View Materialization in a Data Cube:
Optimization Models and Heuristics

Vikas Agrawal, Fayetteville State University, USA

P. S. Sundararaghavan, The University of Toledo, USA

Mesbah U. Ahmed, The University of Toledo, USA

Udayan Nandkeolyar, The University of Toledo, USA

ABSTRACT

Data warehouse has become an integral part in developing a DSS in any organization. One of the key architectural issues concerning the efficient design of a data warehouse is to determine the "right" number of views to be materialized in order to minimize the query response time experienced by the decision makers in the organization. We consider a bottleneck objective in designing such a materialization scheme which has the effect of guaranteeing a certain level of performance. We examine linear integer programming formulations, and develop heuristics and report on the performance of these heuristics. We also evaluate heuristics reported in the literature for the view materialization problem with a simpler objective.

Keywords: data cube; data warehouse; view materialization

INTRODUCTION

In today's fast-paced, ever-changing and wants-driven economy, information is seen as a key business resource to gain competitive advantage (Haag, Cummings & McCubbrey, 2005). Effective use of this information requires good decision support systems. Most decision support systems require reliable and elaborate data backbone which needs to be converted into useful information. With the widespread availability and ever-decreasing cost of computers, telecommunications technologies, and Internet access, most businesses have collected a wealth of data. However, that is only the first and easy step. Many firms are becoming data rich but remain information and knowledge

poor (Gray & Watson, 1998; Grover, 1998; Han & Kamber, 2001; Nemati, Steiger, Iyer & Herschel, 2002). To alleviate this problem, many corporations have built, or are building, unified decision-support databases called *data warehouses* on which decision makers can carry out their analysis. A data warehouse is a very large data base that integrates information extracted from multiple, independent, heterogeneous data sources into one centralized data repository to support business analysis activities and decision-making tasks.

Business analysts run complex queries over this centralized data repository housed in a data warehouse to gain insights into the vast data and to mine for hidden knowledge. The key to gaining such insight is to design a decision support system which would get the right information to the right person and at the right time which will aid in making quality and often strategic decisions. In order to achieve this objective, design of the data warehouse architecture plays a pivotal role. There are many architectural issues concerning the efficient design of a data warehouse system. Lee, Kim, and Kim (2001) highlighted the importance of metadata for implementing data warehouse. They pointed out that integrating data warehouse with its metadata offers a new opportunity to create a more adaptive information system. Furtado (2006) proposed the concept of node partitioning, a method for parallelism, to improve the performance of a data warehouse system. Huang, Lin, and Deng (2005) proposed an intelligent cache mechanism for a data warehouse system in a mobile environment. They pointed out that because mobile devices can often be disconnected from the host server, and due to the low bandwidth of wireless networks, it is more efficient to store query results from a mobile device in the cache.

Data cube design is one such important aspect of the data warehouse architecture. Data cubes are constructs to store subsets of summarized data by some measures of interest for easy and quick access, and for timely and dynamic updates of these summarized data on an ongoing basis (Chun, Chung & Lee, 2004).

Accessing data from a data cube, if not materialized, can be a time consuming and resource intensive process. A data cube consists of many views with existing interrelated dependencies among themselves (such view is also known as a cuboid or a query). If such a view is stored, it is denoted as a materialized view. The problem of quick and easy access to the data cube may be alleviated by an efficient selection of a set of views to be materialized. Since not all views in a data cube may be materialized due to constraints imposed on the system, selecting the right set of views to materialize is an integral part of the design of data cube and its associated views. An efficient design will dramatically reduce the execution time of decision support queries and hence prove pivotal in delivering competitive advantage.

Many researchers have studied the problem of selecting the "right" set of views to be materialized in a data cube in order to minimize decision support query response time. The problem is generally described as the materialized view selection (MVS) problem, which has the objective of minimizing the access time subject to constraints on either the number of views that may be materialized or the storage space that may be used for materialization of views (Gupta & Mumick, 2005; Harinarayan, Rajaraman & Ullman, 1996, 1999). In this article we have worked on several variants of the MVS problems and have solved these optimally as well as using heuristics. Our specific contributions may be summarized as follows:

- We have presented a linear integer programming formulation for two versions of the MVS problem with a bottleneck objective, which minimizes the maximum weighted access time experienced by any class of users.
- We have developed heuristics for the above two problems and reported on their performance.
- We have also presented linear integer programming formulations for the general MVS problem reported in Harinarayan et

Related Content

Translating Advanced Integrity Checking Technology to SQL
Hendrik Decker (2002). *Database Integrity: Challenges and Solutions (pp. 203-249).*
www.igi-global.com/chapter/translating-advanced-integrity-checking-technology/7883?camid=4v1a

Data Warehousing Interoperability for the Extended Enterprise
Aristides Triantafillakis, Panagiotis Kanellis and Drakoulis Martakos (2004). *Journal of Database Management (pp. 73-84).*
www.igi-global.com/article/data-warehousing-interoperability-extended-enterprise/3317?camid=4v1a

An XML Multi-Tier Pattern Dissemination System
Ashraf Gaffar and Ahmed Seffah (2005). *Encyclopedia of Database Technologies and Applications (pp. 740-744).*
www.igi-global.com/chapter/xml-multi-tier-pattern-dissemination/11233?camid=4v1a

Information Quality: Critical Ingredient for National Security

www.igi-global.com/article/information-quality-critical-ingredient-national/3324?camid=4v1a