication cost

From the above Figure 4 it is clear that cost of communication plays major role in the computation of Total Cost of system

resources used for the execution of a DSS Query. The Figure 3 suggests that by using faster communication media with respect to input output media one is able to drastically reduce the Total Cost of system resources. The study reveals that that exist linear relationship between Communication Cost and Total cost of system resources.

7. CONCLUSION

Query processing and optimization is one the dominant research area in the field of database theory. Previous research shows that a distributed query can be optimized by using different methods like exhaustive approach, dynamic approach, heuristic approach etc. In this study one of the important types of distributed queries i.e. DSS queries are analyzed and optimized stochastically using Genetic Algorithm by considering Total Cost of System Resources as Fitness Function. From the simulation result it is concluded that GA is best suited for optimization of DSS queries in distributed database system as compare to traditional optimization methods due to their incapability of providing optimized result in definite amount of time. The simulation study indicates that by eliminating the difference between the speed of input output and communication one is able to optimize the DSS query in an effective way. The cost of DSS query plan can be further optimized by reducing the mismatch between speed of input output and communication media. The simulation study reveals that there exist linear relationship between Communication Cost and Total cost of system resources. It is also analyzed that communication cost of a DSS query increases significantly with the increased number of joins and sites.

8. ACKNOWLEDGEMENT

Authors are highly thankful to Punjab Technical University, Jalandhar.

9. REFERENCES

[1] Nilarun Mukherjee, Synthesis of Non Replicated Dynamic Fragment Allocation Algorithm in Distributed Database System", Published in Proceeding of International conference on advances in Comp. Sc., 2010.

[2] Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database System", Fifth Edition, Pearson Education, Second Impression, pp 894, 2009.

[3] M. Tamer Ozsu, Patrick Valduries, "Principles of DDB System", Second Edition, Pearson Education, pp169.

[4] T.V.Vijay Kumar, Vikram Singh, "Distributed Query Processing Plans Generation Using GA", IJCTE, Vol 3. No.1, Feb 2011.

[5] Narasimhaiah Gorla, Suk-Kyu Song, "Subquery allocation in Distributed Database using GA", JCS & T, Vol. 10, No.1.

[6] Deepak Shukla, Dr. Deepak Arora, "An Efficient Approach of Block Nested Loop Algorithm based on Rate of Block Transfer", IJCA, Vol.21, No.3, May 2011.

[7] Swati Gupta, Kuntal Saroha, Bhawna, "Fundamental Research in Distributed Database", IJCSMS, Vol. 11, Issue 2, Aug 2011.

[8] Reza Ghaemi, Amin MilaniFard, Hamid tabatabee, "Evolutionary Query Optimization For Heterogeneous Distributed Database System", WASET, 43, 2008.

[9] Johann Christoph Freytag, "The Basic Principles of Query Optimization in Relational Database Management System", Internal Report, IR-KB-59, March 1989.

[10] Rajinder Virk, Dr. Gurvinder Singh, "Optimizing Access Strategies for a Distributed Database Design using Genetic Fragmentation", IJCSNS, Vol 1, No.6, Jun 2011.

[11] Clark D. French, "One Size Fits All- Database Architecture Do Not Work for DSS", SIGMOD 95, Published by ACM, USA.

[12] Sourabh Kumar, Gourav Khandelwal, Arjun Varshneyet. Al. "Cost-Based Query Optimization with Heuristics", International Journal of Scientific & Engineering Research, Vol. 2, Issue 9, Sep. 2011.

[13] Sangkyu Rho, Salvatore T. March, "Optimizing Distributed Join Queries: A GA Approach", Annals of OR 71, pp 199-227.

[14] PedroTrancoso, Josep-L.Larriba-Pey, Zheng Zhanget. Al., "The Memory Performance of DSS Commercial Workloads in Shared-Memory Multiprocessors", Published in the IEEE proceeding of the third International Symposium on HPCA held at San Antonio, USA, 1997.

[15] S. Vellev, "Review of Algorithms for the Join Ordering Problems in Database Query Optimization", Information Technologies and Control, 2009.

[16] Rajinder Singh, Gurvinder Singh, "A Stochastic Simulation of Optimized Access Strategies for a Distributed Database Design", IJSER, Vol 2, Issue 11, November-2011.

[17] Rajinder Singh, Dr. Gurvinder Singh, "Optimizing Access Strategies for Distributed Database Design using Genetic Fragmentation, IJCSNS, Vol. 11, No. 6, June 2011.

[18] TPC Benchmark DS, Version 1.1.0, April 2002 online: www.tpc.org.

[19] Manik Sharma, Gurdev Singh, Rajinder Virk, "Analysis of a DSS Queries in a Distributed Database System", IJNPC, Volume 1, Issue 3, Dec2012-Jan2013.

[20] M. Sinha, SV Chande, "Query Optimization using Genetic Algorithm", Research Journal of Information Technology 2 (3): 139-144, 2010.

[21] Noraini Mohd Razali, John Geraghty, "Genetic Algorithm Performance with Different Selection Strategies in Solving TSP", Proceeding of World Congresss on Engineering 2011.

[22] Zehai Zhou, "Using Heurisitcs and Genetic Algorithm for Large Scale Database Query Optimization", Journal of Information and Computing Science, Vol. 2, No. 4, 2007.

[23] Song Kyu Rho. Salvatre T. March, "Optimizing Distributed Join Queries: A Genetic Algorithm Approach", Annals of Operations Research, 7 (1997). \

[24] David E. Goldberg, " GA in Search Optimization and Machine Learning", Seventh Impression, Pearson.

[25] M K Pakhira, "A Hybrid Genetic Algorithm using Probabilistic Selection", Vol. 84, May 2003.

[26] Vinay Harsora, Dr. Apurva Shah, "A Modified GA for Process Scheduling in Distributed System", IJCA Special Issue on Artificial Intelligence Techniques- Novel Approach & Practices Applications, 2011.

[27] Kirti Nagpal, Vaishali Wadhwa, "Proposed Algorithm For Optimization Of Job Scheduling In Multiprocessor Systems Using Genetic Approach", International Journal of Computer Applications and Information Technology (IJCAIT), Vol 1, No. 3, 2012.

[28] Rachhpal Singh, "Task Scheduling with Genetic Approach and Task Duplication Technique", International Journal of Computer Applications and Information Technology (IJCAIT), Vol. 1, No. 1, 2012.

[29] Rachhpal Singh, "Genetic Algorithm for Parallel Process Scheduling", International Journal of Computer Applications and Information Technology (IJCAIT), Vol. 1, No. 2, 2012.

[30] Garima Mahajan, "Query Optimization in DDBS", International Journal of Computer Applications and Information Technology (IJCAIT), Vol. 1, No. 1, 2012.