Chapter VII

Dynamic Multidimensional Data Cubes

Mirek Riedewald
University of California, Santa Barbara, USA

Divyakant Agrawal
University of California, Santa Barbara, USA

Amr El Abbadi
University of California, Santa Barbara, USA

ABSTRACT

Data cubes are ubiquitous tools in data warehousing, online analytical processing, and decision support applications. Based on a selection of pre-computed and materialized aggregate values, they can dramatically speed up aggregation and summarization over large data collections. Traditionally, the emphasis has been on lowering query costs with little regard to maintenance, i.e., update cost issues. We argue that current trends require data cubes to be not only query-efficient, but also dynamic at the same time, and we also show how this can be achieved. Several array-based techniques with different tradeoffs between query and update cost are discussed in detail. We also survey selected approaches for sparse data and the popular data cube operator, CUBE. Moreover, this work includes an overview of future trends and their impact on data cubes.
INTRODUCTION

For modern data warehouses it is common to manage Terabytes (Tb) of data. According to a survey by the Winter Corporation (2001), for instance, the decision support database of SBC, a telecom company, reaches a size of 10.5 Tb. Furthermore, 17% of the surveyed companies expect a tenfold increase of the size of their databases over the next three years. Already today retail companies like Kmart and Wal-Mart prepare their data warehouses to accommodate dozens of Terabytes of business data.

Human analysts cannot “digest” such large amounts of information at a detailed level. Instead they rely on the system to provide a summarized and task-specific view of some part of the data. Consequently, efficient summarization and aggregation of large amounts of data play a crucial role in On-Line Analytical Processing (OLAP) and Decision Support Systems (DSS). Such aggregated information is often maintained in data cubes which also provide an easy-to-understand conceptual model. Queries can be described as regions in the data cube. Pre-computed aggregate values that speed up query execution for group-bys, cross-tabs, and sub-totals can be easily included in the model (e.g., CUBE operator as proposed in Gray et al., 1997).

The primary goal of data warehouses is to support data analysis, i.e., queries. Update costs were often considered to be unimportant. Hence systems are typically oriented towards batch updates which are applied to the warehouse during times of low system load, e.g., overnight. The corresponding data cubes would be batch-updated as well, or even be re-computed from scratch. However, for very large data collections, incrementally maintainable dynamic data cubes are clearly preferable. There are several arguments why there should also be data cubes that support efficient single updates, not only in large batches.

Batch updates might effectively reduce the average cost per operation, but processing the whole batch makes the data cube inaccessible to analysts for a long period of time. Today businesses demand a 24/7 availability of their important data, hence finding a suitable time slot for a large batch update window becomes increasingly difficult. In most applications, especially decision support and stock trading, having the latest information incorporated as early as possible can give a decisive advantage over competitors that have to wait for an overnight batch update processing. Also, OLAP is inherently an interactive information exploration process that includes what-if scenarios. What-if analysis requires real-time and hypothetical data to be integrated with historical data for the purpose of instantaneous analysis and subsequent action.

One possible solution to address the above concerns is to use additional data structures that temporarily maintain updates until they are applied to the cube in a batch or lazy manner. In contrast, this chapter examines truly dynamic solutions that do not require batch processing and additional update structures. The presented techniques offer different tradeoffs between query, update, and storage costs,
Related Content

Theoretical vs. Practical Complexity: The Case of UML
[www.igi-global.com/article/theoretical-practical-complexity/3336?camid=4v1a](www.igi-global.com/article/theoretical-practical-complexity/3336?camid=4v1a)

From Data Quality to Big Data Quality
[www.igi-global.com/article/from-data-quality-to-big-data-quality/140546?camid=4v1a](www.igi-global.com/article/from-data-quality-to-big-data-quality/140546?camid=4v1a)

The Application-Based Domain Modeling Approach: Principles and Evaluation
[www.igi-global.com/chapter/application-based-domain-modeling-approach/39364?camid=4v1a](www.igi-global.com/chapter/application-based-domain-modeling-approach/39364?camid=4v1a)
Automatic Categorization of Web Database Query Results
www.igi-global.com/chapter/automatic-categorization-web-database-query/52295?camid=4v1a