Genetic Algorithm Based Query Execution Plan Generation Using Join Site Mechanism in Heterogeneous Distributed Database

N. S. Gajjam[1]

Walchand Institute of Technology, Computer Science and Engineering Department, Solapur, India

Dr. Mrs. S. S. Apte[2]

Walchand Institute of Technology, Computer Science and Engineering Department, Solapur,India

Abstract

A heterogeneous distributed database system (HDDBS) is an attractive research area where query optimization plays important role. Heterogeneous Distributed Database is an integration of distribution of data with database schema. Retrieving the proper result with best query processing strategy is nothing but optimization.

In this paper, a Genetic Algorithm (GA)-based query optimizer is used to optimize Distributed Queries. In this paper we concentrated on join-site mechanism for finding different execution plans. Main aim of this work is to choose the right set of plans for queries which minimizes the total execution time. Mobile agents is used to perform specific task by migrating and executing on several hosts connected in the network.

By implementing this, we will get the optimized plan for the particular query so that when similar query comes again for execution, we will directly execute that query with optimized plan

1. Introduction

Today database come up with extra features like distribution, heterogeneity etc. Many features are there which made distributed database more popular like reliability, fault tolerance, availability, small response time etc. Distributed database is the collection of unrelated data spread over the network [6].

Database is made distributed using 2 techniques

1) Replication
2) Fragmentation

Replication is the technique in which particular table is stored on multiple sites. Fragmentation is the technique in which tables of the database is fragmented using horizontal and vertical fragmentation techniques and stored on different sites.

Heterogeneous Distributed Database is combination of Distributed Database and different database schema at different sites.

Optimization: Optimization is the process of finding most efficient way for execution of the SQL statements [1]. Generally optimized plan takes less time for executing the query.

Main goals of query optimization are to retrieve the data faster from the database, to reduce amount of wear on the hardware and allows server to work more efficiently.

Genetic Algorithm

Genetic Algorithms (GA) are direct, parallel method for global search and optimization. GA is part of the group of Evolutionary Algorithms (EA).

Main ingredients of GA are Chromosomes, Selection, Recombination and Mutation.

Selection – between all individuals in the current population are chose those, who will continue and by means of crossover and mutation will produce offspring population. At this stage elitism could be used – the best n individuals are directly transferred to the next generation.

Crossover – the individuals chosen by selection recombine with each other and new individuals will be created. Different chromosomes are created using interchanging the genes of the previous chromosomes.

Mutation – by means of random change of some of the genes, it is guaranteed that even if none of the individuals contain the necessary gene value for the extremum, it is still possible to reach the extremum

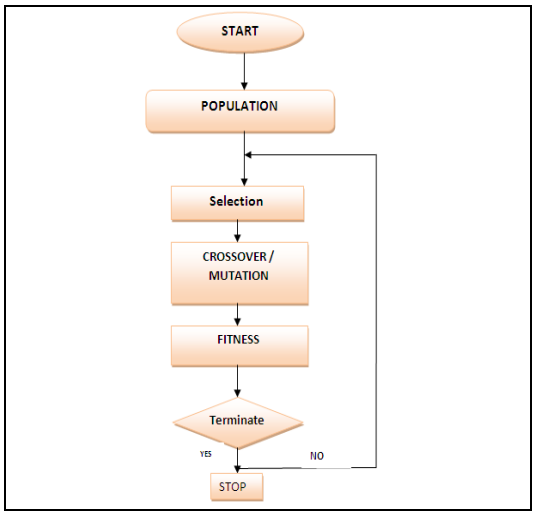

Fig 1: Genetic Algorithm

In this paper, we are presenting an approach that finds number of query execution plans for distributed query and finds best among them. Genetic Algorithm can be used for finding many execution plans [7].Query Transformation is also plays important role while executing query plan.
We use concept of mobile agent which acts as a travelling agent. Mobile agent travels in a network and performs the specific task allocated to it. Created in one execution environment, it can transport its state and code with it to another execution environment in the network, where it resumes execution [8].

2. Related Work

Murat Ali Bayir, Ismail H. Toroslu, and Ahmet Cosar[4] presented the Genetic Algorithm for Multiple Query Optimization problem. Technique used for this is to choose the right set of plans for queries which minimizes the total execution time by performing common tasks only once. This technique works in 2 phases.1st phase finds identification of common task and 2nd is to find global execution plan. They have compared Genetic Algorithm with A* heuristics technique for optimization and concluded that GA works better for Large number of queries[1].They performed their work on non-distributed database.

Ender Sevinc and Ahmet Cosar [1] presented an evolutionary Genetic Algorithm for Optimizing Distributed queries in Distributed Database. They used replicated relations, but not used horizontal fragmentation or vertical fragmentation. They compared the performance of New Genetic Algorithm with a previously defined Genetic Algorithm.

YANNIS E. IOANNIDIS [5] studied about Query Optimization and gives idea regarding optimization of a single select-project-join query in a centralized relational DBMS .But he doesn't tell the questions regarding issues like parallel, distributed, semantic, global, parametric, dynamic, nested, rule-based, object-oriented, heterogeneous, recursive, and aggregate query optimization.

Ishtiaq Ahmed, M. Rizwan Beg ,Kapil Kumar Gupta,Mohd.Isha Mansoori [2] presented a paper on A Novel Approach of Query Optimization for Genetic Population.They have given general concepts of query optimization in relational database system.They have also given implementation plan using join ordering with the use of Genetic Algorithm.

T.V. Vijay Kumar, Vikram Singh and Ajay Kumar Verma presented idea regarding Distributed Query Processing Plans Generation using Genetic Algorithm[3] .They proposed an approach which finds different query execution plans using Genetic Algorithm.These plans are based on closeness of data that is required for the query.

3. Proposed Work

There are 3 sites to store 3 database servers those are MySQL, MSSQL, and Oracle. Data is distributed on different sites. Heterogeneity is achieved by storing different fragments on different database servers.
Figure 2 represents overall structure of the system. Components of the proposed system are

Genetic Algorithm Executer(GAE)
Genetic Algorithm Executer will find out different execution plans for given query using crossover and mutation operations. Output of GAE is chromosome which specifies plan for the query.GAE has Optimized Plan Table with it consist of best execution plan for particular query.
.
Execution Unit
It takes different plans that are generated by Genetic Algorithm Executer. Main work of Execution Unit is to execute the plan and finds fitness value for it.
MySQL,MSSQL and ORACE
These are different databases stored on different sites.

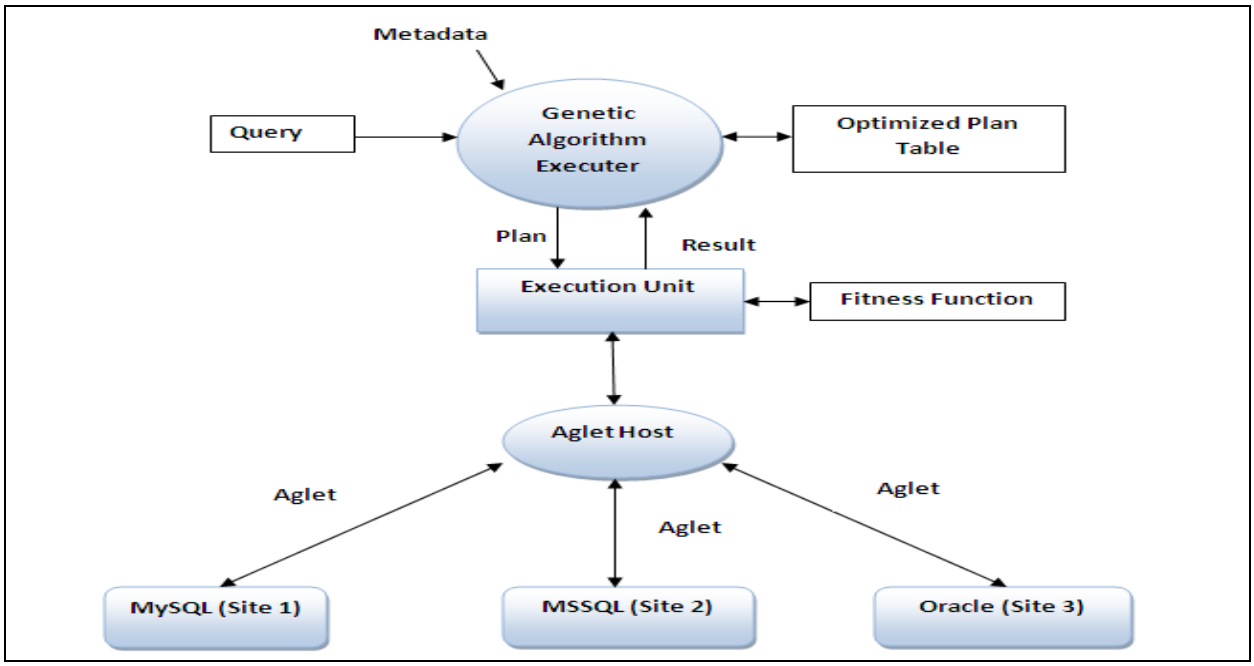

Fig 2: Overall structure of proposed system

4. Methodology

Centralized Server has metadata file with it consisting of which table is stored on sites 1, 2, 3. This information is useful for finding different execution plans. Suppose we have 4 tables(Student, Teacher, Dept, College) stored on different sites as

Student	Teacher	Dept	College
1	2	2	1
2	3	3	3
3			

User gives query consisting of joins of different tables which may store on different sites. Here centralized server having 2 main components that are Genetic Algorithm Executer and Execution Unit.
Genetic Algorithm Executer takes input from the user (query) and metadata file with it, using this GAE finds different execution plans. Suppose input query consist of student ⋈ teacher ⋈ dept ⋈ college then there are total 3*2*2*2=24 different combinations will be available for retrieving the tables. Here 3 means Student table can be retrieved from site 1 or 2 or 3.

Chromosome Structure: Each Chromosome represents different execution strategy for query. Chromosome gives idea to execution unit regarding join site. {1321} is one of the chromosomes. Here 1321 represents student table should retrieved from site1,Teacher table should retrieved from site3,Dept table should retrieved from site2,College table should retrieved from site1.

These plans are given to the Execution Unit.
Execution Unit will transform the given plan and give it to the aglet host. Aglet Host will create aglet and insert plan in it. This aglet will travel in network and retrieve result.
Fitness: Fitness value will be time required to execute the plan.
Optimized plan table consist of best execution plan for the query. This is used when similar type of query comes for execution. At that time we will direct execute that query using best execution plan.

5. Conclusion and Future Work

Main objective of this work is to get optimized plan for the query in heterogeneous and distributed environment. Here we have concentrated to retrieve the tables from site which take less time. Though it will take more time to execute at first but subsequent similar queries will be executed in best possible time.

Nature of subsequent SELECT queries is similar in Heterogeneous Distributed Database environment, then our work expecting to be more efficient.

We can improve the optimization by considering join order with join site mechanism.

6. References

[1] Ender Sevinc and Ahmet Co¸sar(2011). An Evolutionary Genetic Algorithm for Optimization of Distributed Database Queries. The Computer Journal, Vol. 54 No. 5.

[2] Ahmed I, Beg, M.R, Gupta, K.K, & Mansoori, M.I."A Novel approach of query optimization for genetic population". International Journal of Computer Sciences Issues, Vol. 9, No. 1, 2012, pp. 85-91

[3] T.V. Vijay Kumar, Vikram Singh, Ajay Kumar Verma, "Distributed Query Processing Plans Generation using Genetic Algorithm", International Journal of Computer Theory and Engineering, Vol.3, No.1, February, 2011, ISSN: 1793-8201

[4] Murat Ali Bayir, Ismail H. Toroslu, and Ahmet Cosar(2007). Genetic Algorithm for the Multiple-Query Optimization Problem. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 37, NO. 1.

[5] Y. E. Ioannidis, "Query optimization," ACM Computing Surveys, vol. 28, no. 1, pp. 121–123, 1996.

[6] S.Ceri and G. Pelagati, "Distributed Database: Principles and Systems," McGraw Hill, 1984

[7] S. Rho and S.T. March, "Optimizing distributed join queries: A genetic algorithmic approach," Annals of Operations Research, 71, pp. 199-228, 1997

[8] Danny B. Lange and Mitsuru Oshima. Seven good reasons for mobile agents.Communications of the ACM,42(3):88{89, March 1999

[9] H. Herodotou, N. Borisov, and S. Babu. Query Optimization Techniques for Partitioned Tables. In SIGMOD, 2011.

[10] M. Stillger and M. Spiliopoulou, "Genetic programming in database query optimization," in Proc First Annu. Conf. Genetic Programming,Stanford, CA, July 1996