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

1.1 Bitmap Index

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.

$$101 \tag{1}$$

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

$$=100 \tag{3}$$

The example above is good for showing the basic idea of bitmap indexing. But in the industrial application, the trivial implementation of the Bitmap Index is not often used because of its high complexity of time and space. So there are tons of different implementations of bitmap indexes that can lower the time complexity and the space complexity.

1.2 Time Complexity and Space Complexity

In the analysis of algorithms, time complexity and space complexity are two methods to measure the efficiency of an algorithm.

The time complexity can measure or estimate the time consumed for running one algorithm. Not specifically, a bigger time complexity usually means more time consumed for the same input. While the space complexity can measure or estimate the space consumed for running one algorithm. Similaryly and not specifically, a bigger space complexity usually means more space consumed for the same input.

In section 2 these implementations of Bitmap Indexes will be revealed. And in section 3 these implementations will be discussed and compared with each other to show their differences in time complexity and space complexity. Finally we will give advice on how to choose the appropriate implementations of Bitmap Index in section 4.

2 Different Implementations of Bitmap Indexes

Here, we focus on three kinds of bitmap indexes. They are value-based bitmap index, range-based bitmap index and bit-transposition bitmap index, respectively.

2.1 Value-Based Bitmap Index

2.2 Range-Based Bitmap Index

2.3 Bit-Transposition Bitmap Index

(Wong et al. 1986)

3 Comparisons of Implementations of Bitmap Indexes

4 Choosing the Right Implementations

5 Conclusion

(?)

References

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

Wong, H. K. T., Li, J. Z., Olken, F., Rotem, D. & Wong, L. (1986), ‘Bit transposition for very large scientific and statistical databases’, *Algorithmica* **1**(1), 289–309.
URL: *https://doi.org/10.1007/BF01840449*