Scalable Bitmap Index

Department of Computer Science and Technology

马玉坤

1150310618

2017/11/18

Abstract

Bitmap Index is a widely used data structure in the database. It has a very low complexity to program. There are many kinds of bitmap indexes presented by different people. This article mainly focuses on the time and space analysis of these kinds of bitmap indexes and try to offer suggestions about how to choose appropriate implementations of bitmap index. Finally this article will try to show the goodness of the combination of different implementations.

1 Background

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, in the column named "math score", the Bit Vector for value "A" is *101*, indicating that Alice has an **A**, Bob does not have an A and Dean has an **A**.

Table 1: An Ordinary Table

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

When we use the Bit Vectors of different attributes to do logical math, we can get different results to get the answers for so many kinds of queries. Take this as another example: We want to know the students who has an **A** in math and a **C** in Chinese. Then we can use the result of the *bitwise and* of the Bit Vector of “math A” (which is **101**) and the Bit Vector of “Chinese C” (which is **100**). As the calculation below, the result will be **100** which indicates that Alice is the only one who gets an **A** in math and a **C** in Chinese.

$$101 \tag{1}$$

$$\&100 \tag{2}$$

$$=100 \tag{3}$$

2 Introduction

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.