Chapter XI
Accelerating Multi Dimensional Queries in Data Warehouses

Russel Pears
Auckland University of Technology, New Zealand

Bryan Houliston
Auckland University of Technology, New Zealand

ABSTRACT

Data Warehouses are widely used for supporting decision making. On Line Analytical Processing or OLAP is the main vehicle for querying data warehouses. OLAP operations commonly involve the computation of multidimensional aggregates. The major bottleneck in computing these aggregates is the large volume of data that needs to be processed which in turn leads to prohibitively expensive query execution times. On the other hand, Data Analysts are primarily concerned with discerning trends in the data and thus a system that provides approximate answers in a timely fashion would suit their requirements better.
In this chapter we present the Prime Factor scheme, a novel method for compressing data in a warehouse. Our data compression method is based on aggregating data on each dimension of the data warehouse. Extensive experimentation on both real-world and synthetic data have shown that it outperforms the Haar Wavelet scheme with respect to both decoding time and error rate, while maintaining comparable compression ratios (Pears and Houliston, 2007). One encouraging feature is the stability of the error rate when compared to the Haar Wavelet. Although Wavelets have been shown to be effective at compressing data, the approximate answers they provide varies widely, even for identical types of queries on nearly identical values in distinct parts of the data. This problem has been attributed to the thresholding technique used to reduce the size of the encoded data and is an integral part of the Wavelet compression scheme. In contrast the Prime Factor scheme does not rely on thresholding but keeps a smaller version of every data element from the original data and is thus able to achieve a much higher degree of error stability which is important from a Data Analysts point of view.

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

Data Warehouses are increasingly being used by decision makers to analyze trends in data (Cunningham, Song and Chen, 2006, Elmasri and Navathe, 2003). Thus a marketing analyst is able to track variation in sales income across dimensions such as time period, location, and product on their own or in combination with each other. This analysis requires the processing of multi-dimensional aggregates and group by operations against the underlying data warehouse. Due to the large volumes of data that need to be scanned from secondary storage, such queries, referred to as On Line Analytical Processing (OLAP) queries, can take from minutes to hours in large scale data warehouses (Elmasri, 2003, Oracle 9i).

The standard technique for improving query performance is to build aggregate tables that are targeted at known queries (Triantafillakis, Kanellis, and Martakos 2004; Elmasri, 2003). For example the identification of the top ten selling products can be speeded up by building a summary table that contains the total sales value (in dollar terms) for each of the products sorted in decreasing order of sales value. It would then be a simple matter of querying the summary table and retrieving the first ten rows. The main problem with this approach is the lack of flexibility. If the analyst now chooses to identify the bottom ten products an expensive re-sort would have to be performed to answer this new query. Worse still, if the information is to be tracked by sales location then the summary table would be of no value at all. This problem is symptomatic of a more general one where Database Systems which have been tuned for a particular access pattern perform poorly as changes to such patterns occur over a period of time. In their study (Zhen and Darmont, 2005) showed that database systems which have been optimized through clustering to suit particular query patterns rapidly degrade in performance when such query patterns change in nature.

The limitations in the above approach can be addressed by a data compression scheme that preserves the original structure of the data. The chapter is organized as follows. In the next section we review related work. The next section introduces the Prime Factor Compression (PFC) approach. We then present the algorithms required for encoding and decoding with the PFC approach. The On Line reconstruction of Queries is discussed thereafter. Implementation related issues are then discussed, followed by a performance evaluation of PFC and a comparison with the Haar Wavelet algorithm. We then discuss future trends in optimizing multi-dimensional queries in the light of the results of this research. We conclude with a summary of the main achievements of the research.

BACKGROUND

Previous research has tended to concentrate on computing exact answers to OLAP queries (Ho, and Agrawal, 1997, Wang 2002). Ho describes a method that pre-processes a data cube to give a prefix sum cube. The prefix sum cube is computed by applying the transformation: $P[A_i]=C[A_i]+P[A_{i-1}]$ along each dimension of the data cube, where P denotes the prefix sum cube, C the original data cube, A_i denotes an element in the cube, and i is an index in a range $1..D_i$ (D_i is the size of the dimension D_i). This means that the prefix cube requires the same storage space as the original data cube.

The above approach is efficient for low dimensional data cubes. For high dimensional environments, two major problems exist. Firstly, the number of accesses required is 2^d (Ho *et al*, 1997), which can be prohibitive for large values of d (where d denotes the number of dimensions). Secondly, the storage required to store the prefix sum cube can be excessive. In a typical OLAP environment the data tends to be massive and yet sparse at the same time. The degree of sparsity

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/accelerating-multi-dimensional-queries-data/4299

Related Content

Extended Action Rule Discovery Based on Single Classification Rules and Reducts
Zbigniew W. Rasand Elzbieta M. Wyrzykowska (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications (pp. 2313-2323).*
www.irma-international.org/chapter/extended-action-rule-discovery-based/8039

Common Sense Reasoning in Automated Database Design: An Empirical Test
Veda C. Storey, Robert C. Goldsteinand Jason Ding (2002). *Journal of Database Management (pp. 3-14).*
www.irma-international.org/article/common-sense-reasoning-automated-database/3272

The Evolution of the Meta-Data Concept: Dictionaries, Catalogs, and Repositories
Mark L. Gillensonand Raymond D. Frost (1993). *Journal of Database Management (pp. 17-26).*
www.irma-international.org/article/evolution-meta-data-concept/51122

Text Mining, Names and Security
Paul Thompson (2005). *Journal of Database Management (pp. 54-59).*
www.irma-international.org/article/text-mining-names-security/3326

Data Modeling: An Ontological Perspective of Pointers
Hock Chuan Chan, Chuan-Hoo Tanand Hock-Hai Teo (2014). *Journal of Database Management (pp. 17-37).*
www.irma-international.org/article/data-modeling/138624