Chapter 8

Index Structures for Data Warehousing and Big Data Analytics

Veit Köppen  
Otto von Guericke University Magdeburg, Germany

Martin Schäler  
Karlsruhe Institute of Technology, Germany

David Broneske  
Otto von Guericke University Magdeburg, Germany

ABSTRACT

With the ongoing increasing amount of data, these data have to be processed to gain new insights. Data mining techniques and user-driven OLAP are used to identify patterns or rules. Typical OLAP queries require database operations such as selections on ranges or projections. Similarly, data mining techniques require efficient support of these operations. One particularly challenging, yet important property, that an efficient data access has to support is multi-dimensionality. New techniques have been developed taking advantage of novel hardware environments including SIMD or main-memory usage. This includes sequential data access methods such SIMD, BitWeaving, or Column Imprints. New data structures have been also developed, including Sorted Projections or Elf, to address the features of modern hardware and multi-dimensional data access. In the context of multidimensional data access, the influence of modern hardware, including main-memory data access and SIMD instructions lead to new data access techniques. This chapter gives an overview on existing techniques and open potentials.

SCOPE OF THIS ARTICLE

With the ongoing increasing amount of data, these data have to be processed to gain new insights. Therefore, a solution is necessary to store and query multidimensional data efficiently. In relational databases, index structures like the B-tree (Bayer et al, 1972) have improved the data access drastically. In
a multidimensional domain, which is common for data warehouses as well as big data applications, these index structures have either a limitation to support a specialized scenario or cannot scale sufficiently. Additionally, the authors in (Berchtold et al, 1998) address the problem, called curse of dimensionality, which is an important insight for multidimensional data access. As a result many approaches and most present-day database systems rely on optimized sequential scans taking advantage of the capability of modern hardware. Moreover, this leads to the optimization of sequential scans for multidimensional data to support OLAP (Online Analytical Processing) analyses.

Predicate evaluation is a challenging task in the OLAP domain (Johnson et al, 2008) required amongst others for slice operations on the data cube. More generally, to extract important data for further analysis, fact and dimension tables are passed through several selection predicates involving several dimension attributes. Shrinking the processed data amount as soon as possible has become an important task, when all data have to fit into main memory. So, the I/O bottleneck is eliminated and a full table scan becomes less expensive. Therefore, we focus on how all approaches support this operation.

In case all data sets are available in main memory (e.g., in a main-memory database system (Boncz et al, 2008; Kemper et al, 2001; Plattner, 2009), the selectivity threshold for using an index structure instead of an optimized full table scan is even smaller than for disk-based database systems. In a recent study (Dasa et al, 2015), the authors propose to use an index structure for very low selectivities only, such as values below 2%. Hence, most OLAP queries would never use an index structure to evaluate the selection predicates. However, an interesting fact neglected by this approach is that the accumulated selectivity of several selection predicates is favored if this exploits the relation between all selection predicates of the query. Consequently, when considering multi-dimensional queries, we achieve the selectivity required to use an index structure instead of an accelerated scan.

Another fact for indexes that completely fit into the main memory is, that the structures should consider two aspects carefully: an optimization according to the restrictions of CPU caches and the opportunity of multi-core parallelism (Faebert al, 2016). New main memory database systems, such as C-Store, HyPer, and SAP HANA, do not use a page-based indirection, but use the provided storage efficiently. Additionally, a pointer directly addresses the record and therefore, identifiers are omitted. Consequently, we illustrate how all approaches make efficient use of present-day hardware.

Altogether, there is a wide range of different approaches, each having their own advantages and limitations. In this paper, we aim at giving an overview on how such approaches work and how they are related to each other. This paper particularly addresses readers that start getting into contact with the domain of efficient data access methods on modern hardware. To this end, we introduce different approaches in the next section using one example data set to illustrate differences between the approaches. After that, we show how an exemplary evaluation of different approaches looks like and we also give a first indication of strengths and weaknesses of either approach.

TECHNIQUES FOR MULTIDIMENSIONAL DATA ACCESS

In principle, there are three groups of approaches. The first group follows a direct sequential access pattern, most database systems rely on. Therefore, this can be considered as industry standard. The second group consists of state-of-the-art approaches usually aiming at highly compressing the data set and making use of modern CPU capabilities, such as SIMD, to process multiple data items in one step.