Chapter 7.24
Optimization of Multidimensional Aggregates in Data Warehouses

Russel Pears
Auckland University of Technology, New Zealand

Bryan Houliston
Auckland University of Technology, New Zealand

ABSTRACT

The computation of multidimensional aggregates is a common operation in OLAP applications. The major bottleneck is the large volume of data that needs to be processed which 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 article 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. We used both real world and synthetic data to compare our scheme against the Haar wavelet and our experiments on range-sum queries show that it outperforms the latter scheme with respect to both decoding time and error rate, while maintaining comparable compression ratios. 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.

INTRODUCTION

Data warehouses are increasingly being used by decision makers to analyze trends in data (Cunningham, Song & Chen, 2006, Elmasri & 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 groups by operations against the underlying data warehouse. Due to the large volumes of data that needs 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 (Elmasri, 2003; Triantafillakis, Kanellis, & Martakos 2004). For example the identification of the top 10 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 10 rows. The main problem with this approach is the lack of flexibility. If the analyst now chooses to identify the bottom 10 products an expensive re-sort would have to be performed to answer this new query. Worst 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 & 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. For example, a 3-dimensional warehouse that tracks sales by time period, location and products can be compressed along all three dimensions and then stored in the form of "chunks" (Sarawagi & Stonebraker, 1994). Chunking is a technique that is used to partition a d-dimensional array into smaller d-dimensional units.

In principle, any data compression scheme can be applied on a data warehouse, but we were mindful of the fact that a high compression ratio would be needed to offset the potentially huge secondary storage access times. This effectively ruled out standard compression techniques such as Huffman coding (Cormack, 1985), LZW and its variants (Lempel & Ziv, 1977; Hunt 1998) Arithmetic Coding (Langdon, 1984). These schemes enable decoding to the original data with 100% accuracy, but suffer from modest compression ratios (Ng & Ravishankar, 1997). On the other hand, the trends analysis nature of decision making means that query results do not need to reflect 100% accuracy. For example, during a drill-down query sequence in ad-hoc data mining, initial queries in the sequence usually determine the truly interesting queries and regions of the database. Providing approximate, yet reasonably accurate answers to these initial queries gives users the ability to focus their explorations quickly and effectively, without consuming inordinate amounts of valuable system resources (Hellerstein, Haas, & Wang, 1997).

This means that lossy schemes which exhibit relatively high compression and near 100% accuracy would be the ideal solution to achieving acceptable performance for OLAP queries. This article investigates the performance of a novel scheme, called prime factor compression (PFC) and compares it against the well known wavelet approach (Chakrabarti & Garofalakis, 2000; Vitter & Wang, 1998; Vitter & Wang 1999).