MANAGEMENT REVIEW

Evaluation of query term and genetic multiple query in distributed query processing environment

Er.Sambit Kumar Mishra

Associate Professor
Deptt. of Computer Sc.& Engg.
Ajay Binay Institute of Technology
Cuttack

Prof.(Dr.)Srikanta Pattnaik

Professor
S.O.A. University , Bhubaneswar

Abstract:

Query optimization is the task of improving the strategy for processing a database query. Query processing refers to the range of activities involved in extracting data from a database. A database query on relational databases can be represented as a query tree, where the leaf nodes represent accesses to relations. The intermediate nodes process and combine the data from their input nodes using physical implementations of the relational operations of projection, join etc. Usually the exploratory queries are loosely structured and require only minimal user knowledge of the source network. Evaluating an exploratory query usually involves the evaluation of many distributed queries. As the number of such distributed queries can quickly become large, we attack the optimization problem for exploratory queries by proposing several multi-query optimization algorithms that compute a global evaluation plan while minimizing the total communication cost, a key bottleneck in distributed settings.

Keywords

Sub population, chromosome, mutation, exploratory queries, query optimization, genetic multi query processing

Introduction

Query optimization is the task of improving the strategy for processing a database query. Query processing refers to the range of activities involved in extracting data from a database. A database query on relational databases can be represented as a query tree, where the leaf nodes represent accesses to relations. The intermediate nodes process and combine the data from their input nodes using physical implementations of the relational operations of projection, join etc. Usually the exploratory queries are loosely structured and require only minimal user knowledge of the source network. Evaluating an exploratory query usually involves the evaluation of many distributed queries. As the number of such distributed queries can quickly become large, we attack the optimization problem for exploratory queries by proposing several multi-query optimization algorithms that compute a global evaluation plan while minimizing the total communication cost, a key bottleneck in distributed settings.

Review of Literature :

Goldberg & Richardson et.al[1] have presented an implementation of the concept known as the sharing method. In the context of their study, each individual in a sub population can consume a fraction of the available resources : the greater the population size of the niche, the smaller the fraction. This leads towards a steady state in which subpopulation size are proportional to the amount of the corresponding available resources.

Horng & Yeh et.al[2] proposed an approach to automatically retrieve keywords and then uses genetic techniques to tune the keywords

weights. The effectiveness of the approach is demonstrated by comparing the results obtained to those using a PAT-tree based approach.

In their work, their goal is to develop a specific genetic model for multiple query evaluation. As the main aim of this technique is dispersion of the relevance regions in the document space, they proposed the use of a suitable genetic technique for solving multimodal problems.

Yang & Korfhage et.al[3] have proposed a GA for query optimization by reweighting the query term indexing without query expansion . They used a selection operator based on a stochastic sample, a blind crossover at two crossing points, and a classical mutation to renew the population of queries. The experiments showed that the queries converge to their relevant documents after six generations.

Kraft & al et.al[4] have applied GA programming in order to improve the weighted boolean query formulations. Their first experiments showed that the GA programming is a viable method for deriving good queries.

Implementation using G.A.:

Inspired by natural selection, G.A. are abstractions of biological evolution and is thus a method for moving from one population of chromosomes to a new population by using a kind of natural selection together with genetic inspired operators of recombination, mutation and inversion. In G.A. , the solutions are called individuals or chromosomes. After the initial population is generated randomly, selection and variation function are executed in a loop until some termination criterion is reached. When G.A. is applied to query optimization, the initial population is randomly generated. For each generation, genetic operations are carried out and thus the population evolves usually decreasing the average cost of its individuals. The goal of our GA is to find an optimal set of data which best matched the user's needs. The GA attempts to involve, generation by generation, a population of sub queries to the outcome of the system. The set of selected sub queries represents a set of individual queries exploring a specific region of the data according to their evaluation results. The representation of an individual query is of the form Qu (qu1, qu2, …, quT). Each gene corresponds to an indexing term or concept. Its value is represented by a real value and defines the importance of the term in the considered query. Initially, a term weight can be computed by any query term weight scheme; it will then evolve through the generations.

Problem formulation :

In this representation, the following formula may be used to evaluate query term .

$$qui = \frac{1 + \log(tfui)) * \log(ni)}{\sum Tk=1(1 + \log(tfuk)) * \log(nk))}$$

where qui = query term
ni= number of data containing term ti
tfuk= frequency term tk in data

Algorithm :

Begin
Submit the initial query and do the search
Judge the top thousand data in queries
Build the initial population
Repeat
For each sub population of the population
do the search
build the local list of data
Endfor
Build a merged list
Renew the sub population
Judge the top fifteen data
Compute the fitness of each individual query
For each sub query Qs(s) of the population
Repeat
parent1= Selection (Qs(s))
parent2= Selection (Qs(s))
Crossover (Pc , parent1, parent2,son)
Mutation (Pm , son, sonmut)
Add sub population (sonmut,Qs(s+1))
Until Sub population size (Qs(s+1)) = Sub population size (Qs(s))
Until a fixed number of feedback iterations End

Experimental results :

The experiments are carried out over 50 queries and set of relevant data of each query. The main goal of the experiment is to evaluate the effectiveness of the GA for multiple evaluations in comparison with a traditional query evaluation.

Population size=50
Crossover probability=0.7
Mutation probability=0.07

Table-1 : Single query evaluation

Iteration-1 (120)	Iteration-2 (230)	Iteration 3 (250)	Iteration-4 (285)
100	90	80	65

Table-2 : Genetic Multiple query evaluation

Iteration-1 (179)	Iteration-2 (160)	Iteration 3 (220)	Iteration-4 (270)
170	120	93	69

These experiment results indicate that the approach yields large improvements over a traditional simple evaluation process. It is noticed that the improvements vary from 9% to 15%.

Conclusion:

This approach is generally based on the integration of the sub population technique in a GA which performs a multiple query evaluation. The GA is also characterized by using operators adapted to the context of the retrieval task.

Reference:

1. *Goldberg & Richardson, Genetic algorithms with sharing for multimodal function optimization, in Proceedings of the second International Conference on Genetic Algorithm (ICGA) ,1987 pp 41-49.*

2. *J.T Horng & C.C Yeh , Applying genetic algorithms to query optimisation in document retrieval, In Information Processing and Management 36(2000) pp 737-759.*

3. *J.J Yang & R.R Korfhage, Query optimisation in informationretrieval using genetic Algorithms, in Proceedings of the fifth International Conference on Genetic Algorithms (ICGA),1993, pp 603-611, Urbana, IL.*

4. *K. L Kwok , A network approach to probabilistic information retrieval, ACM transactions on information systems, 1995 ,vol 13 N°3, pp 324-353.*