Query Optimization of Distributed Database Based on Parallel Genetic Algorithm and Max-Min Ant System

Wenjiao Ban, Jiming Lin, Jichao Tong, Shiwen Li
College of Information and Communication,
Guilin University of Electronic Technology,
Guilin 541004, China
Email:18607832467@163.com

Abstract— **Since the era of big data is coming, the first important problem is how to enhance the speed of database query. For the query optimization of distributed database, the speed of query depends on the data transfer and order of join. The cost model minimizing communication cost is the emphasis of research. Parallel Genetic Algorithm-Max-Min Ant System was proposed to seek a best query execution plan, which combines faster convergence of Genetic Algorithm, globally search ability of Max-Min Ant System and parallel property of both them. The experiment results show that the proposed algorithm is effective for query processing of multi-join, and plays important role in improving the performance of distributed database.**

Keywords—distributed database; Genetic Algorithm; Max-Min Ant System; query execution plan; parallel

I. INTRODUCTION

Query optimization is one of the most important and expensive stages in distributed database query. As the most fundamental and common operator, multi-join query optimization is a difficulty of query optimization. Query optimization aims to search an optimal Query Execution Plan (QEP) that can return query result to the user quickly and efficiently. Distribution and redundancy characteristics of distributed database require the influence of the factors, such as network model, data allocation, transport cost and so on should be considered comprehensively when generating an optimal QEP [1]. In fact, the choice for optimal QEP which involves multiple relations is a NP problem. So often, it's necessary to rely on simple hypothesis of processing conditions to choose a relatively optimal QEP.

Until now, many algorithms have been applied to solve the query optimization problem of distributed database. The dynamic programming algorithm as a kind of approximate exhaustive search algorithm has huge computation problem when the number of relations increases [2]. Iterative improvement algorithm and simulated annealing algorithm are two kinds of random search algorithm [3], compared with the dynamic programming algorithm, although these two algorithms reduce the search cost, but do not guarantee the optimal solution. In [4] and [5], the method based on Genetic Algorithm (GA) was proposed to generate distributed QEP. In [6], Ant Colony Algorithm was used to solve this problem.

The present study proposed a query optimization algorithm that combines Genetic Algorithm (GA) and Max-Min Ant System (MMAS) to improve query efficiency. Firstly, use the fast convergence characteristics of GA to take a set of relatively optimal QEPs. Then transform them into the initial pheromone of MMAS, which guide ants to converge faster to find the optimal QEP. Meanwhile process the hybrid algorithm in parallel to improve the solving speed further. Practice has confirmed the truth of our improvement.

II. THE QUERY COST MODEL OF DISTRIBUTED DATABASE

Uncertainty of Join order of the relations cause the QEP diversity, the query optimization use a good algorithm to determine the QEP with optimal join order, achieve the goal of efficient query. Usually, for a join involves n relations, its query execution strategies can be divided into three types of query connection tree roughly [7], respectively are left-deep tree, bushy tree and right-deep tree. Generally, with n relations, it may create $\binom{2(n-1)}{n-1}\frac{(n-1)!}{2^{n-1}}$ query tree which include $n!$ left-deep trees [8]. [6] has shown that in most cases, the optimal solution is often appear in the left-deep tree, so this paper select left-deep tree as the search strategy.

Usually, choose QEP in distributed database according to the size of query relations, in addition, the size of the intermediate results of two join relations also can be used as cost estimation., For a join J=R_1 join R_2 in distributed database, it is necessary to consider its computational cost in two different cases. When two join relations are in the same site, just calculate join operation cost; but when they are not in the same site, transport cost should be considered additionally. In conclusion cost estimation model is given as follows:

$$cost(J)=\begin{cases}\dfrac{|R_1|\times|R_2|}{\prod_{c_i\in C}\max(V(c_i,R_1),V(c_i,R_2))} & R_1 \text{ and } R_2 \text{ are in the same site}\\ \dfrac{|R_1|\times|R_2|}{\prod_{c_i\in C}\max(V(c_i,R_1),V(c_i,R_2))}+\min(|R_1|,|R_2|) & \text{other case}\end{cases} \quad (1)$$

978-1-4673-9587-8/15 $31.00 © 2015 IEEE
DOI 10.1109/ISCID.2015.199

In (1), |R| is the size of relation R, C stands for public attribute set of relations R_1 and R_2, V (c, R) is numbers of the distinct values of attribute c in R.

In a join that contains n relations, $(j_1, j_2, ..., j_i)$ are intermediate nodes of join tree, the cost estimation model is:

$$COST = \sum_{i=1}^{n-1} \cos t(j_i) \tag{2}$$

III. Parallel Genetic Algorithm-Max-Min Ant System and It's Application In The Distributed Query Optimization

Although genetic algorithm convergence faster than ant colony algorithm, it is easier to fall into an early-maturing state. Relatively, as an enhanced ant colony algorithm, Max-Min ant system has stronger global search ability, but the lack of initial pheromone lead to slower convergence speed in the early stage of generating an optimal solution. So in order to overcome the shortcomings of above two algorithms and give full play to their advantage, it is entirely feasible to combine them together. Besides, considering that two algorithms can run in parallel and distributed database provide parallel environment, the parallelization technique can be applied to improve the hybrid algorithm further. For convenience, we call this new algorithm PGA-MMAS which is the abbreviation of parallel genetic algorithm and Max-Min ant system. Application of PGA-MMAS algorithm in distributed database query optimization is showed as follows.

A. Genetic Algorithm module

The core content of genetic algorithms including problem abstraction coding, initial population, fitness function design, genetic operation design and control parameters setting.

(1)Coding

According to the characteristics of distributed multi-join query, string structured coding was used in this paper. Chromosome is divided into two parts: the coding part of relations' join order and the coding part of relations' size and site. In the relations' join order code segment, implementing the postorder traversal to final join relations to transform query tree into string structured coding. Join relations represented by leaf nodes are in the form of the integers 1, 2, 3. In the n relations' join order code segment, length of code is the number of relations. And relations' size and site code segment is a n*2 two-dimensional matrix, column 1 and column 2 of the i*th* row of matrix correspond the i*th* relations' size and site in the relations' join code segment respectively.

(2) Fitness function

The goal of query optimization is to obtain a QEP by spending the least cost. In GA algorithm, we calculate the cost of each QEP in each generation of population according to the equation (1) and (2), the inverse of the cost, namely fitness function value of each QEP, calculated as:

$$f(x) = 1 / COST(x) \tag{3}$$

Fitness function value is inversely proportional to the cost of QEP, the higher the fitness function value is, the more optimal corresponding QEP is.

(3) Population initialization

First of all, we randomly generated a certain size of a set of population. Each individual in the initial population is generated by using heuristic rules to avoid Cartesian product operation in individuals. Because not only Cartesian product operation will produce larger execution cost, but also the intermediate result of this operation is relatively large which will increase subsequent join cost.

(4) Selection operator

Selection is a main operator and also is an evaluation criterion of performance in genetic algorithm. If the influence of fitness to individual survival probability is too big in the process of selection, as a result, only a small number of the optimal individuals appear in the population of solutions, and then make GA trap in local optimum. On the contrary, if the fitness unable to restrain the change of individual survival probability, algorithm will end up with not regular random wandering behavior and can't convergence. In order to avoid the above problems, selection operator was introduced as follows:

Define all individuals within the current population as a set of P, $x^{(1)}$, $x^{(2)}$, …,$x^{(n)}$ is a fixed array of P. If x is a certain individual in the p, f(x) is the fitness of x, so the fitness of population P is defined as follow:

$$s(P) = \sum_{i=1}^{n} f(x^{(i)}) \tag{4}$$

For any individual in population P, their relative fitness is defined as r(x) =f(x)/s(P). Relative fitness of individual reflects their proportion of the population fitness. The selected probability is proportional to the individual fitness function. Accumulated fitness is defined as:

$$c(x^{(k)}) = \sum_{i=1}^{k} r(x^{(i)}) \tag{5}$$

Generating a real number t between 0 and 1 randomly before selecting, if t satisfy $r(x(k)) \le t \le r(x(k+1))$, then chose the k + 1*th* individual. Repeat the above process until it is finished to choose all maternal individuals of the next generation populations.

(5) Crossover operator

In GA algorithm, crossover operator imitates natural mating process through exchanging information between individuals. Crossing and regrouping a portion of two maternal individuals obtained by selector operator with probability P_c to get a new individual. Crossover operation makes the search ability of GA algorithm leap. The following order crossover operation was used in our study:

i Choose the cross region from code of two maternal individuals.

24|6317|58

85|7136|42

ii Add cross region to each other.

7136|24|6317|58

6317|85|7136|42

iii Remove the duplicate chromosomes, then get the new individual.

71362458

63178542

(6) Mutation operator

Mutation operator is designed to mimic biological gene mutation process. Mutation operator changes some of the genes on each of the individuals of generation with probability P_m. If fitness value of mutated offspring is higher than before, then offspring chromosomes is retained, otherwise, still retain the parent chromosomes. Our study uses the method of inversion mutation. Inversion Operation reverses gene sequence between two gene loci sequence randomly assigned in individual, thus form a new chromosome, which is the new join order.

Suppose the encoded string of individual X is (1 2 3 4 5 6 7 8). If the number rand () produced randomly less than P_m, then produce offspring X1 as (1 3 8 6 2 4 5 7) by randomly selecting two points from individual, such as 3 and 7, and inversing the part them.

At the end of every evolutionary, elite preserving strategy of GA algorithm directly make the best group of individuals in a population retain to the next group of generation, and the worst solution in the current iteration will be replaced by the best solution of last iteration, this ensures that population quality will not degenerate, so as to accelerate the convergence rate.

B. Dynamic Join Point of PGA-MMAS Algorithm

The key of the PGA-MMAS algorithm is to find the best connection point of two algorithms. An adaptive dynamic fusion method was proposed to ensure that genetic algorithm and Max-Min ant system combine in the best time.

(1) Set the minimum and maximum genetic iterations.

(2) Calculate evolution rate of offspring population in genetic algorithm, and set the minimum evolution rate.

(3) Compare evolution rate of offspring population with the minimum evolution rate in the range of determine iterative times, if evolution rate of n consecutive generations is less than the minimum evolution rate, then optimize efficiency of genetic algorithm identified as low, so stop the process of GA algorithm and begin to MMAS algorithm.

C. Max-Min ant System Module

In MMAS algorithm [9], maximum all path pheromone are limited between the max value τ_{max} and the min value τ_{min}, pheromone which above or below this area will be adjusted automatically to τ_{max} or τ_{min} to ensure that the path has least pheromone is within the scope of the choice of ants, thus effectively avoid the stagnation of the search.

(1) Pheromone initialization

MMAS set the initial pheromone of each path as the maximum τ_{max}, but in order to solve the lack of pheromone in the early phase of MMAS, GA algorithm are applied to get a better solution group which is transformed into a certain amount of pheromone, so initial pheromone is set as:

$$\tau_I = \tau_C + \tau_G \tag{6}$$

Here, τ_C is the constant of pheromone, which is set according to query optimization problem. τ_G is a initial pheromone transformed by GA.

(2) Transition probability

Every ant on the node choose the next node according to transition probability defined by the formula (7):

$$P_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s \in allowed_k} \tau_{is}(t)\eta_{is}(t)} & j \in allowed_k \\ 0 & other\ \ case \end{cases} \tag{7}$$

Here, τ_{ij} represents pheromone intensity of edge (i, j), $\eta_{ij}=1/C_{ij}$ represents expectations that ant transfer from relation i to j, C_{ij} is join cost between relation i and j. α is relative importance of trajectory, β is relative importance of expectations. $allowed_k$ is a set of relations that ant k hasn't choose.

(3) Pheromone update

Local pheromone updating formula:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + (1-\rho)\tau_{ij}^{\text{int}} \tag{8}$$

In the process of path search, the ant connects the next relation according to probability formula (7). Local pheromone updates again every time each ant join a relation. τ_{ij}^{int} represents the initial amount of pheromone of connection edge (i, j), ρ is a constant between 0 and 1, representing the volatile factor, which prevent MMAS algorithm getting stuck in local optimal solution.

Select the best ant after all ants completing join, only the best ant can leave pheromone, namely enhanced pheromone for only the optimal solution. In order to speed up the evolution algorithm, and make ant to be concentrated around the optimal path, in proposed algorithm, cut down the worst path pheromone at the same time. Update global pheromone in accordance with the formula (9) - (12).

Global pheromone updating formula:

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + (1-\rho)\Delta\tau_{ij}^{best} \tag{9}$$

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + (1-\rho)\Delta\tau_{ij}^{worst} \tag{10}$$

$$\Delta\tau_{ij}^{best} = 1/L_{best} \tag{11}$$

$$\Delta\tau_{ij}^{worst} = 1/L_{worst} \tag{12}$$

$\tau_{ij}(t+n)$ is the pheromone of join edge (i,j) of relations i and j in optimal join order after all ants finished all join of relations. L_{best} and L_{worst} represent the cost of the optimal and worst join order in every iteration.

D. GA-MMAS Parallelization

When dealing with large-scale data join query, GA algorithm and MMAS algorithm are extraordinarily time consumption, in order to solve this shortage, our study explore the application of hybrid GA and MMAS algorithm from aspects of parallel and distributed processing.

Parallel GA - MASS algorithm taking advantage of the distributed computing cluster to allocate a GA-MASS algorithm to several parallel processor to improve the execution speed and shorten the execute time. Firstly, a set of optimal solution is produced by GA algorithm in every processor and transform them into a certain amount of the initial pheromone. Then unify the initial solution of each ant colony. Finally, execute MMAS algorithm in parallel to get the more optimal solution. The basic process of PGA-MMAS algorithm is as follows:

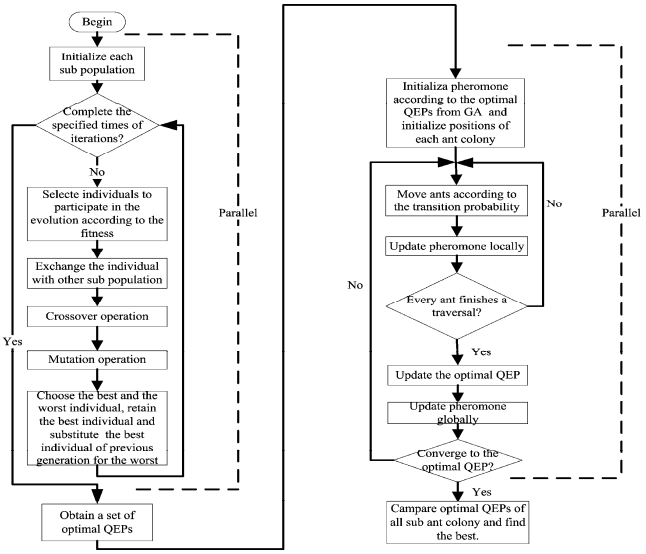

Figure 1. The flow chart of PGA - MMAS algorithm

In each iteration process of GA algorithm, genetic operation between each individual are independent, so we can take advantage of the inherent parallelism of this algorithm to improve the calculation speed [10]. MMAS algorithm is a high-performance ant colony algorithm, but its complex pheromone update rule also increases the time complexity of the algorithm, and its inherent parallelism can be a new breakthrough to solve the problem, as much as possible to shorten the search time, thus algorithm can quickly converge to higher quality solution.

In PGA-MMAS algorithm, first of all, the entire population was divided into q sub populations with same size, each thread performs genetic operation for a sub population independently. Sub population communication with each other when performing selection operation, first of all, compute local accumulation fitness r (x) of each population according to (5), and then select a certain number of individuals in accordance with the r (x) and exchange them with other threads. At the same time, each sub populations select maternal individuals to wait to cross with individual come from other sub individuals. Take following step for individuals come from other sub populations:

(1) Randomly select one from the local maternal individuals;

(2) Compare fitness of local maternal individual and individuals from other sub populations, and select one with highest fitness.

In every process of evolution, each sub population retains their local optimal solution to replace the worst solution in next evolution. Main thread get the best one of local optimal solution of each sub population when every sub populations finish the preset times of evolution.

Similarly, PGA-MMAS process searching process of q sub ant colony in parallel. Each sub ant colony search an optimal QEP according to serial MMAS algorithm and submit them to the main thread, then the main thread compare all of submitted QEP and modify the global optimal QEP. Finally adjust current global pheromone by applying the global update rule to guide ant to optimize continually. And that cycle repeats until PGA-MMAS algorithm get a globally optimal QEP.

IV. Experimental Results

There are four kinds of query optimization algorithm to be simulated for different number of relations. They are genetic algorithm, Max-Min ant system, GA-MMAS algorithm, PGA-MMAS algorithm promoted by our paper.

For GA algorithm, the initialization parameters set as follow: α=1, β=2, n=100, P_m=0.2, P_c=0.5. Here, n stands for population size. For MASS algorithm, the amount of ant is 30, the, α=1, β=5, ρ=0.8, τ_{max} =10, τ_{min}=0.1. The degree of parallelism is set as 2. The numbers of join relations are 5, 10, 15, 20, 25, 30 respectively. Running 200 experiments for each case and randomly select 10 times to average, the result are showed in figure 2:

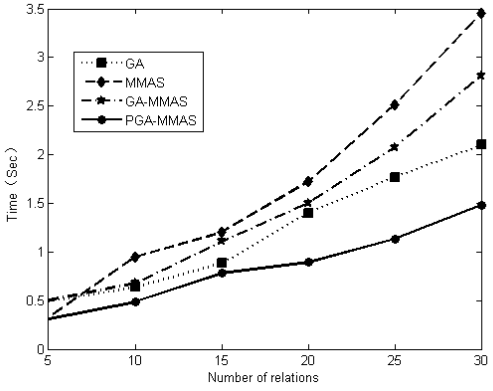

Figure 2. Comparison of execution time of four query optimization algorithms

From the graphs in Figure 2 we can find that the search time of GA-MMAS algorithm is between GA and MMAS. This is because in GA-MMAS, GA obtains a set of optimal solution and transform them into initial pheromone of MMAS, thus to speed up the search for the optimal QEP. PGA-MMAS is the most time-efficient by executing GA-MMAS in parallel.

In order to prove the advantages of PGA-MMAS algorithm further, we verify it in distributed database. We build a distributed database according to the requirements for an enterprise, which has about 150 million records in every enterprise data table. Execute the query operation according to the optimal QEP obtained by each algorithm, the total time of execution of algorithms plus execution of found QEPs are showed as in Figure 3:

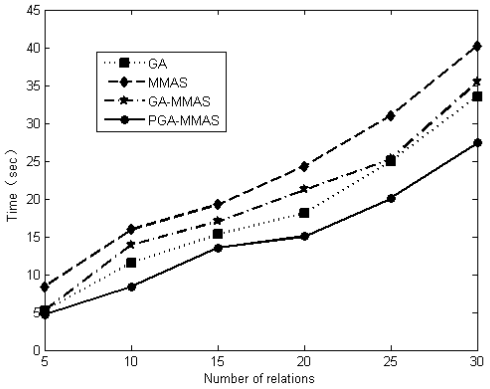

Figure 3. Total time (execution time of algorithms plus execution time of found QEPs) for four algorithms

As can be seen in the figure 2 and figure 3, although GA spend relatively few time to search the optimal QEP, but its unsatisfactory QEP take a longest time to execute, while MMAS is opposite. GA-MMAS combines the advantages of two basic algorithms, and its query efficiency is elevated. PGA-MMAS fully shows the superiority of parallelism, especially when the number of relationship is greater, the search time of optimal QEP of PGA-MMAS relatively less compared with other algorithms and its high quality QEP also reduced the query execution time.

V. CONCLUSION

PGA - MMAS algorithm proposed in this paper through the fusion of GA and MMAS attain to complement each other's advantages. GA makes up for slow convergence of MMAS caused by inadequate initial pheromone, and MMAS can solve the problem of the GA that it is easier to fall into the local convergence. At the same time, take advantage of parallelism of algorithm itself and distributed database cluster environment to accelerate the algorithm convergence further. The experimental results have showed the efficiency and effectiveness of PGA - MMAS algorithm.

ACKNOWLEDGMENT

This research was supported by the Guangxi Natural Science Foundation (2013GXNSFAA019334 and 2014GXNSFAA118387).

REFERENCES

[1] A. Hameurlain and F. Morvan, "Evolution of query optimization methods," Transactions on Large-Scale Data-and Knowledge-Centered Systems I. Springer Berlin Heidelberg, pp. 211-242, 2009.

[2] D. Kossman and K. Stocker, "Iterative dynamic programming: a new class of query optimization algorithms," ACM Trans. Database Syst., 25(1), pp. 43-82, 2000.

[3] A. K. Giri and R. Kumar, "Distributed query processing plan generation using iterative improvement and simulated annealing," 2013 IEEE 3rd International Advance Computing Conference, pp. 757-762, Feb. 2013.

[4] T. Kumar, V. Singh and A. K. Verma, "Distributed query processing plans generation using genetic algorithm," International Journal of Computer Theory and Engineering, pp. 38-45, 2011.

[5] Z. Zhou, "Using heuristics and genetic algorithms for large scale database query optimization," Journal of Information and Computing Science, pp. 261-280, 2007.

[6] N. Li, Y. Liu, Y. Dong, et al, "Application of ant colony optimization algorithm to multi-join query optimization," Advances in Computation and Intelligence. Springer Berlin Heidelberg, pp.189-197, 2008.

[7] H. Kadkhodaei and F. Mahmoudi, "A combination method for join ordering problem in relational databases using genetic algorithm and ant colony," IEEE International Conference on Granular Computing, pp. 312-317, Nov. 2011.

[8] C. Lee, C. S. Shih and Y. H. Chen, "Optimizing large join queries using a graph-based approach," IEEE Trans. Knowl. Data Eng., vol. 13, no. 2, pp. 298-315, Apr. 2001.

[9] T. Stützle and H. H. Hoos "MAX – MIN ant system," Future generation computer systems, pp. 889-914, 2000.

[10] F. González Bulnes, R. Usamentiaga, D. Fernando Garcia, et al. "A parallel genetic algorithm for optimizing an industrial inspection system," IEEE Trans. in Latin America, vol. 11, no. 6, pp. 1338-1343, Dec. 2013.