2010

Reducing Database Bitmap Index Size Using Variable Length Compression

Fabian J. Corrales
University of Puget Sound

Reducing Database Bitmap Index Size Using Variable Length Compression

Fabian Corrales

UNIVERSITY of PUGET SOUND
Est. 1888

Utilizing the variations in bitmap columns to achieve greater compression.

Introduction

Modern research endeavors generate large amounts of data. One technique to manage such repositories is to create a bitmap index. This type of index is a coarse binary representation of the data, which can be quickly queried using logical operations (e.g. AND). The data in the DB is transformed into a bitmap index through a process called binning, in which data is put into categorized bit vectors; 1 if the data falls into the category, 0 if not. Unfortunately these indexes can become quite large and are usually compressed. Two commonly used compression schemes are Word Aligned Hybrid(WAH) [1] and Byte-aligned Bitmap Code(BBC) [2]. New research has shown Gray code [3] ordering to be effective in increasing compression. We exploit a specific aspect of Gray code ordering to create another compression scheme which can achieve better compression than BBC, with a minimal loss in query efficiency when compared to WAH: the Variable Length Compression (VLC) scheme.

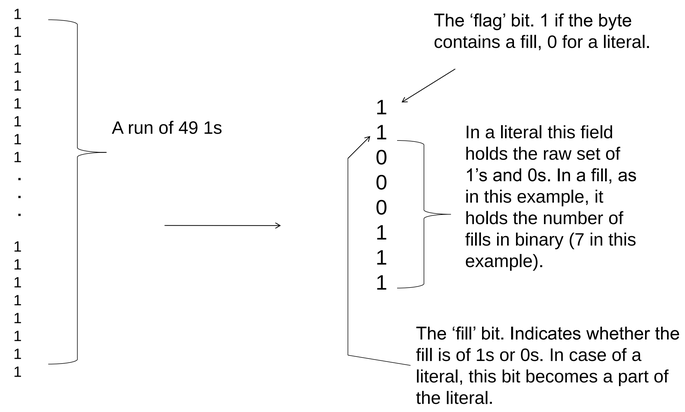

Fig. 1 An example of BBC type compression. This example shows a run with a fill bit of 1, 7 runs long (runs are 7 bits long, so it represents 49 consecutive 1s).

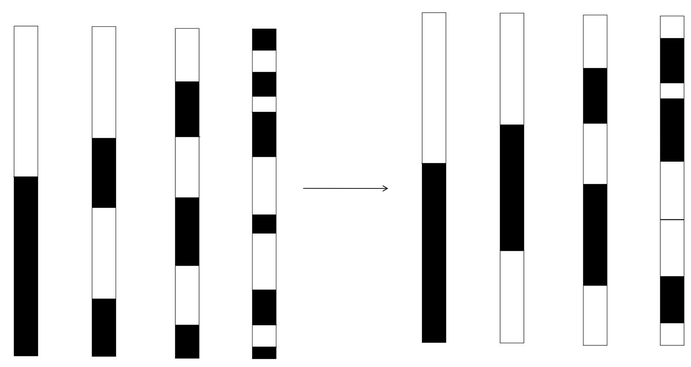

Fig. 2 Gray code ordering reorders data to allow for longer runs. The example on the left is a set of bitmaps in a lexicographical order, the example on the right is bitmaps after Gray code ordering has been applied. White blocks represent runs of 0s, and black blocks represent runs of 1s.

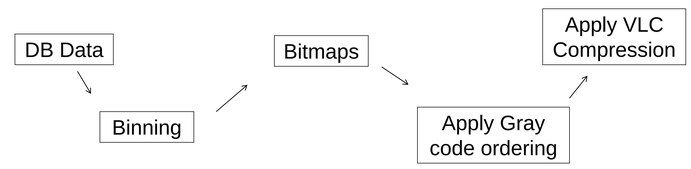

Fig. 4 Database data is first binned, which groups data into categories rather than precise values. Bins are then transformed into bitmaps, which are reordered using Gray code, and then compressed.

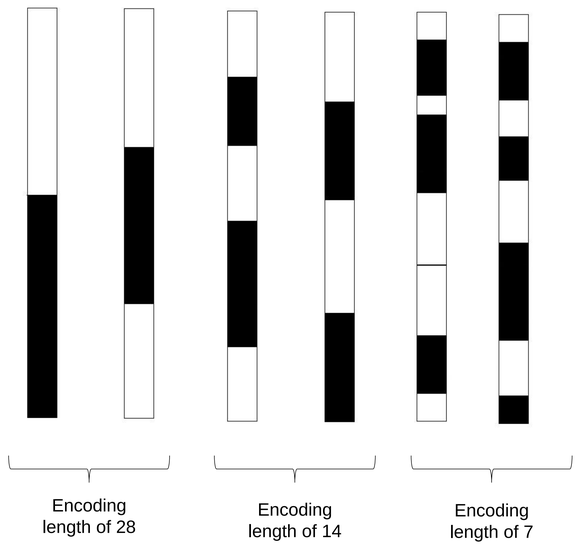

Fig. 3 The first few columns are compressed using the largest encoding length, 28. As the run lengths grow smaller, our encoding lengths will also diminish to maintain good compression.

Table 1
Compression sizes

Dataset	VLC	BBC	WAH	Uncomp.
HEP	1.1 MB	1.4 MB	2.1 MB	272.0 MB
STOCK	622 KB	621 KB	605 KB	6.7 MB
LANDSAT	17.8 MB	17.8 MB	26.7 MB	238.0 MB
HISTOBIG	573 KB	576 KB	1.0 MB	21.4 MB
UNIFORM	19 KB	30 KB	32 KB	10 .2 MB

Table 2
Average query times in milliseconds

Dataset	VLC	BBC	WAH
HEP	194	232	226
STOCK	32	29	18
LANDSAT	436	415	473
HISTOBIG	140	145	137
UNIFORM	11	15	10

BBC and WAH Compression

WAH and BBC are lossless run length compression schemes, their results can be queried without first being decompressed. WAH uses a 32 bit encoding length which is the size of a typical word in a computer, and BBC uses an 8 bit encoding length which is the size of a single byte of data(Fig. 1). BBC can achieve better compression than WAH since it can compress shorter runs. WAH can achieve better query times since a memory access returns a word, it requires no additional parsing.

Gray Code

Gray code ordering reorders data so that successive values differ only in one bit (Fig. 2). This reordering can be applied to bitmaps because row order does not matter in a relational database. This ordering is employed as a preprocessing technique used to achieve longer runs, however, the run lengths of a Gray code ordered bitmap diminish with more columns (Fig. 2). We mitigate this effect with our new compression scheme.

VLC Compression

VLC uses three different encoding lengths: 28, 14 and 7(Fig. 3). It will determine which one of these three lengths will be optimal for a given column and then compress using that length. These lengths are used because they allow each of the larger encoding lengths to be broken down and compared with the smaller encoding length for querying. As Gray coded run lengths deteriorate, our scheme will change encoding lengths accordingly to maintain efficiency.

Results

VLC had on average 11% better compression than BBC, and 32% better than WAH. VLC’s query times were 6% faster than BBC but 6% slower than WAH. Tables 1 and 2 detail our findings for several real world data sets.

Future Work

In the future we plan to modify VLC to be able to encode using any length, to further increase compression efficiency. Other areas that can be explored are parallelization and different row ordering techniques.

Acknowledgements

This research was made possible by the University of Puget Sound McCormick research grant and aided by my advisor Jason Sawin, professor Brad Richards, my labmate John Granville, and viewers like you.

References

1) G. Antonshenkov. Byte-aligned bitmap compression. In *Data compression Conference,* 1995. Oracle Corp.
2) K. Wu, E. J. Otoo, and A. Shoshani. Compressing bitmap indexes for faster search operations. In *SSDBM*, pages 99-108, 2002.
3) A. Pinar, T. Tao and H. Ferhatosmanoglu. Compressing Bitmap Indices by Data Reorganization. In *ICDE*, pages 310-321, 2005.