Efficient Utilization of Materialized Views in a Data Warehouse

Don-Lin Yang, Man-Lin Huang, and Ming-Chuan Hung

Department of Information Engineering
Feng Chia University, Taichung, Taiwan 407
dlyang@fcu.edu.tw, mlhuang@fcu.edu.tw, mchong@fcu.edu.tw

Abstract. View Materialization is an effective method to increase query efficiency in a data warehouse. However, one encounters the problem of space insufficiency if all possible views are materialized in advance. Reducing query time by means of selecting a proper set of materialized views with a lower cost is crucial for efficient data warehousing. In addition, the costs of data warehouse creation, query, and maintenance have to be taken into account while views are materialized. The purpose of this research is to select a proper set of materialized views under the storage and cost constraints and to help speedup the entire data warehousing process. We propose a cost model for data warehouse query and maintenance along with an efficient view selection algorithm, which uses the gain and loss indices. The main contribution of our paper is to speedup the selection process of materialized views. The second one is to reduce the total cost of data warehouse query and maintenance.

Keywords: Data warehouse, OLAP, materialized view, gain index, and loss index.

1 Introduction

A data warehousing system is not resource-oriented but subject-oriented, which is arranged upon users’ view and can be conveniently visualized via on-line analytical processing (OLAP) tools. It is opposite to an on-line transaction processing (OLTP) system that focuses on transaction throughput and maintaining consistency. A data warehouse (DW) serves as a large warehouse containing data from local, remote, and heterogeneous sources. In addition to better utilization of storage space, researchers are concerned about the problem of view materialization [1]. Materialization of views is strongly related to query effectiveness in a DW system. However, storage limitation prohibits the materialization of all possible views. In this research, we look for optimal query effectiveness on the premise that source data might change over time and the total cost of DW query and maintenance is minimized.

As indicated before, if available space is sufficient, we would like to generate all possible views in advance to speed up DW query process. For practical information services, one must also allow periodical updates (i.e., additions) to the DW for data validity and correctness. This will increase the maintenance cost of the materialized views. We should consider the following constraints when dealing with optimal
utilization of materialized views: (1) available storage space, (2) query processing efficiency, and (3) costs of DW query and maintenance.

In this paper, we use the vector data structure to determine the relationship of lattice and develop an algorithm to materialize an optimal pool of shared views with the lowest cost. There are three characteristics in this algorithm. First, “indices of gain and loss” and “candidate view” are introduced in the optimization of adding and deleting materialized views. Second, in addition to query cost, maintenance cost is also taken into account. Third, we improve the query process by adopting a vector data configuration to speedup the lattice operation in searching dependent views.

The outline of our paper is as follows. Section 2 discusses materialized views and reviews some related work. In Section 3, we present the architectural design of our algorithm. Section 4 describes the cost model used in our algorithm. In section 5, we detail our algorithm for materialized view selection. Section 6 discusses the experiment results. We conclude the paper in Section 7 and sketch some ideas for future work.

2 Literature Review

Selection of optimal materialized views is an NP-complete problem. Since 1995, researchers have been interested in materialized views. However, they were in an all or nothing mode for materializing views. In 1996, Harinarayan proposed the “greedy algorithm” [11], in which one materializes the top view that other views depend on. Then new views are added according to the benefit from the query efficiency of materialization in steps. In the same year, Ross [8] considered the use of additional views to reduce maintenance cost.

Unfortunately, if storage space is short of containing all the views that are dependent by other views, then greedy algorithm has the worst result. In 1997, Chen [2] used the two-phase algorithm to select materialized views. The purpose of the first phase is to reduce the space need, and the second phase is to reduce the query cost under the remaining space. Yang [7] also proposed an analysis framework for materialized views by using Multiple View Processing Plan (MVPP). The MVPP method considers the cost as the factor of view materialization. It does not take required space into account.

In 1999, Lin [9] proposed an improved solution on performance using genetic algorithms. Gupta [5] thinks storage devices are cheap. Under given maintenance cost, he developed a method to select materialized views effectively without worrying about space. Theodoratos [4] proposed the simple views and auxiliary views. The simple views are for users’ queries. The auxiliary views are used in order to maintain the simple views. They used the exhaustive incremental algorithm to analyze materialized views. Since the algorithm takes too much time, authors use the r-greedy algorithm to restrict the search space and later use heuristics algorithms to delete the unneeded search space. Liang [12] devised an algorithm for finding such an auxiliary view set by exploiting information shared among the auxiliary views and materialized views themselves. This can reduce the total size of auxiliary views.

Yang and Huang [13] figured out available space for materialization, which includes the minimal space and views as well as the maximal space and views. The