ANALYSIS AND COMPARISON OF ALGORITHM FOR SELECTING MATERIALIZED VIEWS IN A DATA WAREHOUSING ENVIRONMENT

Jiratta Phuboon-ob and Raweewan Auapanwiriyakul
School of Applied Statistics, National Institute of Development Administration, Bangkok, Thailand
EMAIL: jiratta.p@msu.ac.th, raweewan@as.nida.ac.th

ABSTRACT

Information is important corporate asset for supporting management’s decision. A data warehouse (DW) is a system, which has value and role for decision-making by querying. Queries to DW are critical because most of them are long and complex. They often access millions of tuples, and involve joins between relations and aggregations. Materialized views can provide the better performance for DW queries. But these views have maintenance cost, so materialization of all views is not possible. An important challenge of DW environment is materialized view selection because we have to consider trade-off between performance and view maintenance. Therefore, we have decided to analyze and compare some current proposal aimed to solve this challenge.

Key words: data warehouse, view selection problem, materialized views

1. Introduction

A data warehouse (DW) can be defined as a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management’s decision [1]. It can bring together selected data from multiple database or other information sources into a single repository [2]. To avoid accessing from base table and increase the speed of queries posed to a DW, we can use some intermediate results from the query processing stored in the DW called materialized views. Although materialized views speed up query processing, they have to be refreshed when changes occur to the base tables. Therefore, materialized view selection involved query processing cost and materialized view maintenance cost. So, many researches aim to make the sum of that cost minimal. Especially, join operation is one of the most expensive operations in query optimization. Some of them consider only join order optimization or aggregation operation, or both. The existing algorithms for solving query optimization, multiple query optimization, and materialized view selection can be classified into four categories according to deterministic algorithm, randomized algorithm, evolutionary algorithm and hybrid algorithm [3]. Some researchers apply those algorithms with and without employ techniques of multiple query processing (MQP). The rest of the paper is organized as follows. Section 2, we describe multiple-query optimization (MQO). We analyze and compare algorithm which aimed to solving the materialized view selection problem in section 3, and conclude in section 4.

2. Multiple-Query Optimization (MQO)

The previous works in the area of materialized view selection dealing with multiple query processing (MQP) have been proposed such as [3] [4] [5] [6] and [7]. The main idea of MQP is to find an optimal execution plan for multiple queries that will be executed by sharing common data and temporary result based on common operation/sub-expression. These common operations/sub-expressions are construct the global execute plan. Then MQP should be cheaper compared with a serial execution of queries. Suppose that given a set of queries \( Q = \{Q_1, Q_2, \ldots, Q_n\} \). Every query \( Q_i \) there exists at least one local processing plan. A global access plan that corresponds to \( Q \), provides a way to compute the results of all \( n \) queries, can be constructed by choosing one plan for each query and then merging them together. An optimal local plan is referred to the minimal cost plan for processing a query \( Q_i \), that corresponds to query optimization. An optimal global plan is referred to the global processing plan by merging the common part of optimal local plan. The multiple-query optimization (MQO) problem can be defined as: given \( n \) sets of local processing plans \( P_1, P_2, \ldots, P_n \) with \( P_i = \{P_{i1}, P_{i2}, \ldots, P_{ik}\} \) being the set of possible plans for query \( Q_i \), \( 1 \leq i \leq n; k_i \) is the number of local processing plans for query \( Q_i \). Find a global processing plan by selecting one plan from each \( P_i \) such that query cost of the global processing plan is minimized. We will talk in detail about algorithms that employ MQO technique in the third section.

3. Materialized View Selection
A view is a derived relation defined by an expression much like a query, which do not exist physically [8]. A view defined in this way is recomputed every time when it is invoked. Using view in database systems may be suitable, because information sources are changing frequently. However DW contains historical detailed data of organization to support of management’s decision by querying. So query processing in DW are more frequent than update base table. Using view in DW may be not appropriate. A view can be materialized by storing the result of view in the DW called materialized views. Materialized view is one of the techniques employed in DW to increase speed of queries by pre-calculating expensive operations on the DW. Although it constitutes redundant data, it is much cheaper and more satisfactory response time for returning results from the query to end-user in many cases than to process the whole query. Since, this technique has maintenance cost when base table changed, then it is not possible to materialize all views. So, how to select views to be materialized is challenge, considering trade-off between performance and view maintenance [9]. In this section we analyze and compare some current proposal aimed to solving how to select which views to be materialized. Whereas in section 3.1 we discuss proposals that use Deterministic algorithm, next we focus on the proposals of Randomized algorithm. In section 3.3, we describe paper using Evolutionary algorithm, then we analyze the Hybrid algorithm proposal. Finally, we compare all of these proposals. However, in each section, we will describe only the detail of research that we adapt for our experiment.

3.1 Using Deterministic Algorithm
Deterministic algorithm is an algorithm which behaves predictably. If it runs on a particular input, it will always produce the same output, and the underlying machine will always pass through the same sequence of states. This algorithm usually apply either heuristic or exhaustive search to construct a solution step by step [3]. Some current proposals which focus on deterministic algorithm that determines which views to be materialized when it is too expensive to materialize all of them are following:
Researchers that apply deterministic algorithm without employ MQP are [10] [11] [12] [13] [14] and [15]. Researchers that propose deterministic algorithm with employ techniques of multiple queries processing (MQP) are [5] [6] [7] and [16]. They use MQP techniques to build multiple view processing plan (MVPP) to identify views to be materialized. An MVPP is a directed acyclic graph that represents a query processing of DW views. In our paper, we adapt the algorithm proposed by [6].

3.2. Randomized algorithm
Randomized algorithms are based on statistical concepts that the large search space can be explored randomly. Each solution in randomized algorithms can be thought of as node in a solution space that has a cost to it. The aim of the optimization process is to find the one with the minimum cost. These states are connected by edges that defined by a set of moves from one state to another after applying a simple transformation. The move is called uphill if the destination state has an upper cost than the starting one and called downhill in the opposite case. A state from which no moves are downhill is called local minimum. The local minimum with the minimum cost called global minimum. The graph should be connected and there should be at least one path of transitions between any two states. Randomized algorithms start from a random initial state. Then they apply sequences of transformations trying to detect the global minimum [18].

The application of randomized algorithms, namely Simulated Annealing (SA) invented by [19], Iterative Improvement (II) exposed by [20], Two-Phase Optimization (2PO) inspired by [18], Toured Simulated Annealing (TSA) proposed by [21] and Random Sampling (RA) introduced by [22]. The existing proposal focus on view selection problem using randomized algorithms suggest by [23]. They explore the application of four randomized algorithms: II, SA, RA and 2PO to the view selection problem in data warehouse under the space constraint and the maintenance cost constraint without employ MQP. They found that 2PO is the best performance in all their tested case because it converges fast to a good local minimum.

3.3 Evolutionary algorithms
Evolutionary algorithms (EAs, evolutionary computation, artificial evolution) use a randomized search strategy similar to biological evolution in their search for good solutions. Although an evolutionary algorithm resembles randomized algorithms in this aspect, the approach shows enough differences to warrant a consideration of its own. The basic idea is to start with a random initial population and generate off spring by random variations (e.g., crossover and mutation). The “fittest” members of the population survive the subsequent selection. The next generation is based on these members. The algorithm terminates as soon as there is no further improvement over a period. The fittest individual found is the solution [3]. Examples of EAs are Genetic algorithms (GA), Genetic programming (GP), Evolution strategy and Evolutionary programming. Most of these techniques are similar in spirit, but differ in the details of their implementation and the nature of the particular problem to which they have been applied [24], [17] [25] [26] and [27] apply GA to view selection problem. They use this algorithm without employ MQP. In [17] explore the view selection problem in the context of OR view graphs. Before GA can be applied to solve a problem, we must encode for the solution and fitness function has to be chosen. They encode the solution of view selection problem in chromosome. Each chromosome is consisted of constant number of binary string 0 or 1. This constant number is the number of the
candidate views in the view graph. The string 1 denotes the corresponding node in that graph is selected to materialized, while string 0 is not selected. In the initialization of the population, [17] have chosen a population size of 30. For each genetic operator, they use the roulette wheel method for selection process. They assigned the specific probabilities values of 0.001 and 0.9 to crossover and mutation operators respectively. They applied two genomes to crossover operation. The selection, crossover, mutation and evaluation processes will be repeated in a loop until the termination condition is satisfied. In their experiments it is reached after 400 generations. For their problem, the fitness function has to evaluate a genome, set of selected views to materialized, with respect to the query benefit as well as with respect to the maintenance constraint.

3.4 Hybrid algorithms

Hybrid algorithms combine evolutionary algorithms and deterministic in various ways. Such as solutions obtained by deterministic algorithms are used as starting points for evolutionary algorithms or as initial population members for evolutionary algorithms. In other way, a deterministic algorithm can be applied to the best solution found by evolutionary algorithms [3]. [3] apply a hybrid evolutionary algorithm to solve three related problems. The first is to optimize queries. The second is to choose the best global processing plan from multiple global processing plans. The third is to select materialized views from a given global processing plan. Their experiment shows that the hybrid algorithm delivers better performance than either the evolutionary algorithm or heuristics used alone in terms of the minimal query and maintenance cost.

3.5 Experimental Studies

In our experiment, we employ MQP technique to all of three categories algorithm and use the same cost model proposed by [6] to compute query processing cost, materialized view maintenance cost and total cost. Since, the previous works in randomized algorithm did not employ MQP. But in our experiment, we employ MQP to all of algorithms. So we will not include randomized algorithm in our study. We do not consider any constraints. We use the TPC-H database of size 1GB as a running example throughout our paper. For more details on this benchmark refer to [28]. We adapt TPC-H queries. We observe that our six queries are defined over overlapping portions of the base data or intermediate query results. For example, Q3 can use the intermediate result of Q2. We assume that base table update once a time, and the frequencies of Q1 to Q6 are 2,2,5,1,5,1 and 3 respectively. We modify the algorithm to generate MVPP proposed by [6]. Although in six queries, we cover all the operations, but in MVPP we assume that methods for implementing select and join operation are linear search and nested loop approach respectively. Before comparing the cost, we compute query processing cost, materialized view maintenance cost and total cost of all-virtual-views and all-materialized-view that show in Table 1. We compare these costs between three categories as following:

In Deterministic algorithm, given a MVPP, we execute the view algorithm proposed by [6] to select materialized view. The view selected are Tmp2, Tmp4, Tmp7, Tmp13, Tmp23 and Tmp27. Using these six materialized views, we get 2,443,639,934,436 as query cost, 12,805,319,910,010 as maintenance cost, and 15,248,959,844,446 as the total cost. Based on these results, it would be benefit to materialize them, reducing the cost from 18,543,069,434,458 to 15,248,959,844,446. For evolutionary algorithm, [17] solve this problem without employ MQO. So we adapt the hybrid algorithm presented by [3] by executing evolutionary algorithm alone. We encode the solution of views selection problem in chromosome representation based on tree. We map a tree into a binary string by breadth-first traverse of the tree. For example, result of breadth-first traverse of the tree in Figure 5 is the following ordered list: [result4,0], [result1,0], [result6,0], [result5,0], [result2,0], [result3,0], [Tmp17,0], [Tmp27,0]. Each position in binary string corresponds to each node in the ordered list such as node named “result4” is not materialized if the first position of binary string is 0. So, a binary string of [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] means that no node is materialized. A string of {1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} means that nodes named “result4”, “result1”, “Tmp17” and “Tmp27” are materialized, but others are not. We set GA parameters like [17]. But we adopt the concept of ranking selection as selection operator. The results are Tmp2, Tmp4, Tmp7, Tmp15, Tmp23 and Tmp27. These materialized views get 1,756,326,884,131 as query cost, 13,861,732,693,750 as maintenance cost, and total cost is 15,618,059,577,881. This method is more expensive than the first approach, but better than all-virtual-views or all-materialized-views. For second phase, we run deterministic algorithm like the first experiment. The views selected are Tmp2,Tmp4,Tmp13,Tmp23 and Tmp27. Using these materialized views, we get 2,443,639,934,458 as query cost, maintenance cost is 12,805,319,910,000, and total cost is 15,248,959,844,458. This algorithm is more expensive than deterministic algorithm, but cheaper than GA. This method depends on the first phase. If the GA is very closed to the global minimum, then it will get the best solution.
Table 1 The Query Processing, Maintenance and Total Cost

<table>
<thead>
<tr>
<th></th>
<th>Cost of query processing</th>
<th>Cost of maintenance</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-virtual-views</td>
<td>18,543,069,434,458</td>
<td>-</td>
<td>18,543,069,434,458</td>
</tr>
<tr>
<td>All-materialized-views</td>
<td>2,402</td>
<td>19,090,167,891,957</td>
<td>19,090,167,894,359</td>
</tr>
</tbody>
</table>

4. Conclusions

In this paper, we analyze and compare the existing algorithms aimed to solve which views to be materialized. We classified those algorithms into four categories according to deterministic algorithm, randomized algorithm, evolutionary algorithm and hybrid algorithm. Researchers apply those algorithms with and without employ techniques of multiple query processing (MQP). For our experiment, we compare only three categories; deterministic algorithm, evolutionary algorithm and hybrid algorithm. We did not use randomized algorithm in our study because we employ MQP technique for each of the categories algorithm, while the previous work of randomized algorithm did not employ MQP. We consider neither the space constraint nor the maintenance cost constraint. We find that the result of deterministic algorithm proposed by [6] is better than algorithm proposed by [17] and algorithm proposed by [3].

REFERENCES