Advanced Science and Technology Letters
Vol.45 (CCA 2014), pp.65-69
http://dx.doi.org/10.14257/astl.2014.45.13

Selecting Materialized Views based on Genetic Algorithm

Zhixiang Zhu and Dongjin Yu

School of Computer, Hangzhou Dianzi University, Hangzhou, 310018, China
zhuzx_hdu@163.com, yudj@hdu.edu.cn

Abstract. As the data sets increase in data warehouse day by day, the cost used for OLAP operations becomes extremely high. One of the most effective ways to solve this problem is to build appropriate materialized views in the data warehouse. In order to select the appropriate views to be materialized with the possible minimized cost, we propose a novel approach to the materialized view selection problem based on a genetic algorithm. Experimental results show that the optimal solution for selecting materialized views can be obtained effectively using the approach presented.

Keywords: materialized view, multi-dimensional lattice, genetic algorithm, online analytical processing

1 Introduction

A materialized view is a database object that contains the results of a query. It may be a summary based on aggregations of a table's data. However, because there are many aggregations that can be calculated, often only a predetermined number are fully calculated while the remainder are solved on demand [1][2][3]. The problem of deciding which aggregations (views) to calculate is known as the 'view selection problem', which can be constrained by the total size of the selected set of aggregations, the time to update them from changes in the base data, or both. The selection of the appropriate views to be materialized has proven to be a NP-hard problem [4].

In this paper, we present a novel approach to materialized view selection based on a genetic algorithm. Our approach can select the appropriate views to be materialized with a minimized total cost. It can accelerate the search speed of the genetic algorithm and avoid premature convergence.

The rest of this paper is organized as follows. Section 2 introduces in detail the adaptive genetic algorithm for solving the materialized view selection problem. Section 3 demonstrates the experimental results. Finally, last section concludes the paper and outlines the future work.

2 Selection of Materialized Views based on a Genetic Algorithm

A genetic algorithm (GA) is a search heuristic that mimics the process of natural selection [5]. The parallelism, self-study nature and robustness of the genetic algorithm

ISSN: 2287-1233 ASTL

make it very effective in solving the combinatorial optimization problem [6]. We apply a genetic algorithm to solve the materialized view selection problem.

2.1 Encoding

Encoding transforms the feasible solution from the solution space of the problem to the searching space, in which the genetic algorithm can deal with it. In our approach, we use binary coding to transform the candidate view set of the multi-dimensional lattice model into an array of 0-1 integers. The node's serial number corresponds to the array index, whereas the value of the element determines whether the view should be materialized. The array element with a value 1 means that its corresponding view needs to be materialized, 0 means not.

2.2 Population Initialization

We use a random algorithm to generate n individuals as the initial population. For the given condition that the root node view in the multi-dimensional lattice model must be materialized, it is necessary to assign 1 to the value of the individual's first code element if its initial value is 0.

2.3 Fitness Function

In a genetic algorithm, the greater the fitness of the individual, the greater the chance it goes into the next generation. During each successive generation, the individual with the greatest fitness in the population represents the optimal solution. We adopt the reciprocal of the materialized views' total cost as the value of the individual's fitness.

$$Fit(x) = \frac{1}{TotalCost(Q,M)_{g=x}} \quad (1)$$

2.4 Selection Operator

As a process of screening, selection retains those individuals with a high fitness and eliminates those individuals with a low fitness within a population. The selection operator that denotes the probability of an individual evolving to the next generation, which can be calculated as follows:

$$p_s = \frac{Fit(x)}{\sum_{k \in Population} Fit(k)} \quad (2)$$

in which, *Population* represents a group of individuals and *k* represents an individual in the population.

2.5 Crossover Operator

To 'crossover' means to combine parent individuals according to certain rules and produce offspring individuals. It generates feasible solutions to the problem and searches the candidate solution at the same time. In our approach, we apply a two-point crossover as the crossover operator. In other word, we exchange the parent individuals' code segments separated by two randomly selected crossover points to generate the offspring individuals.

2.6 Crossover Probability

The crossover probability determines whether the two parent individuals perform a crossover operation. The larger the crossover probability, the faster the speed of the introduction of new individuals and the greater the loss rate of individuals with a high fitness. In contrast, if the crossover probability is set too small, it may cause a search block and premature convergence. We recommend that the crossover probability ranges from 0.4 to 0.99.

2.7 Mutation Operator

Mutation changes the values of some of the genes of individuals to increase the diversity of the population, thus alleviating premature convergence. Since the root view of the multi-dimensional lattice must be materialized, we only make changes to the gene elements of non-root views. We randomly select the non-root view corresponding to the gene element of the individual. If the value of a given gene element is 1, we change it to 0. Otherwise, if the value of a gene element is 0, we change it to 1.

2.8 Mutation Probability

The mutation probability p_m determines whether the parent individuals should perform the mutation operation. We recommend that the mutation probability p_m ranges from 0.0001 to 0.1.

2.9 Terminal Condition of Iteration

Theoretically, the genetic algorithm terminates only when it obtains the global optimal solution to the problem. However, this solution is usually unknown. Therefore, it is necessary to set some approximate convergence criteria to terminate the execution of the algorithm. For our approach, we pre-set a maximum number of iterations that are used as the termination condition.

3 Experiment

In order to evaluate the correctness of the algorithm and the effectiveness of its adaptive adjustment mechanism, we carried out experiments on Windows 7 with an Intel Core 2 Duo E7500 (CPU 2.93 GHz, RAM 2GB). We compared our algorithm with the QAGA algorithm proposed in [7] for solving the problem of selecting materialized views. The total cost of the materialized views is used as the criterion to evaluate all three algorithms. The QAGA algorithm considers the size and query frequency of the views and employs the greedy strategy to select top-k views, which can bring the maximum benefit if the views are materialized. The costs of the two algorithms for different dimension materialized view selection problems are shown in Table 1, in which the figures are the averages after running 1,000 times. As Table 1 shows, no matter whether it deals with the low-dimensional materialized views or the high-dimensional ones, the AGA algorithm costs less than the QAGA algorithm.

Table 1. Comparison of the Total Costs of Two Algorithms

Dimension	AGA algorithm	QAGA algorithm
3	47,047	56,415
4	109,042	149,798
5	9,363,002	10,979,542
6	19,633,311	23,136,097
7	43,294,097	56,416,653
8	99,007,181	125,263,171
9	219,545,489	303,528,268
10	516,666,803	745,969,332

4 Conclusion

In this paper, we proposed a novel approach to the selection of materialized views based on a genetic algorithm. It selects the appropriate views to be materialized for a minimized total cost. The experiment verifies the appropriateness of the approach and the effectiveness of the adjustment mechanism. In the future, we will investigate approaches to the maintenance of existing materialized views at a minimal cost. The selection of materialized views for querying big data will be also on the list of our future work.

Acknowledgments. This work is supported by the national natural science fund project (No. 60903053) and the Zhejiang province natural science fund project (No. LY12F02003).

References

1. Li J, Li X, Lv J: Selecting Materialized Views Based on Top-k Query Algorithm for Lineage Tracing. 2012 Third Global Congress on Intelligent Systems (GCIS), IEEE, pp. 46--49 (2012)
2. Cuzzocrea A, Hacid M S, Grillo N.: Effectively and efficiently selecting access control rules on materialized views over relational databases. Proceedings of the Fourteenth International Database Engineering & Applications Symposium, ACM, pp. 225--235 (2010)
3. Harinarayan V, Rajaraman A, Ullman J D.: Implementing data cubes efficiently. ACM SIGMOD Record 25(2), pp. 205--216 (1996)
4. Sohn Jong-Soo, Yang Jin-Hyuk, Chung In-Jeong: Improved View Selection Algorithm in Data Warehouse. IT Convergence and Security, Lecture Notes in Electrical Engineering 215, pp. 921--928 (2013)
5. Mendes J J M, Gonçalves J F, Resende M G C.: A random key based genetic algorithm for the resource constrained project scheduling problem. Computers & Operations Research, 36(1), pp. 92--109 (2009)
6. Mehta M H.: Hybrid Genetic Algorithm with PSO Effect for Combinatorial Optimization Problems. International Journal of Advanced Computer Research, 2(4), pp. 300--305 (2012)
7. Kumar T V V, Haider M.: A query answering greedy algorithm for selecting materialized views. Proceedings of the Second international conference on Computational Collective Intelligence: Technologies and Applications. Springer Berlin Heidelberg, pp. 153--162 (2010)

