

# **Scalable Bitmap Index**

Department of Computer Science and Technology

马玉坤

1150310618

2017/11/18

# 1 Abstract

Bitmap Index is presented by P. O’Neil in 1987. It was first applied in a commercial database. (Spiegler & Maayan 1985) In the application of database, either for scientific purposes or for commercial purposes, bitmap indexes are widely used.

The original bitmap indexes uses *Bit Vector* to indicate the indexed attributes in the database. For example, in Table 1

Table 1: An Ordinary Table

name	math score	chinese score
Alice	A	C
Bob	B	A
Dean	A	D

# 2 Introduction

Bitmap indexing has been touted as a promising approach for processing complex adhoc queries in read-mostly environments, like those of decision support systems. Nevertheless, only few possible bitmap schemes have been proposed in the past and very little is known about the space-time tradeoff that they offer. Within that space, I identify (analytically or experimentally) the following interesting points: (1) the time-optimal bitmap index; (2) the space-optimal bitmap index; (3) the bitmap index with the op-

timal space-time tradeoff (knee); and (4) the time-optimal bitmap index under a given disk-space constraint. Finally, I examined the impact of bitmap compression and bitmap buffering on the space-time tradeoffs among those indexes.

Developing efficient mechanisms for set indexing and searching can serve as the basis for other improvements. The ability to retrieve sets based on their subset properties can lead to the improvement of many other algorithms that depend on database set processing. As an example, let us consider a system for automated web site personalization and recommendation. Such systems are commonly used in various e-commerce sites and on-line stores. The system keeps information about all products purchased by the customers. Based on these data the system discovers patterns of customer behavior, e.g., in the form of characteristic sets of products that are frequently purchased together. When a new potential customer browses the on-line catalog of available products, the system can anticipate which product groups are interesting to a customer by analyzing search criteria issued by a customer, checking visited pages and inspecting the history of purchases made by that customer, if the history is available. By issuing subset queries to find patterns relevant to the given customer the system can not only dynamically propose interesting products, preferably at a discount, but it can also tailor the web site navigation structure to satisfy specific customers' requirements.

Unfortunately, currently available database systems do not provide mechanisms to achieve satisfactory response times for subset queries and database sizes in question.

The ability to efficiently perform set-oriented queries in large data volumes could greatly enhance web-based applications. This functionality is impatiently anticipated by many potential users.

The most common class of queries which often appear in terms of set-valued attributes are subset queries that look for sets that entirely contain a given subset of elements. Depending on the domain of use subset queries can be used to find customers who bought specific products, to discover users who visited a set of related web pages, to identify emails concerning a given matter based on a set of key words, and so on. In our study and experiments we concentrate on subset queries as they are by far the most common and useful class of queries regarding set-valued attributes. We recognize other classes of queries as well and we describe them later.

Typical solution to speed up queries in database systems is to use indexes. Unfortunately, even though the SQL3 supports set-values attributes and most commercial database management systems offer such attributes, no commercial DBMS provides to date indexes on set-valued attributes. The development of an index for set-valued attributes would be very useful and would improve significantly the performance of all applications which depend on set processing in the database.

### 3 Scalable Bitmap Index

Scalable Bitmap Index is based on signature index framework. It employs the idea of exact set element representation and uses hierarchical structure to compact resulting signature and reduce its sparseness. The index on a given attribute consists of a set of index keys, each representing a single set. Every index key comprises a very long signature divided into  $n$ -bit chunks.

Let us now discuss briefly the physical implementation of the hierarchical bitmap index. An example of the entire index is depicted in Fig. 2. Every index key is stored as a linked list of all index key nodes (both internal and leaf nodes) except the index key root. Those linked lists are stored physically in a set of files, where each file stores all index keys containing an equal number of nodes.

(?)

### References

Spiegler, I. & Maayan, R. (1985), 'Storage and retrieval considerations of binary data bases', *Information processing & management* **21**(3), 233–254.