Chapter 1

A Physical Design Strategy for Datasets with Multiple Dimensions

Marlene Goncalves
Universidad Simón Bolívar, Venezuela

José N. Mendoza
Universidad Simón Bolívar, Venezuela

ABSTRACT

Since the huge volume of data that is generated today, businesses need to have tools to efficiently manage such data. MonetDB is a column-oriented database management system which has shown to have better query processing time with respect to row-oriented systems. The main objective of this chapter is to propose a physical design strategy that improves query execution times in MonetDB. The proposed physical design strategy was empirically studied for 18 TPC-H queries. The experiments were conducted on the basis of cold cache. Each of the queries were executed first using the proposed physical design strategy in this work and then without any physical design. The reported results show that the runtimes using physical design strategy are better for all queries with a minimum percentage improvement of 29%. Also, they showed that the improvement was statistically significant by means of statistical tests.

INTRODUCTION

Relational databases have been essential for applications in the last decades. Likewise, there is the promise of giving to organizations, Data Base Management Systems (DBMS) that covers all application needs, e.g., systems for handling data related to sales, orders, human resources and analysis for decision-making.

With respect to the promise to cover all the needs of applications, it has not been fulfilled at all, since it has given greater prominence to transaction processing in order to support core applications of the companies. These applications are known as Online Transactional Processing (OLTP) and its main feature is the need to constantly perform read and write operations on the data. Therefore, DBMS as PostgreSQL, Oracle and MySQL are suitable for online transactions because they are optimized for the

DOI: 10.4018/978-1-5225-1776-4.ch001
insertion, modification and deletion of data stored in databases. On the other hand, applications that support the data analysis have been developed separately and not necessarily with relational DBMS, but with other tools that require a better performance when the user works with them. This application type is known as Online Analytical Processing (OLAP) and its operations are primarily read.

In this sense, there are two possibilities to cover the need of DBMS for OLAP applications. The first possibility is to utilize DBMS that are regularly used in OLTP applications. This first option may be a mistake because the way in which data is physically stored into databases for the support of OLTP and OLAP applications are totally different. Firstly, in OLTP databases, the tuples are arranged in rows and are stored in disk blocks. Moreover, OLAP databases are based on star schemes and compression of attributes by columns using dictionaries.

On the other hand, traditional DBMS that normally work with OLTP applications have seen the need to reinvent themselves to be able to deal with OLAP applications. For example, CitusDB is a PostgreSQL extension for working with a storage model by columns.

The second possibility is to look for other DBMS that are suited to the type of work in OLAP applications. In this context, it has lately increased interest in column oriented DBMS or Column-Stores because they have better performance on data managed by OLAP systems w.r.t. row oriented DBMS or Row-Stores. In fact, several works reveal that the performance of Column-Stores is better compared to Row-Stores. Particularly, if MonetDB is compared against CitusDB, the improvement is not significant and MonetDB continues having best runtimes in most of the queries.

Since the second option is the best, in this chapter, we study the physical design of MonetDB. MonetDB was selected because it is a DBMS open and pioneered in the use of columns as physical structure for storing data. In addition, MonetDB is not obsolete, far from it, has a very active community and participate in major projects, especially in Europe (MonetDB, 2015).

Finally, experimental studies have been conducted on MonetDB using cold cache without any physical design. Thus, in this chapter we propose a physical design strategy in order to improve the performance on cold cache taking into account the specific characteristics of MonetDB.

This chapter is comprised of six sections in addition to Section 1 that introduces the problem. Section 2 describes column-oriented database management systems, execution plans, the physical design, and the MonetDB execution model. Section 3 introduces the strategy for physical design in MonetDB. In Section 4, the performance of the proposed strategy will be empirically evaluated. Finally, the future research directions and conclusion of this chapter will be pointed out in the Sections 5 and 6, respectively.

BACKGROUND

This section describes column-oriented database management systems, execution plans, the physical design, and the MonetDB execution model.

Column Stores

Column-oriented database management systems, also known as column stores, are an alternative to row-oriented database management systems (row stores) and have gained importance in recent years by the rise in analytical applications where queries are based on processing large amounts of data and calculating aggregate functions such as averages, sums and maxima.