Optimizing Access Strategies for a Distributed Database Design using Genetic Fragmentation

Rajinder Singh Virk[1] **Dr.Gurvinder Singh**[1]

[1]As.Prof ,Dept. Of Computer Science & Engineering, Guru Nanak Dev University Amritsar.

Summary
A key component of any Relational Distributed Database Query Optimizer is to fragment various tables and distribute fragmented Data over the sites of network. Then find an near optimal or best possible subquery operation allocation plan in a stipulated time period. In this paper we have proposed a Genetic Algorithm (GA) for finding near optimal fragmentation plan for selecting the various nodes or sites for placing recursively the vertically fragmented data attributes in two components for a Query Transaction on the Database. We discuss advantages of using proposed Genetic Algorithm (PGA) over various other prevalent Algorithms and un partitioned case. Experimental results for a simulated Distributed Database over a Wide Area Network show encouraging results for the use of PGA over other techniques.

Keywords
Distributed Database, Query Optimization, Database Fragmentation, Access Strategies, Genetic Algorithms.

1. Introduction

Two critical components of a Distributed Database Query Optimization Model are i) Determining the database fragmentation scheme and ii) Determining the subquery allocation plan. Most of research has contributed to second aspect i.e. on finding the operation allocation plan and order of execution of various subqueries of a bigger transaction query. In this paper emphasis is on exploring the former aspect i.e. fragmentation of various attributes vertically and recursively, using an innovative Genetic Algorithm(PGA). Queries can be categorised mainly in two groups first OLTP (Online Transaction Processing) and second as DSS(Decision Support System)[1].OLTP Queries are generally repetitive in nature. So Optimizer should seek good thru put and should emphasis on reducing the total cost of query. On the other hand DSS queries are not so touchy about high thru put but seek response time minimization . The proposed Genetic Algorithm 'PGA' works to minimize two important cost components wiz Local Processing Cost and Total Communication Cost for processing whole of query. It starts by generating an initial pool of solution by random generation of integer sequences which corresponds to partitioning a relation into two or more components. It corresponds to first Population. Further Populations are generated using standard principles of GA's i.e. by applying selection, crossover & mutation[2]. Objective Function is based on Total Disk Accesses Minimization for different access path strategies. Selection takes care that better the Objective Function (Cost Minimization) fitter is a the chromosome and gets more number of probabilistic chances of reproducing offspring in next population. Crossover is used to let offspring share features of both parents and possibly improve over fitness value. Mutation is given a very meagre chance so that some important features of parent population are not lost by crossover. Reproduction is stopped when predefined maximum number of generations is reached or any further populations show no fitness improvement for a long number of generations.

2. Related Research Work

Distributed database systems design and query optimization has been and will remain an active area of research for a lot times to come, due to complex and intractable nature of the problem[10-17].Most of the work has concentrated on two aspects: Fragmentation & Data Allocation(The plan of allocating Fragments to various sites) and Operation Allocation(How to generate a sequence of subqueries on various sites). Apers and P.M have discussed in detail the data allocation problem and their fragmentation in [18]. An integrated solution to problems of Data Fragmentation, allocation, replication in Distributed Databases, has been proposed in Tamhankar & Ram[19].

Simulation experiments for comparison of Branch & Bound, Simulated Annealing, Greedy approaches for operation allocation problem have been describes in detail by Martin & Lam[20].

Frieder and Baru [21] propose dynamic site selection strategies for distributed database design on a microcomputer. March & Rho in [10] have proposed an excellent cost model for reducing local I/O costs, CPU Costs and Communication Costs in operation allocation strategy. Johansonn & Noumann in [22] extended their work by considering parallel processing and Load Balancing in Data and Operation Allocation.

Manuscript received June 5, 2011
Manuscript revised June 20, 2011

Many different algorithms have been proposed for various aspects of Query Optimization, few e.g. fragmentation, data and operation allocation etc , may be divided into three major categories
1.Deterministic Search Algorithms
2.Randomized Algorithms
3.Genetic Algorithms
The most prevalent technique in first category is Dynamic Programming, as used in System R[3].The Algorithm exhaustively searches thru all plans for the query and principle of optimality prunes away bad sub-plans as early as possible[4].Algorithm is suitable for a Database Design involving lesser no. of fragments and joins as it guarantees the optimal solution, but it is not time efficient for large no. of joins and relations as its exponential in nature[5].Many variations of this classical version have come, most of which involve heuristic pruning [6,7].
In second category Randomized Algorithms make random choices as they walk thru the state space to find a local minima. Simulated Annealing (SA) was originally developed to model the annealing process of crystals. Difference from Iterative Improvement is that it can accept uphill moves with probability determined by a variable called Temperature as in Annealing process.
The most successful of these Algorithms called 2PO (Two Phase Optimization) combines iterative improvement (a variant of hill climbing) with Simulated Annealing[6].This approach also becomes exponentially interactive soon with increase in number of relation, joins and sites[4].
The Algorithm presented in this paper (PGA) falls under the category of Genetic Algorithms. These algorithms simulate natural evolution process. These encode a potential solution to a specific problem on a simple chromosome (Encoded String of solution data) like structure.
These strings are evaluated for Fitness (Objective Function) and relatively more fit members get more chance to reproduce their features to next population.
Fitter chromosomes are propagated to next population until a predetermined no of population generations are produced or no further improvement in solution is taking place [2].

3. Genetic Fragmentation Model

3.1Assumptions and experimental setup:

We consider retrieval transactions over a relation consisting of selections, projections and join operations. Relation attributes are classified as Clustering(Primary Key), Non-Clustering(Optional Key) and Non Key ones. Restrict attribute selects according to selection predicate, Scan attribute is the attribute which actually scans the relation.

It is assumed that following database transaction statistics are available apriori
(i) Transaction Frequency
(ii) Set of attributes needed for the transaction along with their selectivities
(iii) Set of restrict attributes and indexes.
Different access paths are evaluated for different disk accesses required for the transaction and multiplied by its frequency.
Initially a Relation is partitioned into two fragments labelled as Primary Fragment and Secondary Fragment. The total no of disk accesses are equal to the sum of disk accesses required for the Primary and Secondary fragment components. We use formulation of A.F,Cardenas[22], S.B.Yao[23] for estimating the disk accesses required a shown in following equations.

3.2 Estimated Disk Accesses for Primary Fragment :

a) For Clustered Index Scan:
Disk Access Cost is given by
Dci= SsaCrTp/B (3.x1)
where
S_{sa} : Selectivity of scan attribute
C_r : Cardinality of the relation
T^p : Tuple Size of Primary Fragment
B : Block Size (4k for a Page)

b) For unclustered index
Number of Disk accesses are estimates as
$$D_{ui} = P_{pf}(1-(1-1/P_{pf})^T + S_{sa}N_{op} \quad (3.X2)$$
$P_{pf} = C_rT^p/B$
$T = S_{sa}\,C_r$
P_{pf} = Primary Fragment Page Size
T = No. Of tuples after applying selection Predicate
S_{sa} : Selectivity of scan attribute
C_r : Cardinality of the relation
T^p : Tuple Size of Primary Fragment
B : Block Size (4k for a Page)

c) For Sequential Scan
$$D_{ss} = C_rT^p / B * F_b \quad (3.X3)$$
F_b= Prefetch Blocking Factor
For different access strategies we look for minimum cost out of { D_{ci} ,D_{ui}, D_{ss}}.

3.3 Access costs for secondary Fragment

a) Sequential Scan
Using tuple identifiers Estimated disk accesses are given by
$$D_{2is} = P_{pf}(1-(1-1/P_{pf})^T \quad (3.X4)$$

$P_{pf} = C_r T^S / B$

$T = P_s C_r$

Where Ps is selectivity from primary segment due to multiple restrict attributes and is given by

$P_s = S_{sa} \prod_{i=2}^{k} Si$

Si is the selectivity of attribute i of primary fragment or 1 otherwise.

T^S is tuple size of secondary fragment

Using sequential scan

$$D_{2ss} = C_r T^S / B * F_b \qquad (3.X5)$$

Disk Access Costs are taken from minimum of $\{D_{2is}, D_{2ss}\}$, same is true for unclustered index scan.

b) Clustered Index Scan

b) Clustered index scan

Estimated disk accesses are given by

$$D_{2ci} = S_{sa} C_r T^P / B \qquad (3.X7)$$

4. Genetic Algorithm for selection of Access Strategies (PGA):

Input: Transaction Profile
output: Different Fragmentation schemes along with their costs
1.Intialize gene with randomly generated binary strings to initialize N initial Parent Chromosomes.
Repeat steps 2-6 until No_of_Gen's = M (Max_no_of_gens) ,or previous five generations have not improved any further on Avgfit.
2.Read Input Data File containing various transactions and their frequencies, attributes required and their selectivities.
3.Read clustering attribute, restrict attributes and indexes available.
4.Calculate fitness of each chromosome by calculating the Objective Function by estimating disk accesses for each case of access strategy, as given above by set of equations 3.X and choosing one access strategy which minimises the disk access costs costs.
3.MaxFit = Maximum Fitness of a chromosome from present generation, Avgfit=Average fitness of the generation.
4 Elitism: Pass the MaxFit Chromosome to next generation.
5 Generate the mating pool with various chromosomes entering it with copies proportional to the fitness .Exchange parts of chromosomes with Crossover Operator with probability=7/10 and apply Mutation Operator with probability=2/100.
6Generate the remaining N-1 chromosomes of the new population. Generate the population Report .
7.Print Final_Fragmentation_scheme_table .

5. Experimental Results:

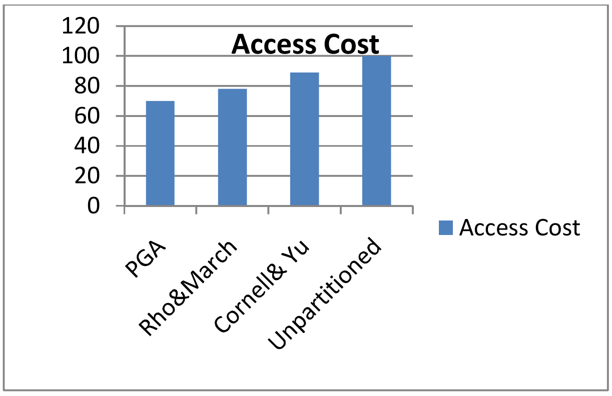
Algorithm was coded into a Genetic Program in PASCAL Language by simulating a Distributed Database Environment. Experiments were conducted on a PC Pentium Core 2 CPU 2.13GHz with 1 GB RAM on WINDOWS-XP Platform.

Evaluation of the GA is done by taking the classic example problem of Cornell & Yu[25]. A transaction profile for this is shown below

Transaction #	***Attributes sizes***										***Rest. attribute***	***Selectivity***
	10	8	4	6	15	14	3	5	9	12		
	Attributes											
	1	2	3	4	5	6	7	8	9	10		
1	1	0	0	0	1	0	1	0	0	0	1	.0025
2	0	1	1	0	0	0	0	1	1	0	2	.005
3	0	0	0	1	0	1	0	0	0	1	4	.0025
4	0	1	0	0	0	0	1	1	0	0	7	.0035
5	1	1	1	0	1	0	1	1	1	0	1	.0025
6	1	0	0	0	1	0	0	0	0	0	1	.0025
7	0	0	1	1	0	0	0	0	1	0	4	.0025
8	0	0	1	1	0	1	0	0	1	1	3	.0015

Comparing results with that of Cornell and Yu[25] and Rho & March[14] and unpartitioned one. we find PGA showing 30% improvement over unpatitioned technique . It also outscores other two techniques by 8% an 19% of the disk access costs respectively.

SOLUTION CATAGORY	***DISK ACCESS COSTS***	***SOLUTION***
PGA	70	{1 2 5 7 8} {3 4 6 9 10}
Rho & March	78	{1 2 3 5 7} {3 4 6 8 9 10}
Cornell & Yu	89	{4,6,10} {1 23 5 7 8 9}
Unpartitioned	100	{1 2 3 4 5 6 7 8 9 10 }



6. Conclusion:

The goal of this research paper is to present a probabilistic Genetic Solution to the fragmentation and access strategies problem of a Distributed Database Design. Significant improvements over other prevalent methods have been highlighted as shown by Results table and graphs above.

References:

[1] Bergsten, B., Couprie, M. & Valduriez 1993 Overview of Parallel Architectures for Database .The Computer Journal,36,734-740.

[2] Goldberg David.E"Genetic Algorithm in Search, Optimization & Learning "Pearson Education 2nd Edition.

[3] Patricia G.Selinger,Mortan M.Astrahan,Thomas G. Price, Access Path Selection in a Relational Database Management System.In proceedings of the ACM SIGMOD International Conference on Management Of Data ,1979.

[4] Amol V Deshpande and Joseph M.'Decoupled Query Optimization for Fedrated Database Systems' Report No.UCB/CSD-01-1140,2001, Berkley

[5] Kiyoshi Ono, Sema Nural,Asuman Dogac . Measuring the complexity of join enumeration in query optimization. In 16th International Conference on Very Large Data Bases,1990.

[6] Yannis E Ioannidis and Youngkyung Cha Kang. Randomized Algorithm for Optimizing Large Join Queries. Procedings of ACM SIGMOD International Conference on Management Of Data ,1990.

[7] Michael Steinbrunn, Guido Morekotte, Alfons Kemper. Heusristic and Randomized Optimization for the join ordering problem.

[8] www.tpc.org

[9] Ozsu & Valduriez. "Principles of Distributed Database Systems" Pearson Education 2nd Edition,pp. 228-298.

[10] March,Rho,"Characterisation and Analysis of a Nested Genetic Algorithm for Distributed Database Design".,Seoul Journal of Business pp 85-121 vol2,Number 1. 1995.

[11] Sacco,G. & Yao"Query Optimisation in Distributed Database Systems"1982,Advances in Computers,21,225-53.

[12] Yu,C.T,Chang" Distributed Query Processing " ACM Computing Surveys,16,399-433.

[13] Graefe,G"Query Evalution Texhniques for a large Database" ACM Computing Surveys,25,73-90, .1993.

[14] March,S.T.,Rho "Allocating Data and Operations to nodes in a distributed database design". IEEE Trans. On knowledge and Data Engg.,7(2). 1995.

[15] Kossman,D. "The state of the art in Distributed Query Processing".,ACM Computing Surveys.,32(4),422-469. 2000.

[16] Cheng,C.H.Lee,W-K,Wong,K-F, "A Genetic Agorithm based clustering approach for database partitioning " IEEE Transactions on System,Man,Cybernetics,32(3),215-230. 2002.

[17] Narasimhaiah Gorla,Suk-Kyu Song " Subquery Allocation in Distributed Database Using Genetic Algoritms",JCS&T (Jounal of Computer Science & Tech.) VOL10.No.1.April 2010.

[18] Apers,P.M.G,1988"Data Allocation in Distributed Database Systems",ACM Trans. On Database Syatems,.13(3),263-304

[19] Tamhankar,A.M & Ram"Database Fragmentation & Allocation: An Integrated Methodolgy and case study." IEEE Transactions on System,Man,Cybernetics,28(3),288-305.

[20] Martin,T,Lam& Russel"An Evaluation of Site Selection Algorithms for Distributed Query Processing"'The Compuer Journal,33(1),61-70,1990.

[21] Frieder, O. Baru"Site and Quey Sechduling policies in Microcomputer Database Systems" IEEE Trans. On knowledge and Data Engg.,6(4).1994.

[22] Johansson,JM,March,ST,Naumann"Modelling Network Latency Paralell Processing In Distributed Database design",Decision Sciences,34(4) 677-706 2003.

[23] Cardenas,A.F" Analysis and Performance of Inverted DataBase Structures",ACM Transactions may-1975,253-63.

[24] Yao,S.B,"Approximating Block Accesses in DataBase Organization" ACM Communications April1977.

[25] Cornell, Yu ,"An Effective approach to vertical partitioning for physical Design of Relational Databases." IEEE Transactions on Software Engineering 16(Feb,1990)