Query Optimization using Clustering and Genetic Algorithm for Distributed Databases

S.Venkata Lakshmi
Dept. of Information Technology,
Gitam University, Visakhapatnam, India.
Email: svlakhsmi.2010@gmail.com

Dr. Valli Kumari Vatsavayi
Professor, Dept. of CS &SE,
Andhra University, Visakhapatnam, India.
Email: vallikumari@gmail.com

Abstract--**Query Optimization is principally a multifaceted exploration job that searches for best plan amongst the semantically equal plans that are obtained from any given query. The execution of any processing datasets essentially depends on the capability of query optimization procedure to acquire competent query processing approaches. A Distributed Database System (DDS) is a group of autonomous cooperating integrated procedure. Query at a specified place may necessitate information from distant places in a Distributed Environment. In query optimization, the cost is accompanied by every query execution plan. Cost is the summation of native cost that is I/O cost, CPU cost at every location and cost of transmitting information amongst locations. The key issue of a Query Optimization in a Distributed Database System is to obtain an effective query strategy with an efficient accuracy and minimum response time or cost to execute the given query. In this paper a novel methodology is suggested that selected the best query plan as to execute the given query employing Genetic Algorithm Strategy for Distributed Databases and a Clustering Approach within the databases so as to execute the query plan. Genetic Algorithms are extensively employed and acceptable methods for very challenging optimization problems. This proposed technique gives efficient performance in different environment. The Experimental analysis of the proposed methodology is carried out on 100 different queries distributed over 20 different sites having 8 relations in each query. This is compared with the DB2 distributed optimizers and achieved an increased reliability and high performance with respect to the optimization cost and accuracy for the queries in the distributed databases**

Keywords-Query Optimization, Genetic Algorithm, Distributed Databases, Clustering Approach, Similarity check algorithm

I. INTRODUCTION

Accumulative information in the databases of procedure specifically those that are multi cluster and physically widespread cause a lot of problems in data storage, retrieval and transmission. Traditionally, Distributed Database System is defined as a connection of several datasets that are scattered or dispersed physically but centralized logically with a combination of computer networks and database systems [5]. Distributed systems are a group of autonomous collaborating organizations that facilitates storing of information at physically distributed positions, depending on the frequency of admittance by consumers confined to a place. The distributed database allows merging information from these distributed positions using queries [8]. Thus, optimizing large queries in distributed relational databases is a difficult task as the discrete domain of different systems to answer a query is huge. Some of the existing Distributed Query optimization procedures offer accurate outcomes in different distributed atmosphere. The distributed query optimization techniques aid to explore exact information and mine the vital one.

In current era, with the progress of computer science and database technology, distributed database to a greater extent extensively employed with the intensifying applications where information requests are gradually being difficult and the proficiency demand for the query are progressively increasing. Thus, query processing is a crucial subject of the Distributed Database System. Query processing is the method of interpreting a query conveyed in an advanced language for example SQL to a low-level data manipulation procedure. Query Optimization is defined as a technique where the finest implementation approach for a specified query is obtained from a group of options. Database queries are optimized depending on cost models that compute costs for the query plans. The cost of a query plan hinges on constraints for instance base and intermediary relation cardinalities, accessible memory, disk bandwidth, processor speeds and survival of access paths. Throughout query optimization, the plan generated by the query optimizer is computed with the numerous approximations and measurements employed by the optimizer.

Query processing is additionally challenging in distributed atmosphere compared to centralized atmosphere since an enormous amount of constraints disturb the performance of distributed queries, associations that might be disjointed and pretended while seeking numerous positions to access, query response time might turn to be very huge [1]. It is reasonably that the success of a Distributed Database System analytically hinges on the capability of the query optimization procedure to originate well-organized query processing approaches. Query Optimization is a part of a database management system that endeavors to define the most effective technique to run a query. The optimizer deliberates the promising query plans for a specified input query, and tries to

978-1-4673-6680-9/16/$31.00 ©2016 IEEE

define which of those plans will be the further most effective. The objective of the Query Optimization is to diminish the whole cost of execution, minimization of response time, hastiness of information employed in responding the query [3] or precision of the information [4].

A. Genetic Algorithm

Considering the Bio-inspired Procedures like Evolutionary Algorithms (EAs) are the stochastic search approaches that simulates the usual genetic algorithm and communal activities of classes. These methodologies have been established to attain near-optimum results for significant optimization complications, where the conventional mathematical procedures might be unsuccessful. The initial evolutionary grounded procedure presented in the literature is the Genetic Algorithms (GAs). Genetic algorithm is introduced depending on the Darwinian principle of "survival of the fittest" and the usual way of progression done with re-generation. Genetic algorithms are becoming an extensively conventional technique for several famous challenging optimization problems. In this paper, the genetic algorithm is described for the implementation of the proposed methodology in distributed database query optimization.

Genetic Algorithm is functioned on query optimization where the initial population is produced arbitrarily. In Genetic Algorithm, the outcomes are known as individuals or chromosomes. Once the original Population is produced arbitrarily, genetic operations are implemented in an iterative manner unless certain stopping criterion is attained. Every execution of the iteration is known as a generation. Selection function is envisioned to progress the regular feature of the populace by specifying chromosomes of enhanced feature where an advanced probability to be considered into subsequent generation. The feature of a chromosome is measured by the Fitness Evaluation. The Fitness Evaluation proposes the optimum result such that specific individual might be categorized in contrast to all the additional chromosomes. For every generation, the genetic functions are computed and therefore the population advances, typically minimizing the mean cost of its chromosome by obtaining minimum number of sites of relations to execute a query. Once the best optimum plan is acquired, it is given to the dataset device as to implement the query.

Crossover and mutation are significant functions of genetic algorithm. Crossover chooses genes from parent chromosomes and generates new offspring. Once the crossover is accomplished, mutation operation is computed. It is to avoid deteriorating from all the outcomes in a population to a native optimal of resolved problem. Mutation arbitrarily alters the new offspring. The GA is a conventional search methodology that applies principles of biological selection to a casually produced group of populations comprising of chromosomes where each signifies a comprehensive solution, and by means of these preliminary solutions attempts to enhance improved ones. For every chromosome, a fitness value is computed to select most economical chromosomes that will create the subsequent generation.

B. Motivation

The complication and costs increased along with the increasing number of associations in the query. Owing to enormous number of constraints on disturbing query execution cost, a particular query is implemented in numerous diverse techniques. A query execution approach is prerequisite to diminish the cost of query processing. In distributed databases, information might be simulated at innumerable positions spread all over the network and diminishing the volume of data broadcasted is significant to decrease the query processing cost. Thus, there can be many possible combinations of relations and number of different query plans that provide answer to the given query. The distributed query optimization has numerous complications associated to the cost prototype, larger group of queries, optimization rate, and optimization intermission [2]. The significant issue for query optimization in a disseminated database is choice of the maximum cost efficient plan to accomplish a query [7]. Therefore, it is needed to retrieve the most preferred query plan that results in effective query processing.

In order to address the above issue of query optimization, author proposed a novel methodology for an efficient query optimization technique that executes a query with less cost and less response time. Two phases are proposed in this paper for efficient optimization technique. In first phase, an evolutionary approach known as genetic algorithm is employed to obtain closely related or minimum number of different database sites for a given query, which is given as an input to the second phase. In this phase a clustering technique is employed where the given query is matched with the existing database of query template cluster. The foremost issue in Query Optimization is that, the exploration area is multifaceted and genetic algorithms are hypothetically confirmed to offer strong examination in multifaceted areas. The GA algorithm is computationally meek however prevailing in their exploration for enhancement. Therefore, the usage of Genetic Algorithm in the Query Optimization is consequently suitable.

C. Organization of the paper

A Brief discussion of Query Optimization and Genetic Algorithm is given in this Section. The section 2 gives the Literature Survey of different Query Optimization Techniques using Different Approaches. The section 3 discusses the proposed approach of the Query Optimization that employed

Genetic Algorithm and Clustering Technique. The experimental results and its analysis is briefly given in section 4. Finally Section 5 concludes the proposed methodology.

II. LITERATURE SURVEY

Query optimization is one of the study area of database systems, even though numerous investigators have finished huge amount of effort, however not adequate by means of useful way of disseminated database technology in information processing is query optimization, which is a challenge that is not healthy determined in relational database systems. Query optimization is a huge part within the database arena. Jarke et al. [6] and Mannino [7] give an extensive survey of the available work into this field. Recently, randomized algorithms are used in query optimization. The evolutionary computation is originated as a feasible method in which Genetic Algorithm (GA) is taken for query optimization [11].

Li et al., [9] deliberated optimization approaches to augment the query optimization as to acquire an optimum strategy. The foremost problem employing sub-queries is the intra query replication where the sub-query has similar number of tables and circumstances, which are aimed at external query. Nevertheless, this worsens the query quality if query has associated iterative enquiries. The authors suggested certain experiential approaches for augmenting query processing. The initial approach suggested is to execute the selection function primarily so as to frontier the number of rows or tuples. The subsequent suggested technique is to frontier the number of columns by accomplishing prediction function. Then, execute the functions by means of minor and meek join initially if there exist successive joins in the query. Lastly store the outcome of the similar relation for upcoming usage.

Chande et al., [10] concentrated on join ordering issues in relational database and employed Genetic Algorithm to minimize the difficulty of competent choosing of join ordering for creating an optimum strategy by an optimizer. The queries employed for genetic query optimizer (GQO) are implemented in diverse atmosphere as to match their accuracy. The implementation periods for entire queries are matched with suggested GQO. Authors accomplished experimentations to associate GQO with PostgreSQL, DB2 and MySQL. Sun et al., [12] suggested a paging query procedure for comprehensive information to enhance query competence and complete application. By means of this technique, complete information sustaining the entire query environment is initially positioned at server's memory and solitary the portion of information desired by the customer is delivered to customer. This methodology is in contradiction to conventional approach of paging query where whole information is accumulated near to customer place which is evidently not an effectual method to handle immense information since it might be a pointer to obstruct the client structure assets.

Hameurlain [13] discovered the evolution of query optimization procedures from integrated Database method to data grid methods. The optimization is deliberated in single processor, distributed, parallel dispensation and significant atmospheres. Optimization approaches and its exceptional features are defined on every situation. In the suggested outcomes of planning issues of single processed relational systems, both the exploration approaches are arbitrarily concentrated.

Lin [14] offered the query optimization stream comprising of numerous components on disseminated datasets. The user component in distributed system examines the user query demand. The System Enquiry component observes verdict of the query, where the semantics, grammar and spellings are tested through altering the query into its equivalent structure. This equivalent tree is given to query tree renovation component that alters it to the comprehensive query tree rendering to data structure designated in query tree. The comprehensive query tree acknowledged from query tree transformation component is plotted to its equivalent physical operator's trees using the optimizer component. Formerly the optimizer component chooses a corporal operator tree using minimum cost. Herodotou et al., [15] concentrated on the optimization of queries executing on segregated tables by means of offering procedure to primly choose the finest implementation strategy. Segregated tables deliver multiple benefits to database systems comprising query trimming, i.e., fast query processing, admittance to information in parallel approach, competent techniques to store information, to holdup information, to preserve measurements for DML tasks, improved cardinality assessment and to evade disintegration. Predominantly, query optimization is not an informal job for huge volume of information. The authors suggested a segregated conscious procedure for PostgreSQL optimizer that produces strategies more enhanced compared to existing optimizer over an enhanced cardinality approximation and enriched examination area.

Bruno et al., [16] observed certain techniques to progress the deprived strategies nominated by the optimizer over query clues. By means of expressive clues, optimizer preserve on refining the strategy unless a finest plan is selected. The authors recommended Phints to seizure conceivable clues for optimizer to acquire healthier strategy. Query clues are a non-trivial multifaceted approaches for attaining improved strategy. Kratica et al [17] suggested a Genetic Approach for resolving the ISP (Index Selection Problem) i.e. the issue of diminishing the reply period for a certain dataset capability with a recommended selection of keys. Its Genetic Approach is grounded on binary coding, data structures for assessment of the independent estimation on the uniform crossover and humble

mutation. The procedure is verified on group of demanding occurrences identified from the study and determine the outcomes acquired specifying its competence and consistency.

Fang et al [19] suggested a novel multi-copy join optimization technique grounded on genetic algorithm to achieve comprehensive query redrafting using replications into consideration. The recommended join optimization technique contemplates together selecting redundant duplicates of a universal vision and probing for a comparative low-cost join order using its encrypting structures and distinct genetic functions. Chande and Sinha, [18] have examined earlier investigations on the usage of genetic algorithm to query optimization and offered an agenda for query optimizer. They also accepted a relative investigation of the genetic join order optimizer.

III. PROPOSED METHODOLOGY

Since the amount of associations from which the information is required rises, the amount of different results for the query likewise up surges extremely for different distributed databases. Query optimization is therefore summarized to an exploration issue where DBMS desires to discovery optimal execution strategy in a massive search area. Every execution idea might be a conceivable result for the problematic of discovering respectable admittance route to recover essential information. The author proposed this methodology for an efficient query optimization technique in two phases as to retrieve the query plan with minimum response time or cost and efficient accuracy for the given distributed query. In first phase, the Genetic Algorithm is used to obtain the query plan containing the essential information that resides in the lesser sites and leads to effective query processing. This approach generates the query plan depending on understanding of the information essential to respond to the user query. The generated query plan from distributed databases with fewer sites is given as input to the second phase of the proposed methodology. In this phase, a clustering technique is employed which groups the related queries into clusters and employs the optimizer introduced strategy for the cluster demonstrative to implement whole upcoming queries allocated to the cluster. The Flow Chart for the proposed Methodology is given in Fig 1.

A. Retrieval of Fewer Data Sites from Distributed Databases Using Genetic Algorithm

In this approach a query plan is generated grounded on understanding of information essential to reply the user query. The understanding of information depends on number of sites entailed in a query plan which is generated apart from the query plan which involve large number of sites. The procedure is based on the genetic approach as given in Algorithm 1. The methodology considers the relationships as inputs in the FROM clause of the given query alongside with the locations comprising them, likelihood of the Crossover and Mutation and pre identified number of iterations.

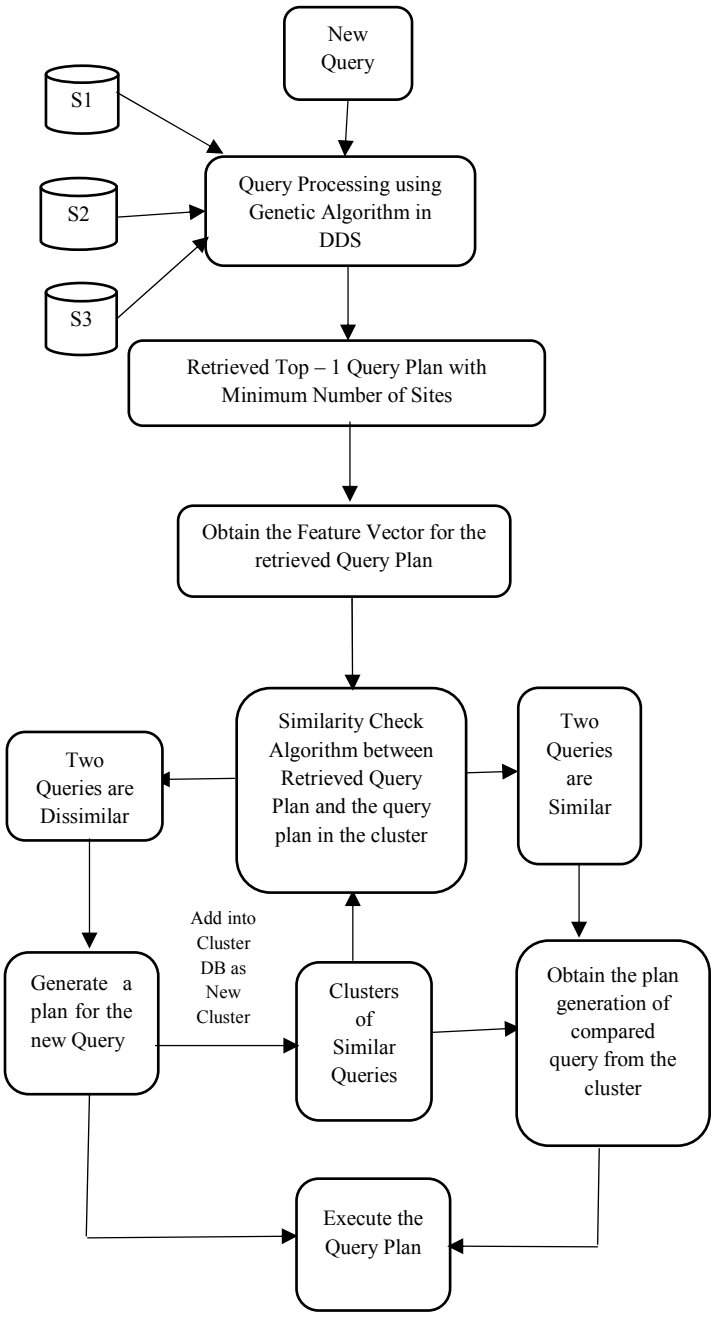

Fig 1: Proposed Methodology

Using GA, it generates a population of individuals where every individual signifies a query plan. The fitness function that is employed in this approach is Query Site Cost (QSC) and is given as

$$QSC = \sum_{i=1}^{M} \frac{S_i}{N}(1 - \frac{S_i}{N})$$

M is number of sites retrieved using query strategy, Si is the number of times the ith site is employed in the query strategy and N is the number of relations retrieved using query. The QSC differs from 0 to N (N-1). The minimum is the QSC values obtain, the fitter is the query plan to the approach. The two fittest query plans is with minimum QSC values are selected and Crossover & Mutation Operations is performed to engender the population for subsequent iteration. This remains until the procedure executes for a prior identified number of iterations. Thus, the top-1 query plan is retrieved depending on QSC values.

Algorithm 1:

Input: RS → Relations in the FROM Clause of the user query alongside the locations comprising it

P_c→Probability of Crossover

P_m→ Probability of Mutation

G_p→ Pre-identified number of Iterations

Output: Top -1 Query Plan

Method: generation = 0

1. Create preliminary population for query plan P_{qp}compliant to RS.
2. Compute QSC of every query plan depending on closeness heuristics.
3. If Generation >G_p
4. Return Top – 1 query plan.
5. Select any two query plans employing unbiased selection methodology.
6. Apply crossover with probabilityP_c.
7. Apply mutation with probabilityP_m.
8. Generation = generation + 1
9. Go to step 2

B. Execution of the query plan using Clustering Approach:

The retrieved query plan with minimum number of sites from the distributed databases are given as input to the second phase as to execute final query result. A Feature Vector is defined for the retrieved query plan that consists of information for the query. The whole query feature route is given in the Table 1. The features are segregated into Global Features that have query varied information and table features that are significant to specific tables. This information comprises together complete structural features for instance number of tables and joins in the query and table precise characteristics for example existence of keys on the query attributes, the number of predicates where the table is entailed and the dimension of the table. A similarity procedure is given to consider a group of query feature vector and measurably calculate the similarities of the two query vectors. The Algorithm for the similarity checking is given in Algorithm 2.

TABLE 1: FEATURE VECTORS FOR SIMILARITY CHECK ALGORITHM

Feature	Characteristics
Global Characteristics	
NTQ	Number of tables contributing to the query
DSQ	Degree sequence of query
JP	Number of join predicates
JC-I [0…2]	Number of Join Predicates with index features of 0, 1 and 2, correspondingly
$NPCsarg$	Number of SARG$able$ predicates
$NPCnsarg$	Number of non- SARG$able$ predicates
Table Characteristics	
DT_i	Degree of table T_i
IF_i	Boolean representing index solitarily admittance to T_i
$PCsarg_i$	Number of SARGable predicates on table T_i
$PCsarg_i$	Number of non-SARGable predicates on T_i
JIC_i [0…2]	Number of Join Predicates of index characteristic 0, 1 and 2 involving T_i
TS_i	Dimension of T_i
ETS_i	(estimated) Effective magnitude of T_i

Algorithm 2: Similarity Check (q1, q2)

1. IF NTQ (q1)! = NTQ(q2) Return (Not Alike); // Examine whether Queries have same number of Tables
2. IF DSQ(q1) = DSQ(q2) AND JP(q1) = JP(q2) AND $NPCsarg(q1) + NPCnsarg(q1) = NPCsarg(q2) + NPCnsarg(q2)$go to Line 4.
3. Return (Not Alike);
4. For each Group G of tables having similar degree //Find Best Mapping between Tables

$$R_1 = T^1{}_1, T^2{}_1, T^3{}_1 \dots\dots\dots T^k{}_1 R_1 \subseteq q_1$$
$$R_2 = T^1{}_2, T^2{}_2, T^3{}_2 \dots\dots\dots T^k{}_2 R_2 \subseteq q_2$$

5. Specify the mapping of well-matched tables amongst R_1 and R_2 which have the smallest cumulative distance, $mindist_g$ relating to the pair wise table distance task

$$dist_i(T^i{}_1, T^i{}_2) = \frac{w1 * \left|TS^i{}_1 - TS^i{}_2\right| + w2 * \left|ETS^i{}_1 - ETS^i{}_2\right|}{\max\left(TS^i{}_1,\ TS^i{}_2\right)}$$

6. $Totaldist = \sum_{g \in G} mindist_g$ // Calculate Distance amongst Queries
7. IF $Totaldist > threshold$ RETURN (Not Alike);
8. RETURN (Alike);

Because all the queries in a group are alike in feature space, a solitary query is the illustrative of the group. Thus solitary single distance estimation is essential to find how a novel query fits to a group. Entire queries in a group have identical strategies in the plan space. A single plan for every cluster to execute a query may be sufficient in the plan space since all the queries in the cluster are similar. Therefore time taken to obtain query for the clustering approach in the proposed methodology is O(k) where k is the number of clusters in the approach compared to O (n) where n is number of query plans the plan space for traditional optimizers. Sometimes more than single cluster might have identical execution strategy.

IV. EXPERIMENTAL RESULTS

The performance of the proposed methodology is carried on 100 queries distributed over 20 sites or 20 different databases with utmost 8 relations in each query. The Genetic Algorithm grounded approach is executed in MATLAB 7.4 in a Window XP environment. The Genetic Algorithm for the proposed methodology runs for given number of Iterations G_p and the QSC cost for each query is given with the probabilities of crossover and mutation. The average QSC values against specified number of generations with P_c=0.6 and P_m=0.05 is given in table 2, when compared to other probabilities that are experimented by the user such as P_c=0.6 and P_m=0.1, P_c=0.9 and P_m=0.05, P_c=0.9 and P_m=0.1. The graph represents that the convergence of genetic algorithm at minimum QSC is 0.25 at P_c=0.6 and P_m=0.05. Thus, this is the utmost appropriate values to produce the query plan for the given 100 queries distributed over 20 sites.

TABLE 2: NUMBER OF GENERATIONS G_P VERSES QSC VALUES FOR PROBABILITIES PC=0.6 AND PM=0.05 FOR THE PROPOSED METHODOLOGY

SNO	Number of Generation (G_p)	QSC values
1	10	0.75
2	20	0.68
3	30	0.55
4	40	0.42
5	50	0.26
6	60	0.25
7	70	0.25
8	80	0.25
9	90	0.249
10	100	0.248

From the above results, it is observed that the proposed Genetic Algorithm based approach in phase one gives better results for the Clustering approach in Phase two for the probabilities P_c=0.6 and P_m=0.05. The performance of the proposed methodology is carried out by experimenting on the given 100 query datasets. In the proposed methodology, the weight values *w1* and *w2* are given as 0.7 and 0.3 and the threshold value is given as 0.01 which gives the satisfactory accurate results for similarity check algorithm in the clustering approach. The table 3 gives the Cost and Accuracy of the proposed methodology for ten different queries when compared to the existing traditional DB2 Optimizer.

TABLE 3: COST AND ACCURACIES OF 10 DIFFERENT QUERIES FOR PROPOSED DISTRIBUTED QUERY OPTIMIZATION TECHNIQUE AND TRADITIONAL DB2 OPTIMIZER

Query No	Distributed Query Optimization Technique		Traditional DB2 Optimizer	
	Cost(in 's')	Accuracy	Cost (in 's')	Accuracy
#1	0.005	97	0.1s	96
#2	0.005	97	0.1	96
#3	0.005	97	0.1	96
#4	0.004	94	0.09	97
#5	0.005	98	0.1	97
#6	0.004	94	0.09	96
#7	0.004	96	0.09	96
#8	0.005	97	0.1	97
#9	0.005	90	0.1	96
#10	0.005	91	0.1	97

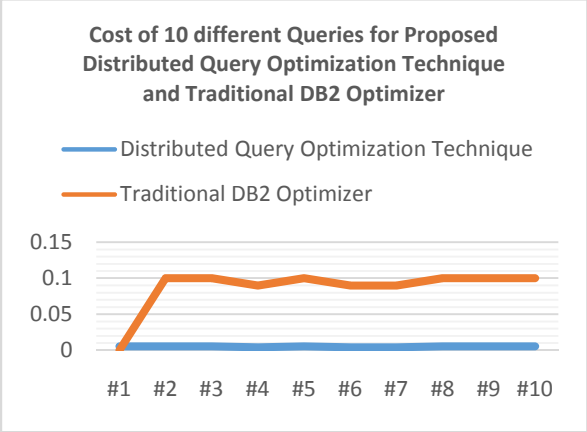

Fig 2: Cost of 10 different Queries for Proposed Distributed Query Optimization Technique and Traditional DB2 Optimizer

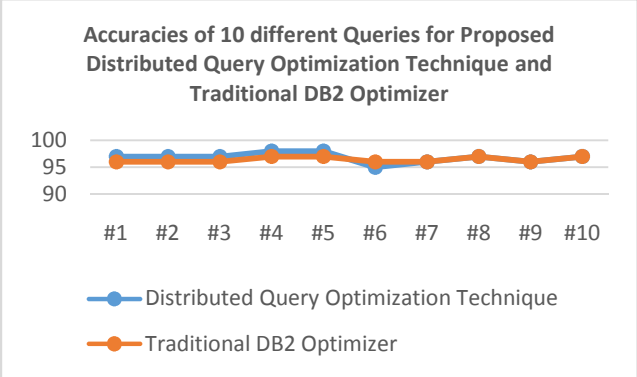

Fig 3: Accuracies of 10 different Queries for Proposed Distributed Query Optimization Technique and Traditional DB2 Optimizer

The cost of this approach is defined as time taken to execute a given query for the proposed methodology. Fig 2 represents the different cost values for the proposed methodology verses existing DB2 Optimizer for 10 queries. Fig 3 represents the accuracies for the proposed methodology verses existing DB2 Optimizer for 10 queries. From the results, it is clearly observed that even though the accuracy obtained for the DB2 Optimizers is somewhat better when compared to the proposed Distributed Query Optimization technique, the cost or the time taken to retrieve the best query strategy or query plan for the given query is efficient in the proposed Distributed Query Optimization Technique when compared to traditional existing DB2 Optimizer.

V. CONCLUSION

The proposed Distributed Query Optimization technique gives a methodology to generate an efficient distributed query processing plan which improves the reply time of user queries. This methodology achieved the distributed query processing plan generation as a single-objective genetic algorithm problem. The fitness evaluation function employed in the proposed Genetic Algorithm is the Query Site Cost (QSC) and a Similarity Check Algorithm is used to find the similarities between the Feature Vectors of both the queries in the clustering approach. The proposed methodology with the Genetic Algorithm is robust, perform well at the start of the search and makes sustained progress to better query plan in the search space. The increased performance achieved by the proposed methodology is in terms of the cost or the time taken to execute a given query plan and also achieved reliable accuracy when compared to the existing traditional DB2 Optimizer. The proposed approach attempts to avoid redundant pre-processing of the plan space. The experimental results also showed a potential for achieving valuable cost reduction.

REFERENCES

[1] Li, Victor O.K. "Query processing in distributed data bases", MIT. Lab. for Information and Decision Systems Series/Report no. LIDS-P; 1107, 1981.

[2] Reza Ghaemi, Amin Milani Fard, Hamid Tabatabaee, and Mahdi Sadeghizadeh, Evolutionary Query Optimization for Heterogeneous Distributed Database Systems", World Academy of Science, Engineering and Technology 43, 2008.

[3] C. Olston and J Widom. “Offering a precision-performance tradeoff for aggregation queries over replicated data.” In VLDB, 2000.

[4] R. Avnur, J M. Hellerstein, B. Lo, C. Olston, B. Raman, V. Raman, T. Roth, and K. Wylie. “Control Continuous output and navigation technology with refinement on-line.” In SIGMOD, 1998.

[5] Cyrus Shahabi, Latifur Khan, Dennis Mcleod.” A probe based technique to optimize join queries in distributed internet bases, Knowledge and Information Systems.”20002.

[6] M. Jarke, J. Koch, “Query optimization in database systems.” ACM Computing surveys, Vol. 16, no. 2, pp.111-152, 1984.

[7] M. V. Mannino, P. Chu, T. Sager, “Statistical profile estimation in database systems.” ACM Computing Surveys, Vol. 20, No. 3, pp. 191-221, 1988.

[8] Barry E. Jacobs, Cynthia A. Walczak, “ Optimization algorithms for distributed queries”, IEEE transactions on software engineering, Vol. SE-9, No.1, January-1983.

[9] D. Li, L. Han and Y. Ding, “SQL Query Optimization Methods of Relation Database System”, Computer Engineering and Applications (ICCEA), 2010.

[10] S. Chande and M. Sinha, "Genetic optimization for the join ordering problem of database Queries", India Conference (INDICON), 2011.

[11] H. Dong and Y. Liang, "Genetic algorithms for large join query optimization," In the proceedings of the Ninth annual conference on Genetic and Evolutionary Computation (GECCO), London, England, pp- 1211-1218, 2007

[12] F. Sun and L. Wing, "Paging Query Optimization of Massive Data in Oracle 10g Database", Computer and Information Science and Service System (CSSS), IEEE International Conference, 2011.

[13] A. Hameurlain, "Evolution of Query Optimization Methods: From Centralized Database Systems to Data Grid Systems", Proceedings of the 20th International Conference on Database and Expert Systems Applications, 2009.

[14] X. Lin, "Query Optimization Strategies and Implementation Based on Distributed Database", Computer Science and Information Technology, 2nd IEEE International conference, 2009.

[15] H. Herodotou, N. Borisov and S. Babu, "Query Optimization Techniques for Partitioned Tables", ACM SIGMOD International Conference on Management of data, 2011.

[16] N. Bruno, S. Chaudhuri and R. Ramamurthy, "Power Hints for Query Optimization", IEEE International Conference on Data Engineering, 2009.

[17] Kratica, J., I. Ljubiae and D. Tosic, A Genetic Algorithm for index selection problem. Lecture Notes Computer Science, 6211: 281-291, 2003.

[18] Chande, S.V. and M. Sinha, 2008c. "The use of genetic algorithms for optimization of relational database queries:" A comparative study. Proceedings of the IEEE sponsored National Conference on Applications of Intelligent Systems-2008.

[19] Fang, L., P. Wang, J. Yan, 2008. "A multi-copy join optimization of information integration system based on a genetic algorithm." Proceedings of the 2008 3rd International Multi-Conference on Computing in Global information Technology, July 27-Aug. 01, Washington, DC, USA., pp: 223-228, 2008.