American Journal of Computing Research Repository, 2015, Vol. 3, No. 2, 14-17
Available online at http://pubs.sciepub.com/ajcrr/3/2/1

DOI:10.12691/ajcrr-3-2-1

Estimation of Fitness Parameter along with Weight of Query Plans in Distributed Database Environment Using Genetic Algorithm Techniques

Sambit Kumar Mishra[1,*], Srikanta Pattnaik[2], Dulu Patnaik[3]

[1]Department of CS&Engg, Ajay Binay Institute of Technology, Cuttack
[2]S.O.A. University, Bhubaneswar
[3]Government College of Engineering, Bhawanipatna
*Corresponding author: sambit_pr@rediffmail.com

Received May 21, 2015; Revised May 29, 2015; Accepted May 31, 2015

Abstract The transmission cost of the data as well as query may be similar between heterogeneous systems which may be visualized as a linear function of the size of the data. The response time of a query schedule may be seen as the time elapsed between the start of the first transmission and the time at which the attributes in the relation arrive at the required system. The minimum response time of a relation may be visualized as the minimum response time among all possible query schedules for the relation. The total time of a query schedule may be obtained as the sum of the costs of all transmissions required in the schedule. The incoming selectivity of a query schedule for a relation may be defined as the product of selectivities of all the attributes in the schedule. Usually the distribution strategy for a query consists of the schedules for all relations that may not reside in the result node used in the query. It is seen that the data transmission costs may be significantly greater than local processing costs in the system and the cost of processing a query may be determined by the transmission costs in its distribution strategy. In this paper, it is aimed to evaluate the fitness parameter along with the weight of the query plans in the distributed database environment.

Keywords: *query plan, query schedule, tuple, join, genetic algorithm, attribute, join*

Cite This Article: Sambit Kumar Mishra, Srikanta Pattnaik, and Dulu Patnaik, "Estimation of Fitness Parameter along with Weight of Query Plans in Distributed Database Environment Using Genetic Algorithm Techniques." *American Journal of Computing Research Repository*, vol. 3, no. 2 (2015): 14-17. doi: 10.12691/ajcrr-3-2-1.

1. Introduction

A distributed database system is characterized by the distribution of the system components of hardware, control, and data. The database may be viewed logically in the relational data model. The database may be allocated across the system in units of relations. The relation distribution allows a general manner of redundancy. The distribution of data may be confined and restricted to the user. In a distributed database, joins on relations may be computed located at different locations. Instead of computing the joins immediately, the sizes of the relations may be reduced first wherever possible by restrictions and projections. Usually all the database operations are performed by using the concept of a query, where query may be viewed as 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. Depending upon the location of data a query may be defined as centralized or distributed query.

It is understood that minimization of the total time of the objective algorithm generally comes from the use of the algorithm in a multiprocessing environment. Minimizing response time leads to an increased number of parallel data transmissions in the query processing strategy. In multiprocessing systems, under moderate to heavy loads, these extra transmissions may lead to significant synchronization delays which may cause poor query response time. By minimizing the total time in a query processing strategy, fewer transmissions may be included and improved actual response times may result in certain system environments. The queries associated with the decision support system may be one of the decisive types of distributed query. These queries may be complex and time consuming in nature. Due to the decentralization of data and the complexity of query, it becomes mandatory to optimize the queries in distributed database system. Depending upon the location of data a query may be visualized as centralized or distributed query. Distributed queries may be more complex to handle and optimize as compared to centralized database queries. One of the important characteristics of distributed queries is that their running time is normally unpredictable as compare to online transaction processing queries. Due to dissemination of data over several locations one important issue in the optimization of distributed query is the placement of data and program across the different

locations available in distributed database system. In this situation, Genetic Algorithm is used to compute the total cost of distributed query based on the Input-Output cost, processing cost and communication cost. All the basic operations of genetic algorithm like selection, crossover and mutation are implemented to find an effective distributed query allocation plan in distributed database system. The use of genetic algorithm helps in finding an effective query allocation plan as compared to significant time taken in the traditional query optimization technique.

2. Review of Literature

N. Tatbul et.al [1] have focused in their paper about the data completeness that measures the percentage of incoming stream data that are used to compute the query results. They have also suggested that due to the bursty nature of stream data, sampling and load shedding techniques may be used to reduce the system workload in case of overload.

Babcock et. al. [2] have proposed a scheduling algorithm called chain scheduling. In their research work they found that the algorithm may be viewed as almost optimal in minimizing run-time memory usage for single-stream queries involving selections, projections and foreign-key joins with stored relations.

W. W. Chu et.al [3] have formulated a solution space of feasible processing strategies for distributed queries. They have also developed an optimal, exponential, optimization algorithm. They have suggested that the use of the semi-join operation may lead to the development of full-reduction methods of processing a distributed query.

E. Wong et.al [4] have discussed in their paper that in a distributed database, it is better to do local processing first because it reduces the amount of data to be transmitted. With local processing, we mean the computation of restrictions, projections, and semi-joins between relations that reside in the same node. After the initial processing, each node that has query data will be considered to contain only one "integrated" relation. The relations at each node remain distinct. However, by reformatting the query so that each node becomes a variable, the distribution aspects of the query are emphasized.

Y.E. Ioannidis et.al [5] have discussed in their paper that most of the queries on distributed relational databases require access to relations from multiple sites for their processing. The number of possible alternative query plans increases exponentially with increase in the number of relations required for processing the query.

M. Jarke et.al [6] have elaborated in their paper that the problem associated with distributed databases is usually a combinatorial optimization problem and may be addressed by simulated annealing, iterative improvement, two-phase optimization techniques, etc. The techniques that may reduce the search space are usually based on plan transformation.

T T.V. VijayKumar et.al [7] have proposed in their paper an approach that uses Genetic Algorithm (GA) to generate ‘close’ query plans. It aims to generate query plans that are optimal with respect to the number of sites involved, and the concentration of relations in these sites, for answering the user query. This in turn would result in efficient query processing.

J. H. Holland et.al [8] have discussed in their paper that as the data required to process the user query is spread over various sites, there may be a need to arrive at a query processing plan that entails an optimal cost for query schemes, fitness function, and selection of parent’s, genetic operators.

M. Mitchell et.al [9] have discussed in their paper that the fitness value of each chromosome in the population, using the fitness function, is evaluated. Designing the fitness function is one of the most crucial aspects of GA as it reflects the “fitness” or “figure of merits” of a chromosome or population indicating the ability or utility of that chromosome. The fitter individuals are then selected for crossover and mutation to arrive at the population for the next generation.

William I et.al [10] uses an adaptive selectivity estimation scheme for multidimensional queries which performs better than non-adaptive methods when the distribution of the data is not known. This research overcomes the disadvantages of previously formulated non-adaptive, static methods which are relatively inaccurate in a dynamic database.

In distributed query optimization, one of the major components is generation of sub-query allocation plan. A complex, high level distributed query is divided into a sequence of smaller, simpler sub-queries. Original query and low level query should have the same syntax that is producing the same results. In order to reduce total cost of the query, these sub-queries need to be executed on various different sites of distributed database. The total cost that will be occurred in processing the query is a good measure of resource consumption. M.Tamer ozsu et.al [11] have discussed in their paper that usually in a distributed database system, the total cost includes CPU, I/O and communication cost that needs to be minimized. An optimizer’s cost model includes cost functions to predict the cost of operators, statistics, base data and formulas. The cost is in the terms of execution time, so a cost function represents the execution time of a query.

Surbhi Bansal et.al [12] have discussed in their paper that cardinality may be evaluated for intermediate results of the query by calculating selectivity factor at run time using selectivity formulae. Thay have also seen that the overall cost of the query may be directly proportional to the cardinality of the intermediate results.

Rajinder Singh et.al [13] have proposed a stochastic model simulating a distributed database environment and shown benefits of using innovative Genetic Algorithms for optimizing the sequence of sub-query operations allocation over the Network Sites. They have also analyzed the effect of varying Genetic Parameters on Solution’s quality.

Faiza Najjar et.al [14] have proposed a statistical method for estimating the cardinality of the resulting relation obtained by relational operator by using sample based estimation that execute the query to be optimized on small samples of real database and use the results of these trials to determine cost estimates.

LARSON, P et.al [15] have worked on the problem of deriving query results based on the results of other previously executed queries is also related to the problem of multiple-query optimization. In the the problem of common subexpression isolation, several different formulations have already been cited under various query

language frameworks such as relational algebra, tuple calculus, and relational calculus.

Carlo et.al [16] have proposed a method for estimating the size of relational query results. The approach is based on the estimates of the attribute distinct values. In particular, the capability of analytic method to estimate selectivity factors of relational operations is considered. They also presented some experimental results on real databases which show the promising performance of analytic approach.

B.M. Monjurul et.al [17] have discussed in their paper that in distributed query processing environment, the query may be decomposed into a group of sub-queries that are to be executed on different sites. The aim of the query optimization is to decide a least cost query execution plan among various feasible plans.

Ridhi kapoor et.al [18] have described the selectivity and cost estimates in query optimization in distributed databases. They have discussed the various cost formulations to evaluate the cost of execution plans and then executing the plan with the minimum cost to the objective function.

3. Problem Formulation with Experimental Evaluation

3.1. Algorithm

Step1 : Evaluate the query plan

Step2 : Calculate the position and the size of query plan

Step3 : Evaluate the weight of the query plans after calculating the size of the query plans

Step4 : Calculate the fitness value of query plans after inputting processing time, number of operations and number of terminals

Total number of operations =20

No. of terminals =20

Processing time= 20sec.

Table 3.1. Estimating fitness and weight parameter of query plans

Sl.No.	Query(Plan)	Fitness value	Weight
1	118	34.875	4.4531
2	39	25	4.1156
3	119	35	4.45
4	4	20.5	2.9031
5	126	35.875	4.4625
6	37	24.75	4.0938
7	58	27.375	4.3438
8	62	27.875	437
9	105	33.25	4.442
10	82	30.375	4.4219

As reflected in Table 3.1 it is observed that the query plan is directly proportional to the fitness parameter. Also it is seen that the fitness value depends on processing time as well as position and size of query plan. The weight of the query plan also depends on the position and size of query plan as shown in the Figure 3.1.

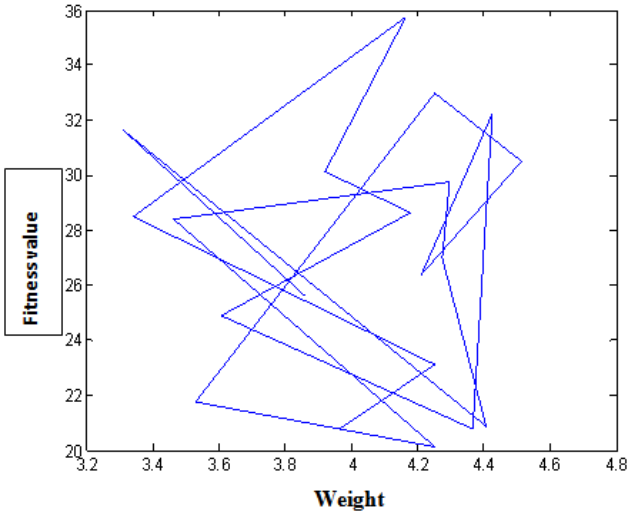

Figure 3.1. Weight VS Fitness parameter

Table 3.2. Estimating fitness and weight parameter of query plans

Sl.No.	Query(Plan)	Fitness value	Weight
1	33	87	4.0969
2	100	120.5	4.2469
3	123	132	5.0406
4	23	81.5	4.0731
5	51	96	4.1469
6	89	115	4.1313
7	76	108	4.1275
8	39	90	4.0979
9	82	111	4.1292
10	107	124	4.3447

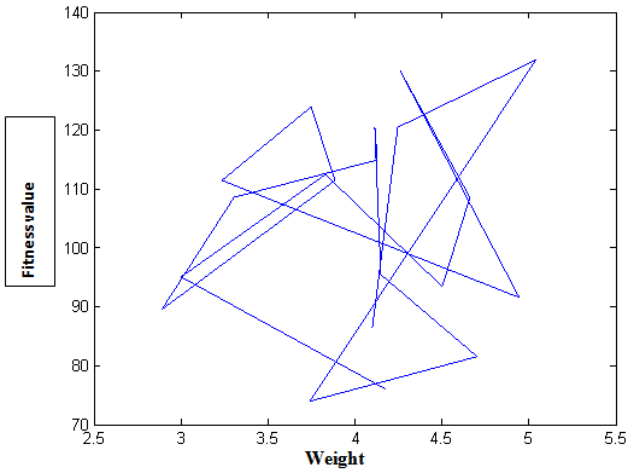

Figure 3.2. Weight VS Fitness value

After inputting the number of operations as 70, number of terminals as 90 and setting the processing time as 70sec. it has been observed as shown in Table 3.2 that the weight of query plan varies according the size of the query plans. And also with the high range of query plans, the query plans are directly proportional to the fitness parameters as shown in the Figure 3.2.

4. Discussion and Future Direction

It has been observed the lack of efficiency in handling data access queries in distributed database system. The concept of query optimization may be used to solve the above said problem.

The use of Genetic approach in this optimization process of distributed query reduces the total time involved in finding an optimal query allocation plan significantly. Also genetic algorithm compromises on the fitness value of the distributed query, e.g. the optimized query plans produced by traditional approaches may not be more optimal as compare to the query plans provided by the genetic algorithm. The application of genetic algorithm may be more essential because in complex distributed queries the traditional as well as enumeration techniques could not give query plans in feasible time.

5. Conclusion

In this research work, usually the distributed queries are optimized using genetic algorithm by implementing fitness and weight parameters. As shown in the experimental analysis it has been observed that genetic algorithm may be the right application technique for optimization of distributed queries as compared to the traditional optimization techniques, which may be because of lack of capability of providing optimized result in appropriate time.

References

[1] N. Tatbul, U. Centintemel, S. Zdonik, M. Cherniack, and M. Stonebraker. Load shedding in a data stream manager. In the 29th International Conference on Very Large Data Bases (VLDB), 2003.

[2] B. Babcock, S. Babu, M. Datar, R. Motwani, and D. Thomas. Operator scheduling in data stream systems. VLDB Journal Special Issue on Data Stream Processing, 2004.

[3] W. W. Chu and P. Hurley, "A model for optimal processing for distributed databases," in Proc. 18th IEEE COMPCON, Spring 1979, pp.116-122.

[4] E. Wong, "Retrieving dispersed data from SDD-1: A system for distributed databases," in Proc. 2nd Berkeley Workshop Distributed Data Management and Comput. Networks, May 1977, pp. 217-235.

[5] Y.E. Ioannidis and Y.C. Kang, “Randomized algorithms for optimizing large join queries, ACM 1990.

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

[7] T.V. VijayKumar, Vikram Singh and Ajay Kumar Verma, “Generating Distributed Query Processing Plans using Genetic Algorithm”, In the proceedings of the International conference on Data Storage and Data Engineering (DSDE 2010), Bangalore, February 9-10, 2010, pp. 173-177, 2010.

[8] J. H. Holland, “Adaptation in natural and artificial systems,” University of Michigan Press, 1975.

[9] M. Mitchell, “An Introduction to Genetic Algorithms,” MIT Press, 1998.

[10] William I. Grosky, Junping Sun, Farshad Fotouhi “Dynamic selectivity estimation for multidimensional queries”,springer, 1993.

[11] M.Tamer ozsu, Patric Valduriez. ”Principles of Distributed Database Systems”, springer, 2010.

[12] Surbhi Bansal, sofia gupta and Rajinder singh virk, “Selectivity Evaluation in Distributed Database Query Operations: Static vs Dynamic techniques”, IJCAIT, 2014.

[13] Rajinder Singh, Gurvinder Singh, Varinder Pannu virk” Optimized Access Strategies for a Distributed Database Design”, IJDE, 2011.

[14] Faiza Najjar and Yahya slimani. “Cardinality estimation of distributed join queries”. 2002.

[15] LARSON, P., AND YANG, H. Computing queries from derived relations. In Proceedings of Znternutionul Conference on Very Large Data Bases (Stockholm, Aug. 1985), 259-269.

[16] Carlo Dell˝ Aquilla, Ezio Lefons, Filippo Tangorra,” Analytic-based Estimation of Query Result Sizes”, 2005.

[17] B.M. Monjurul Alom, Frans Henskens and Michael Hannaford “Query Processing and Optimization in Distributed Database Systems”, IJCSNS, September 2009.

[18] Ridhi Kapoor, Dr. R. S. Virk, “Selectivity & Cost Estimates in Query Optimization in Distributed Databases”, International Journal of Enhanced Research in Management & Computer Applications, June 2013.