Distributed GEP Query Optimization on Grid Service

DENG Yong
College of Computer
Nanjing University of Posts and Telecommunications
Nanjing, China
jsxxt@126.com

WANG Ru-chuan
College of Computer
Nanjing University of Posts and Telecommunications
Nanjing, China
wangrc@njupt.edu.cn

***Abstract*—In order to better solve the problem of distributed query optimization, a querying optimization algorithm on GEP (QO-GEP) is present. On the basis of QO-GEP, distributed GEP query optimization on grid service (DGEPQO-GS) is proposed which combines with grid service. Simulated experiments shows that with the increment of the number of query relation, query time which QO-GEP carries out querying decreases apparently, meanwhile, with the number of grid nodes, the average querying success rate of DGEPQO-GS increases significantly.**

Keywords- Gene expression programming; Distributed query; Grid service; Query relation

I. INTRODUCTION

Distributed query processing is the result of the combination of database and distributed computing technology, and it's one of research topics of distributed database. On the grid platform, the queries users submit requirement by means of distributed query if the applications are established among many grid points.

Presently, many researchers have started to research how to realize the distribute query of distributed database on the grid platform. The Polar project studied in literature [1] has realized a highly-efficient distributed query procession method with the features of grid. In the project, the researchers hold that the most important characteristics of the distributed query on grid is concurrency, self-adaptive and dynamic nature, so it must make full use the powerful parallel and distributed computing power to adapting the characteristic that grid resource is dynamic. Moreover, OGSA-DQP is mainly based on OGSA-DAI, and realize distributed query by using various grid services of OGSA-DAI[2]. OGSA-DAI[3-4] is a middleware of data accessing, which can access and integration many distributed data resource by grid. Its main purpose is referring the data resource accessing and managing services by the form of grid service based on OGSA. These services include GDS, GDSF, DSR and GDTS, but these only provide the basal functions and interfaces. If the user wants to realize the management, accessing and the transparent dealing of distributed data, he musts exploit new services such as distributed query, transaction processing based on the basic services. A distributed query optimizing model on grid service is advanced in literature [5], which integrates the all processing of query optimizing and considers every parameter in the processing. A grid information query optimizing on heuristic search algorithm is put forward in literature [6]. It is showed in simulation that heuristic search algorithm can improve the efficiency of distributed grid query.

Query optimizing is always a very important and complex problem in data processing, which essentially is performing a complex searching question. This means that, for every given query, it will pick out a best program in semantically equivalent programs. There are many query optimizing algorithm now, but along with the addition of the relationship complexity among pending queries, the running time of random search algorithm based GA will be faster, but the efficiently of GA is low and it can't get the best answer easy. The Gene Expression Programming (GEP) is a new member of genetic family, the efficiency of which is higher than the traditional GA and GP arithmetic by 4-6 orders of magnitude [7-8] in solving complex problem. Therefore, this paper proposes the Distributed GEP Query Optimization on Grid Service (DGEPQO-GS), which is based on traditional GEP arithmetic and combines the ideal grid service.

The main job in this paper is listed below: (1) to propose querying optimization on GEP (QO-GEP) is proposed; (2) In order to meet the requirements of distributed query in grid , to propose the distributed GEP query optimization on grid service; (3) to make simulated experiments and give performance analysis.

The rest of the paper is organized as follows: The QO-GEP is introduced in the section 2. Section 3 proposes the distributed GEP query optimization on grid service. Section 4 shows comparative experiments and performance analysis. Section 5 concludes the paper.

II. QUERYING OPTIMIZATION ON GEP

In the grid platform, when the query plan is distributed to the nodes of grid, we must do the optimization process for this distributed query plan. Moreover, query optimization essentially is a complex search problem. For this problem, we propose the Querying Optimization on GEP (QO-GEP) based on the GEP arithmetic.

A. GEP Coding

In the distributed data query constituted by every grid node, a multi-join query can expressed using a query tree.

978-1-4244-5824-0/$26.00 ©2010 IEEE

That shows the priority which should be meted by the son operation of query. And in GEP, the object the arithmetic deals with is also the K-expression tree.

Definition 1(Basic Relation, BR): a basic relation means an operation relation or the combination of some operation relation.

Definition 2 (Basic Connection, BC): a basic connection means a result relation of connection operation formatted after the connection of two basic relations.

Definition 3(Querying Expression Tree, QET): we name the query trees those meet the follow conditions QET:

1) Every inside node has and only has two son nodes;
2) There is a basic relation or basic connection in the left and right node of every inside node;
3) The leaf node means relation.

Definition 4(query chart): query graph is a undirected graph, the point of the graph is denoted by the relation of query plan, and the connection of relation expresses the undirected edge of query graph. The query graph can used to judge if there are Cartesian product in the connection order of QET.

Definition 5(query gene, QG): a QG is build up of gene head and gene tail. The gene head is formed by the random choice in connection functions of multi-connection expression, and gene tail is formed by the relation of every query node.

Definition 6(Query Chromosome, QC): the whole QC is formed by one or many query genes. If there are many genes, the connection of every gene is using "and" function.

The Querying Optimization on GEP takes the QET as its strategy space. Every QET represents a query plan, and also represents the connection order of relations. Because the basic operating object is expressing tree, it will primly code for multi-connection expression of distributed database query in grid by using GEP coding rules.

Example 1: now we have the following coding and corresponding GEP expression tree construction, in which, "j" expresses connection function "join", and the set expresses connection relations:

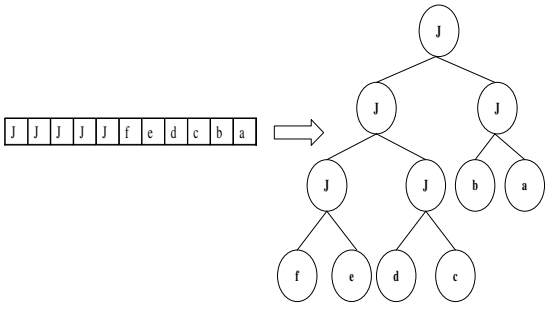

Figure1. coding for multi-connection expression on GEP

Through this coding, GEP can easily analysis this expression to the corresponding query expression. The query expression correspond the Fig.1 is:

((f Join e) Join (d Join c)) Join (b Join a)

B. GEP Fitness Function

The aim of query optimization is getting a operating plan of which the cost(Ti) is least. For every QET, we firstly compute the cost of every QET by using cost estimating function (we can get it by the num of connection relations and the size of connection relations). Then we will make a linear constant divide the cost, the computing function likes follow:

$$f_0(T_i) = K / \cos t(T_i), K = \min_{1 \le i \le N}(\cos t(T_i)) \quad \ldots\ldots (1)$$

Next, at the same time of evaluating the connection cost of QET, we will check if there is one edge of query map between the current relation and the front relation. If no, there must be Cartesian product, and we should add a corresponding penalty factor for that QET. In this, is between -1 and 0, which is relating to the num of Cartesian product. The more the num is, is nearer to -1. Finally, we made the function 2 be the fitness function GEP needed.

$$f(T_i) = f_0(T_i) + \theta = K / \cos t(T_i) + \theta, K = \min_{1 \le i \le N}(\cos t(T_i)) \ldots\ldots (2)$$

C. GEP Genetic Operation

The emerging of the next generation is decided by the genetic operation operators, which mainly are selection, recombining, variation, inserting operator and so on. The computing rules of four operators designed for multi-connection query tree are showed as follow.

(1) Selection operation: the aim of selection operation is selecting the ones whose adaptation is biggish in the population to buildup a new population. The tournament selection method is used in QO-GEP, and in the same time, "holding the best ones" strategy is used to holding the biggest adaptation one appeared in the evolution.

(2) Recombining operation: in the QO-GEP, we use one point recomposing manipulation. We randomly choose two different query chromosomes and the place K for recombining. At place K, we exchange all the elements in front of K each other.

(3) Variation operation: the purpose of variation is protecting the variety of the element in population, increasing the space of arithmetic searching, and speed-up to searching the best answer. Because of the particularity of query optimization, the variation operation can only work in the gene head and tail. In which, the gene head can variant to other connection function or connection relation. And the gene tail can only variant to other connection relation.

D. Description of Algorithm

The problem for answer the query optimization actually is a problem for searching the best answer. However, GEP is one of global optimization arithmetic, is a good choice to answer the query optimization. We take the query response time and query plan generating time to measure the QO-GEP.

Definition 7(Query Response Time): query response time means the time consuming between asking for query from client to returning the answer.

Definition 8(Query Plan Generating Time): query plan generating time means the time consuming between accepting the query plan and getting the best query list by using GEP to operating the query optimization.

We can see from the definitions that the littler the query response time and query plan generating time are, the littler the cost for query optimization by using this arithmetic is.

The whole QO-GEP is described as follow:

Algorithm 1： Querying Optimization on GEP (QO-GEP)

Input: the parameters of GEP;

Output: the best query expression QExpression;

Begin

1. Initialize the population;
2. while (gen<MaxGen) {
3. Decoding and computing the adaptation value using the function 2;
4. Keeping the best element in population;
5. Do the selection, recombination and variation operating;
6. Generate a new generation;
7. Return QExpression;//return the QExpression which's adaptation value is biggest.

III. DISTRIBUTED GEP QUERY OPTIMIZATION ALGORITHM BASED ON GRID SERVICE

A. Algorithm idea

Distributed algorithm is performed on the basis of WSRF and Server-side development is the most important, for the whole performing process is finished in the server side which can simplify the development of the client side and make full use of grid shared resource and the advantage of synergistic processing, for which in the service-side development of this paper various algorithm services and final display service are the most important and the development steps of grid services with different functions, but the specific parameters are different. Next we explain in detail how to use GT4 WS-Core to develop grid service, the all steps are shown as follows:

- Set the corresponding variables of grid service;
- Code WSDL, WSDD, JNDI, build.xml and necessary properties files of the corresponding grid service in which WSDL defines the specific operation of the grid service and necessary parameters, WSDD provides some information about grid service itself for grid container and tells grid service container how to announce grid service, JNDI is the API which is used by query object and through calling it the application can fixed the resource fixed by JNDI name and the location of other application object, build.xml is mainly used for coordinating the operation of GT4 WS-Core compile file and properties files mainly contain guiding all compile task to create using .gar and deploying the properties necessary for the process.
- Code all grid server side. The grid service code of DM algorithm is coded according to the algorithm parameters and operations defined by WSDL files and inputting parameters on the client side can call corresponding grid service perform distribute data mining remotely.

In order to perform distribute query optimization on grid service environment, at first, it is necessary to send the query relationship set to query server side, then call QO-GEP algorithm to perform query optimization and return the result of distribute query.

It is also important that all grid nodes combine the local query optimization relationship sets obtained by query which can get the global query optimization relationship set. In order to obtain the global query optimization relationship set here, is defined as a query optimization relationship set obtained by the ith grid node, is defined as a query optimization relationship set obtained by the jth grid node and there are rules as follows:

Rule 1： $\forall i \neq j$， if $a_i \neq b_j$， then combine { a_i, b_j } ;

Rule 2： $\forall i \neq j$， if $a_i = b_j$， then b_j replaces a_i or a_i replaces b_j ;

B. Description of the algorithm

A complete algorithm under the grid platform includes the client side and the server side and the distribute GEP query optimization algorithm can be described from the server side and the client side as follows

Algorithm 2： Distributed GEP Query Optimization on Grid Service, DGEPQO-GS

Input: all parameters of GEP;

Output: optimal query expression: QExpression;

Begin {

Server side：

1. Receive(GEPpara, QOGEPGSH, i, QueryingSet[i]); //Receive the parameters of GRP algorithm, specific addresses of QOGEP algorithm and query relationship waited to be optimized;
2. Initial (); //Init the population;
3. while (gen<MaxGen) {
4. Decode and calculate fitness value according to formula (2);
5. Reserve the best individual to the population;
6. Perform the operation of choosing, reorganizing and varying;
7. Re-generate a new generation of population;}
8. Return local query optimization expression;

Client side：

9. int gridcodes=SelectGridCodes ();// Make the choice to deploy QO-GEP service algorithm to grid nodes;
10. for (int i=0; i<gridcodes; i++) {
11. DataTransService (i, QueryingSet[i]);//Send their query optimization sets to designated grid nodes through data transmission service;
12. Receive(GEPpara, QOGEPGSH, i, QueryingSet[i]);//Transmit parameters necessary to the server side;

13. Return the local query optimization relationship sets obtained by the *i*th grid node;}

14. Combine the local query optimization relationship sets obtained by all nodes according to rule 1 and rule 2;

End

It can be seen from above algorithm description that the perform time of the complete GEP query optimization algorithm on the basis of grid service mainly includes the transmission time of all query plans and the calculation time of local query optimization sets and the time complexity of performing GEP algorithm under grid is unchanged.

IV. SIMULATION AND ANALYSIS

In order to verify the feasibility and validity of distribute GEP query optimization algorithm on the basis of gird service, this paper do some related experiments in the laboratory environment. The experiment platform is WindowsXP+WS-Core-4.0.2+Jdk1.5+Tomcat5.17+SQL Server 2000 and all code is implemented by Java.

Experiment 1: Gene expression programming is a random evolutionary algorithm, which is influenced by many parameters including population size, reorganization probability, mutation probability, algorithm termination condition and so on. In experiment 1, the GEP parameters used by QO-GEP algorithm displays as table 1. According to the parameters of table 1, experiment 1 compares QO-GEP algorithm, the traditional genetic algorithm and query optimization of the genetic programming. Table 2 shows the average query response time and average query plan generation time of three algorithms.

Table 1 parameters of QO-GEP algorithm

Algorithm name	Population size	reorganization probability	Mutation probability	Number of maximum evolution
QO-GEP	500	0.33	0.044	10000

Table 2 the comparison of query optimization results of three algorithms

Algorithm name	The number of query relationship	The average query response time (s)	The average query plan generation time (s)
Generic algorithm	6	2.13	1.26
	12	4.32	2.34
	24	11.36	5.71
Generic programming	6	1.95	1.38
	12	3.91	2.66
	24	8.26	6.01
QO-GEP	6	1.91	1.43
	12	2.83	2.71
	24	5.23	6.11

It can be seen from table 2 that with the addition of the number of query relationship, the average query response time is 53.96 percent less than generic algorithm and 36.68 percent less than generic programming and the larger the query relationship number is, the larger the decline rate of the average query response time of QO-GEP algorithm. The average query plan generation time of QO-GEP algorithm is larger than generic algorithm and generic programming with the addition of query relationship number, because QO-GEP algorithm uses the form of query expression tree which reduces the actual search space, but decoding costs some time which makes QO-GEP algorithm cost some time. However with the addition of the number of query relationship, the query time of QO-GEP algorithm is bound to a substantial reduction which shows the high efficiency of GEP algorithm solving complex problems.

Experiment 2: *Experiment 1* shows the query response time of solving centralized query using GEP algorithm is shorter than other traditional algorithms, while Experiment 2 shows the average query success rate of grid service distribute query optimization using corresponding GEP query algorithm is better than other algorithms. Fig.2 shows that the average query success rate of DGEPQO-GS query optimization has an upward trend.

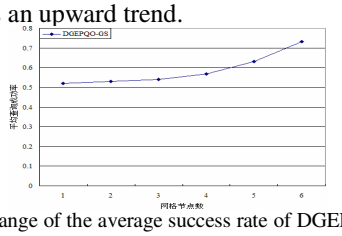

Figure 2. the change of the average success rate of DGEPQO-GS query optimization

Fig.2 shows that with the addition of the number of grid node, the average query success rate has an upward trend, because performing distribute query optimization using grid service and combining rule 1 and 2 make the obtained global query relationship is simple and optimal which is convenient for client side to query the result.

V. CONCLUSION

Distribute query is one of the important direction of the research of grid database. This paper proposes Distributed GEP Query Optimization on Grid Service (DGEPQO-GS) on the basis of GEP query optimization algorithm and combining the concept of gird service. Simulation shows the average query response time and average query plan generation time of the query optimization algorithm on the basis of GEP is smaller than traditional generic algorithm and generic programming and with the addition of the number of grid node, the average query success rate of DGEPQO-GS increases significantly.

ACKNOWLEDGMENT

The subject is sponsored by the National Natural Science Foundation of P. R. China (No. 60573141&60773041), the Natural Science Foundation of Jiangsu Province (BK2008451), National 863 High Technology Research Program of P. R. China (No. 2007AA01Z404, 2007AA01Z478), , Innovation Project for

University of Jiangsu Province (CX08B-085Z、CX08B-086Z).

REFERENCES

[1] J.Smith, A.Gounaris, P.Watson, N.Paton, A.Fernandes, R.Sakellarion. Distributed Query Processing on the Grid[C]. In Proceedings of Grid Computing 2002, Springer, LNCS2536, 2002:279-290.

[2] M.Alpdemir, A.Mukherjee, N N.Paton, P.Watson, A.Fernandes, A.Gounaris, J.Smith. Service-based Distributed Querying on the Grid[C]. In Proceedings of the First International Conference on Service Oriented Computing, Springer, December 2003:467-482.

[3] Anjomshoaa.A, Anlonioletti.M, Atkinson M, et al. The Design and Implementation of Grid Database Services in OGSA-DAI [C]. Nottingham: In Proceedings of UK e-Science All Hands Meeting, 2003..

[4] Xu Zhi-Wei, Feng Bai-Ming, Li Wei. Grid Computing Technology[M]. BeiJing: Publishing House of Electronics Industry, 2004.

[5] LUO Yong-hong; 2; CHEN Te-fang; ZHANG You-sheng. Distributed query optimization model based on data grid[J]. Journal of Computer Applications，2008,28(10):2553-2557.

[6] ZHANG Wei; LI Xian-xian. Grid Information System Search Optimization Based on Heuristic Search Algorithm [J]. Computer Engineering,，2008,34(19):26-29.

[7] C. FERREIRA. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence (2nd Edition) [M]. Springer, May 2006.

[8] Ferreira, C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems [J].Complex Systems, 2001, 13(2):87-129.

[9] Yang Li, Chang Yue-lou. Parallel database technology [M]. Chang Sha：University of Defense Technology Press，2000.