Statistical Sampling to Instantiate Materialized View Selection Problems in Data Warehouses

Mesbah U. Ahmed, University of Toledo, USA
Vikas Agrawal, Fayetteville State University, USA
Udayan Nandkeolyar, University of Toledo, USA
P. S. Sundararaghavan, University of Toledo, USA

ABSTRACT

In any online decision support system, the backbone is a data warehouse. In order to facilitate rapid response to complex business decision support queries, it is a common practice to materialize an appropriate set of the views at the data warehouse. However, it typically requires the solution of the Materialized View Selection (MVS) problem to select the right set of views to materialize in order to achieve a certain level of service given a limited amount of resource such as materialization time, storage space, or view maintenance time. Dynamic changes in the source data and the end users requirement necessitate rapid and repetitive instantiation and solution of the MVS problem. In an online decision support context, time is of the essence in finding acceptable solutions to this problem. In this chapter, we have used a novel approach to instantiate and solve four versions of the MVS problem using three sampling techniques and two databases. We compared these solutions with the optimal solutions corresponding to the actual problems. In our experimentation, we found that the sampling approach resulted in substantial savings in time while producing good solutions.

Keywords: data cube; data mart; data mining; data warehouse; materialized views; sampling

INTRODUCTION

The data warehouse is the heart of the architected environment, and is the foundation of all DSS processing (Inmon, 2002). Atypical data warehouse extracts the relevant information from many different operational databases into one centralized data repository to support business analysis activities and decision-making tasks (Haag, Cummings, & Phillips, 2006; Inmon, 2002). Recent extensions of this concept include building data warehouses to accommodate XML documents and facilitate easy querying of such data warehouses. Recent successful implementations of this concept may be found in Nassis, Rajagopalanpillai,
Dillon, and Rahayu (2005) and Rusu, Rahayu, and Taniar (2005). Data warehouses and OLAP tools are based on a multidimensional data model. This model views data in the form of a data cube. The complete discussion on dimensional modeling can be found in Kimball and Ross (2002). There are many important architectural issues concerning the efficient design of a data warehouse. Data cube design is one such important aspect of the data warehouse architecture. In simple terms, a data cube is a multi-dimensional construct of data that lets us explore and analyze a collection of data from many different perspectives (Han & Kamber, 2001; Kimball, Reeves, Ross, & Thorndhaite, 1998). The data in the data cube may be aggregated in one or more dimensions to generate a view (also known as a cuboid). If such a view is stored physically in a storage device, it is called a materialized view.

The problem of quick and easy access to the summarized data at the data warehouse may be alleviated by an efficient selection of a set of materialized views. The general problem of selecting an appropriate set of views to materialize is called the Materialized View Selection (MVS) problem. In these problems, researchers attempt to achieve an objective such as minimizing the query response time given a limited amount of resource such as materialization time, storage space, etc. (Gupta & Mumick, 2005) Literature has reported several variants of the MVS problem obtained by different combinations of objective functions, resource constraints, and problem solving methodologies (Gupta et al., 2005; Harinarayan, Rajaraman, & Ullman, 1996, 1999). In order to apply any problem solving technique, the first step is to know the parameters of the problem instance. To completely specify an MVS problem instance, one must know the number of rows present in each view and the weight associated with each view in the data cube. The evaluation of weights is based on managerial discretion. However, determining the number of rows in a view may be a time consuming process, which may run into hours or even days.

There are many online applications, such as implementing a data cube within a commercial package environment, which may need to determine the views to be materialized before it can interact with the user (Jacobson, 2000). The set of materialized views may have to be altered dynamically to accommodate changes in the frequency and importance of incoming queries as well as the changes in the size of the base cuboid. This demands a quick method to generate an appropriate instance of the MVS problem based on current needs. Typically, this implies running several queries on a data warehouse to count the number of rows in each view. In order to address the need for generating problem instances quickly, we tested three statistical sampling techniques to estimate the actual number of rows present in each view (Figure 1). We then instantiate a pair of MVS problems using the estimated and the actual number of rows in each view. Next, we demonstrate the efficacy of the methods by comparing the solutions obtained by solving the two problem instances. Our experiments reveal that the sampling approaches to instantiate the MVS problems is an effective alternative to complete enumeration.

Specific Contributions

The highlight of the contribution of this chapter is to apply the concept of sampling-based row estimation in a relational table to estimate the size of the views in a data cube, and use it to instantiate and
Electronic Payment Systems in Developing Countries for Improved Governance System
www.igi-global.com/chapter/electronic-payment-systems-developing-countries/29068?camid=4v1a

Rule-Based Data Mining Cache Replacement Strategy
Ramzi A. Haraty and Joe Zeitouny (2013). International Journal of Data Warehousing and Mining (pp. 56-69).
www.igi-global.com/article/rule-based-data-mining-cache/75615?camid=4v1a