Using Genetic Algorithm in a Machine with Parallel Structure for Optimizing Queries in the Query Graph

Javad Sohafi-Bonab, Babak nariman jahan, and Rahman haj asadollahi

Abstract**—**Optimization query is expensive and costly process, also the number of permutations for a query, growth as exponential respect to the number of relations that contribute in the query. Current techniques of optimization of a query are not suitable for supporting applications such that their database includes variety of joins of queries. In Other hand, the success key of a database manager is efficiency of his query model. In this paper, we consider the queries such that including cycle and requiring a lot of times to response to the query. We use two methods for optimizing such queries. First, by following an algorithm, we decompose graph to several trees, then distribute the trees on computers, and finally by using Genetic Algorithm, optimize it. Second, without decomposition the graph, by distributing it on computers, and then by using Genetic Algorithm, we optimize the query. After ten times executing the algorithms, we conclude first method gets better result than the second.

Keywords**—**Database, Query Optimization, Genetic Algorithm, Query graph, Parallel Machine

I. Introduction

SUCCESS mystery of a DBMS especially if we deal with relation model is a optimal efficiency of a query system.

In this system, an input is a query like q for DBMS by user.

Suppose S is a set of all possible strategies to response to query of q, each member of S likes has cost C(S).(regarding cpu & i/o). Aim of optimization of an algorithm is finding of one member of s_0 from S as[13]:

$$C(s) = \min_{s \in S} = C(s_0)$$

Now we continue discussion about relation database. One strategy for being responsible for query of q, use sequential algebra operation for database relationship to find an answer for q.

Javad Sohafi-Bonab, Deparment of computer engineering, Islamic Azad University,Bonab Branch,Bonab,Iran

Babak nariman jahan, Department of computer engineering, Islamic Azad University,Bonab Branch,Bonab,Iran

Rahman haj asadollahi, Department of computer engineering, Tabriz university,Tabriz,Iran

Cost of one strategy is total cost of process of each operator. Among these operators the difficult of them process and optimization is join operator [which is present by $\rhd\lhd$or join]This operator received two relation as input then with combination of tuples and regarding of type join produce a new relation as an output Join is a portability and associative operator.

So number of strategies for response to an query increase exponentially with number of join.[2][3]p[4][12][15].

For example although two statements (R1join R2) join R3=R1JOIN (R2JOINR3) are equal and have a same output, but time cost of them can not be same because cardinality of interrelation which should be store may not be equal in each side. for above phrase and with due attention to portability and associability of join there are twelve combination and for seven relation 665280 combination and for ten relation about 17milliard.Generally for N relation number of possible combination is equal with (2(n-1)!/(n-1)![2].Due time complexity formula, N relation is equal with $\Omega(N!)$.Because optimization of query which contain of join operator is very time consuming rather than other relation operators like select operator or project operator. Basically all optimization query algorithms related to queries with join [2][4].In modern applications of each query is usually include a few relation such as 10 relationship[12].But it is prognosticated in future application, number of relations would be more than 16. At present optimization query techniques for support some application of database are unsuitable. For example system-R is not able to performance with more than 16. join even for less than 16 join query, it encounter is with problem. [1] It sometimes encounters a proposal design to solve problem (query optimization) is to apply random algorithm such as SA(Simulated Annealing),II(Iterative Improvement)and 2PO a combination of II and SA which up to now would be successful for query optimization and provide enough reason to apply. According to experiments for small query with 5 to 10 join there is not any difference among three random algorithms.

Basically when the size of query increases so improvement of 2PO output is comparison able with SA and also with II. Even though at the similar time average cost of output strategy of all algorithms has less resistance. This present that there are many cases which these algorithms loss its best state. All in All (for all that), among these three algorithms2PO is fairly more resistance. Totally random

algorithms was acceptable for optimization and also for presented problem obtain better response rather than other method.[6][11][14][18][19]

When these Algorithms were in recursive query they will answer very well. But in case of being recursive these Algorithms spend a lot of time to response (we can imagine an equal graph for every query).In fact the problem can be put forward in this way:

When query graph be a tree graph these Algorithms are suitable approach for optimization. But if query graph be a graph these Algorithms with present structure encounter with problem. Therefore to optimization a large query which its query graph include cycle, genetic algorithm is used.

In this paper we use two methods for optimization for large queries which their query graph include a cycle. In first method we have presented an algorithm according depth search which with proposal algorithms query graph decompose to several trees and have performed every tree with application of Genetic algorithm distributed in parallel on a network of computers which connected in ring state and best solution acquired of union of one by one solutions and in second method performed whole query graph without decomposition to trees with using parallel genetic algorithm distributed. Contents of this paper organized as below:

in section (2) an algorithm is presented according deep search for decomposition of query graphs, in section (3) a parallel machine model is presented for finding optimization case of join relations, and in section (4) a parallel genetic algorithm is presented for finding optimum case, and in section (5) computing environment which program is executed on environment, and in section (6) result of experiments will expressed and finally summary and future task will be expressed.

II. An Alghorithm Base On Deep Search For Decomposition Of Query Graph

In this paper with use of depth method search which discussed in [20][9],we presented an algorithm which by that we can decompose one graph include cycle to several trees, so apply this obtained trees by mentioned method[10] for response to query. Imagine query figure (1)

$(S \triangleright\triangleleft T)$ and $(S \triangleright\triangleleft P)$ and $(P \triangleright\triangleleft H)$ and

$(H \triangleright\triangleleft T)$ and $(H \triangleright\triangleleft R)$ and $(P \triangleright\triangleleft M)$ and

$(M \triangleright\triangleleft R)$ and $(T \triangleright\triangleleft R)$

Fig 1

For each query related query graph can be obtained. In this graph nodes show relation, edges show join of two relation, and numbers on edges show types of relation of two join.

Figure (2) show query graph(1).

Considering fig(2),it will be clear that this query graph include cycle ,so response time to this query in comparison with those queries which graph of them is a tree, will be long.[14]

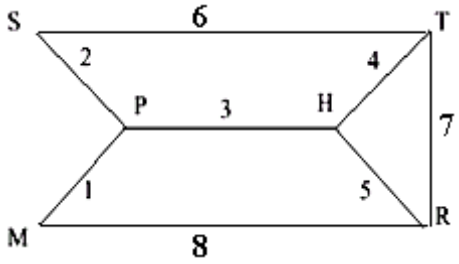


Fig. 2 Query graph

To solve this "query graph" we decompose query, that is, we will find a tree with elimination of existing cycle of graph. Note that after decomposition of graph some obtained trees may have cycle. (be a graph)

In this case we will continue above process to receive a jungle at the end.

Presented algorithm to this problem emanate from a depth search algorithm. Depth search algorithm is used to measurement of a graph. In fact this algorithm doesn't consider met nodes. So out put of this algorithm will be a tree.

Appointed algorithm will act in this way:

We made a list of joins from its near nodes for every vertex. Every node of this join list include of a vertex and a join which show type of join.

We make array of above join list called graph (number of vertex) which include this join lists. In this algorithm, first a vertex like, v, is met and enter the weight, if weight of this vertex have not entered in a tree which call tree. Then consider vertex (w) near to this vertex, if this vertex don't be met already we meet it and enter its weight in related array. Otherwise eliminate it from stack and continue this process till stack be empty. At end all vertexes is met and a list called "tree" will obtained.

```
Tdfs( integer v )
{Node_ptr w ;
Visited [ v ] =TRUE ;
print( v ) ;
if ( ! Search_Weight ( w->join , tree )
tree [I++] = w->join ;
for ( w = gragh [ v ] ; w ; w = w-> next )
if ( ! visited [ w->vertex ] )
Tdfs ( w->vertex ) ; }
```

Output vertexes by this algorithm can be match to each other by join numbers between vertexes. The result will be a tree. If acquire remain numbers which there aren't in array tree and match them to each other considering joined numbers, we succeed to decompose graph to several trees.

We should note that obtained trees have not cycle otherwise continue above process again till receive to a jungle. Now survey query (1) with this algorithm. Considering fig (1) ,proximity list of query graph will be like fig3:

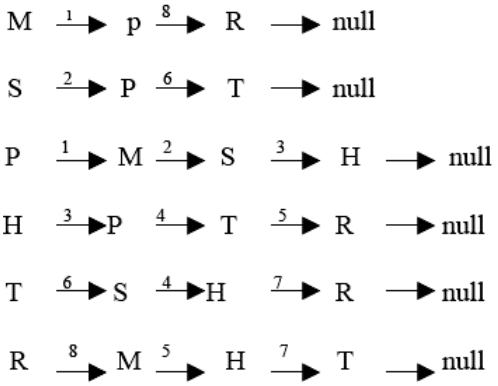


Fig. 3 Proximity list

Fig (3) is presentation of link list of query graph(1).Applying Tdfs algorithms for query graph, array Tree will take following numbers:

1	2	6	3	8

and met vertexes will be as below:

M ,P ,S ,T ,H ,R

Considering content of "array tree" and met vertexes can make query tree that has come in the fig 4:

Can be seen that cycle of query graph of fig (1) is eliminated and the result is a tree.

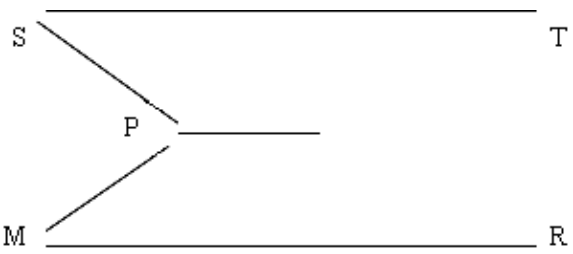
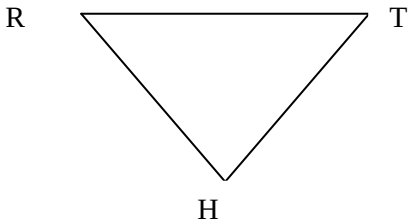


Fig. 4 The resulting query graph

Now if consider remain weights of tree then we will have:

4 ،5 ،7

Obtained graph or tree is like:



With continuing of above process, graph will decompose to two trees. So according to fig (2) we could decompose query (1) to three tree.

III. Studying Of Probable Problems

It is possible with increase of number of query relations number of obtained trees would be more. So below solution is advised:

Obtained trees which make jungle except main tree could change to one or some larger tree (max 16 join).

We can do this task by below method:

If there wasn't any join between two relations with temporary label of zero connect two trees. This mechanism will be repeated till number of joins achieved to 16.

For remain trees above mechanism will be repeated.

IV. Parallel Machine Model

Since proposal genetic algorithm will performed in network of processors, first we deal with structure which algorithm have done on it and we have concluded acceptable results. We suggest a parallel distributed execute, of parallel algorithms in multi- processors which every processors has its own local memory (not shared memory) and each processors do a genetic operation on set of its chromosome and relation with neighbors is minimum. Circumstances of distributed parallel execution is completely described. In [5][7][8][18].

To demonstrate mentioned method we use of eight computers which are connected in ring state to optimization queries with cycle in their queries in computer network. Advantage of using network with ring topology against structures with mesh on or hypercube in parallel distributed execute from genetic algorithm is completely described in [5].

In this part we turn to parallel distributed detail on multi processor system.

Since distinct execution of genetic algorithm on processors is not effective so we use transfer of chromosome (which we call migration) between neighbor processors. Two processors are neighbors if there was a direct link. In order select a chromosome for migration to distinct neighbor before producing next generation and after genetic operation like crossover and mutation , we use of two method of selection of high qualities chromosome, and random way of selection to migrate. In this method before producing next generation of processor, select a chromosome by mentioned selection method and migrate to distinct neighbor. If quality of this sent chromosome was better than neighbor processors chromosome, accept it and otherwise reject it.

Also of next generation selection method in proposal algorithm is done by using SUS method. Reason for selection of SUS method to select next generation is possibility of select of first chromosome in first population which avoid rapid convergence to solution of problem. In fact algorithm engages less in local optimum solutions.

V.Application Of GA In Parallel

At the beginning of execution, each processor produce set of its chromosome based on GA parameters (mutation and crossover) which is sent from host processor then each processor operate genetic operation like crossover, leaping on its own population in each generation similar to ordinary genetic algorithm [17] . Before producing next of generation or action selection, moving chromosome are done by

neighboring processors, if contain the condition of displacement. That we call migration condition.

We introduce migration condition which each processor send high quality or random chromosome to its neighbor and if quality of received chromosome was better than quality of its chromosome.It will accept it and or not it will reject chromosome. Procedure for each processor is like below:

Procedure PGA;
begin
Receive GA parameters and a signal for
starting execution from the host processor;
Initialize a population;
repeat
Evaluate all the chromosomes;
If the migration condition holds **then**
Take place the chromosomes migration;
endif;
Select the chromosomes for the next generation
by the SUS way with the elite strategy or random;
Create new chromosomes by applying
Crossover rate the mutation rate;
until the termination condition holds;
Send result to the host processor;
end;

VI. Applied Computation Envoironment

All computation in a network environment with Linux operating system is described in table (1). Proposal genetic algorithm is implemented by using c++ and v2.0 is used as a interface to transfer messages and communication between genetic algorithms between different computers.

TABLE I
APPLIED COMPUTATATIONAL ENVIRONMENT FOR THE IMPLEMENTATION OF THE ALGORITHM

Computing Environment Component	Description
Number of Nodes Used	8
Processor Type and core Speed	Intel Pentium 3 @ 600 MHz
Network Type and Bandwidth	Ethernet 100Mbps
Network Protocol/Topology	TCP/IP /Ring
OS Kernel Type and Version	Linux 8.0 (2.4.20-19.8)
MPI Type and Version	LAM 6.5.6 / MPI 2
Compiler Type and Version	mpiCC and gcc (3.2)

VII. Result Of Experments

Before of expressing result of experiments we turn to some applied expressions and needed parameters to execute genetic algorithms.

In migration step, some chromosome migrates from a sub population to neighbor sub population. We migration frequency by M_{freq} which described as a number of generation. For example if $M_{freq}=4$ chromosome will migrate to neighbor one time every four generation. Also we show number of migration by M_{rate}. For example if it was 5 it shows in each migration five chromosome are migrated we also use selection SUS method to select chromosome for next generation and use of two method of best selection of chromosome or random selection to choice a chromosome to transfer to neighbor. Also genetic algorithm spam show number of produced generation and $N_{relation}$ use for number of applied relations to optimization. Fix applied parameters experiment is shown below:

- Generation span = 100
- Crossover rate = 0.8
- Mutation rate = 0.01
- $M_{rate} = 1$
- Select method = SUS
- Migrated chromosome is a best fitness chromosome or random
- $N_{relation} = 8$

We have supposed below parameters to better evaluate of found solutions variable:

- Number of chromosome per sub population = 3 , 6 , 8
- $M_{freq} = 1 , 4$

As it is already mentioned we use two methods to optimization query for query graph. Each method is described separately and we turn to result of them at below.

In first method query graph is decomposed with using Tdfs algorithm to several tree and each tree is solved by mentioned parameters with using PGA algorithm. Finally after execution of all trees final results obtained of union of answers. For example one method to select parameters of variable genetic algorithms to execute can be as below:

- Number of chromosome per sub population = 3
- $M_{freq} = 4$

This means size of primary population in each processor 3, and every four generation selection of chromosome to migrate is done. Obtained result of execution of first method is shown in table 2, 3. In second method query graph execute without decomposition to tree with using mentioned parameters in first method and results shown in table 4,5.

By comparison of obtained result we can understand optimal result is obtained first method with four chromosome parameter in each sub population and $M_{freq}=4$.

VIII. Conclusion And Future Works

Because time of execution of query graph with cycle is longer than other relation query graph we turn to solve query graph on network with using two proposal methods and applying distributed parallel genetic algorithm with different parameters. In first method we present an algorithm that we decompose graph with cycle and change it to different tree

and then distribute every tree separately in executing processors and each processors with using genetic algorithm which is called PGA we would be able to find optimal case. Final results obtained at the end from union of results. In second method without decomposition of query graph to tree we solve it with using genetic algorithm. Also applied parameters in distributed parallel genetic algorithm is divided to two fix and variable part, fix parameters in all execution of two mentioned method in paper are fix and presented result in this paper is based on variable parameters for example migration rate and number of chromosome in primary population.

We execute to presented method in this paper ten times and separately. In first time we suppose migration frequency of chromosome one and execute genetic algorithm with using number of existing chromosome in primary population. Second time migration frequency chromosome is supposed four and execute genetic algorithm as before from obtained result and comparing them we found out that in first number of chromosome in primary population, most primary solutions in obtained and time of finding results is very worthy in comparison with ordinary genetic algorithm .

In future works we will try select chromosome to transfer to neighbor by array linked and tournament selection method and verify its effect on founded solution..

TABLE II
NUMBER OF FOUND SOLUTION FOR THE METHOD OF WITHOUT DECOMPOSITION QUERY GRAPH TO TREE

AVG	RUN9	RUN8	RUN7	RUN6	RUN5	RUN4	RUN3	RUN2	RUN1	RUN0	#sub pop
2287.2	2658	2457	2036	1989	2785	2217	1957	2652	2014	2134	3
2312.4	2123	2583	2042	2145	2004	2954	2471	2652	2145	1996	4
2056	1891	2486	1652	1980	2104	2753	1932	2037	1584	2141	8

TABLE III
NUMBER OF FOUND SOLUTION FOR THE METHOD OF WITHOUT DECOMPOSITION QUERY GRAPH TO TREE

AVG	RUN9	RUN8	RUN7	RUN6	RUN5	RUN4	RUN3	RUN2	RUN1	RUN0	#sub pop
2432.2	2984	2387	2057	2139	2785	2217	2853	2541	2225	2134	3
2699.4	3047	2883	2143	2314	2594	2957	2987	2647	2965	2457	4
2570	2994	2967	2035	2019	2636	2753	2876	2738	2541	2141	8

TABLE IV
NUMBER OF FOUND SOLUTION FOR THE METHOD OF WITHOUT DECOMPOSITION QUERY GRAPH TO TREE

AVG	RUN9	RUN8	RUN7	RUN6	RUN5	RUN4	RUN3	RUN2	RUN1	RUN0	#sub pop
183	206	204	159	147	98	145	258	180	243	190	3
501	281	286	212	207	236	264	284	247	230	203	4
488.5	512	542	465	500	498	563	502	498	398	407	8

TABLEV
NUMBER OF FOUND SOLUTION FOR THE METHOD OF WITHOUT DECOMPOSITION QUERY GRAPH TO TREE

AVG	RUN9	RUN8	RUN7	RUN6	RUN5	RUN4	RUN3	RUN2	RUN1	RUN0	#sub pop
119.6	114	107	146	151	101	92	141	123	123	98	3
189.1	296	201	195	203	163	196	175	178	142	141	4
572.6	580	601	578	599	587	537	612	596	532	504	8

REFERENCES

[1] Ozsu . m , Valduriez . *P, Principle of Distributed Database System*,2ed , Prentice Hall,USA,1999
[2] Silberchatz,Database System Concepts,3ed,WCB/Mc GRAW Hill,USA,2008
[3] Garcia, Molina , Database System Implemantation, Prentice Hall,NJ,2000
[4] Chaudhuri.s ,"An Overview of Query Optimization in Relational Systems"
[5] Y.Takahashi, "Parallel Processing Mechanism",Maruzen ,1990 in Japanese
[6] L.Davise,"Genetic Algorithm and Simulated Annealing",Morgan Kaufman Publishers
[7] Culler . D , Single.J, Parallel Computer Architecture:A Hardware/Software Approach,Mogan Kauman,San Francisco,USA,2009
[8] Desrochers.G, Principles of Parallel and Multiprocessing,McGRAW-Hill CO,New York,USA
[9] Aho.A , Ullman.J, The Design and Analysis of Computer Algorithms, Addison-Wesley

[10] Tanha, J., "Using the Genetic Algorithm for Optimization of Queries in Distributed Databases", Master 's Degree Thesis, Amirkabir University, March 1381
[11] Y. Kang . Randomized Algorithms for Query Optimization. PhD thesis, University of Wiconsin, Madison, May 1991 .
[12] Y.E.Ioannidis Query Optimization,supported by Oracle ,IBM, Research 1996
[13] Lanzelotter .R ,"On the Effectiventees of Optimization Search Strategies for Parallel Executive Space",Proceding of VLDB Cnference,1993
[14] R.Sterrite,"A Parallel Genetic Algorithm for Cause and Effect Network", Proceedings of the IASTED International Conference on Artificial Intelligence and Soft computing ,pp.105-108,1997
[15] S. Nahar, S.Sahni and E.Shragowitz . Simulated Annealing and Combinatorial Optimization ,in Proceedings of the 23rd Design Automation Conference , 1986 ,pp 293-299
[16] Y.E.Ioannidis and E.Wong , Query Optimization By Simulated Anneling, in Proceedings of the 1987 ACM-SIGMOD Conference, San Francisco, Ca , June 1987 , pp 9-22.
[17] Tarjan.R, "Depth-first search and linear graph algorithm", SIAM Journal of Computing