Reorganizing Bitmap Tables for Improved Compression*

Ali Pınar† Hakan Ferhatosmanoglu‡

Advances in technology have enabled the production of massive volumes of data through observations and simulations in many scientific applications such as biology, high-energy physics, climate modeling, and astrophysics. These new data sets and the associated queries are significantly different from those of the traditional database systems, most importantly due to their enormous size and high-dimensionality (more than 500 attributes in high-energy physics experiments). These new data sets and the associated queries pose a new challenge for efficient storage and retrieval of data and require novel indexing structures and algorithms.

Most of the scientific databases of practical interest are read-only, i.e., large volumes of data are stored once and never updated. Further use of the data is typically by means of selection queries. Various types of queries, such as partial match and range queries, are executed on these large data sets to retrieve useful information for scientific discovery. As an example, a user can pose a range query to retrieve all events with energy less than 15 GeV, and the number of particles less than 13. When the data are large and read-only, as in the case of scientific databases, indexing technologies are well-known to significantly improve the performance of query and data analysis, thus developing index structures tailored for scientific data is crucial to effectively explore such data. Due to the scale and high dimensionality of these databases, simple extensions of traditional indexing strategies are inadequate: R-trees and its variants are well-known to lose effectiveness for high dimensions; hashing-based indices lack storage efficiency; and transformation based approaches are not effective for partial match and range queries. Furthermore, most of the indexing approaches do not focus on the size of the index structure itself. However, due to the huge data volume in a typical scientific database, the size of the indexing structure becomes as important as other parameters and must be taken into account.

Focusing on the major characteristics of scientific data, such as being read-only, having special access patterns and numerical attributes, researchers have managed to develop indexing techniques that are feasible for high dimensional scientific databases. Bitmap indexing, which has been effectively utilized in many major commercial database systems has also been the most popular approach for scientific databases. Several techniques have

*Supported by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract DE-AC03-76SF00098.

†Corresponding author. Computational Research Division, Lawrence Berkeley National Lab, 1 Cyclotron Road, Berkeley, CA 94720. Email: apinar@lbl.gov

‡Department of Computer Science and Engineering, The Ohio State University,395 Dreese Lab, 2015 Neil Ave Columbus, OH 43210. Email: hakan@cse.ohio-state.edu

been proposed exploiting the bitmap indexing approach for scientific data. The general idea is to organize the data as a two dimensional table. Events are stored row-wise as tuples. Every attribute is partitioned to several bins, which form the columns of the table. A table entry is 1, if the tuple of this row is in the bin of the column, and 0 otherwise. Thus, the index table is a 0-1 table. This table needs to be compressed to be effective on a large database. General purpose text compression techniques are clearly not suitable for this purpose since they significantly reduce the efficiency of queries. Specialized bitmap compression schemes have been proposed to overcome this problem. The two most effective schemes in the literature are Byte-aligned Bitmap Code (BBC) and Word-Aligned Hybrid Code (WAH). Both of these schemes, like many others, are based on run-length encoding, i.e., they both replace repeated runs of 0's or 1's in the columns by a single instance of the symbol and a run count. These methods not only compress the data but also enable fast bitwise logical operations, which translate to faster query processing.

Run-length encoding and its variants exploit uniform segments of a sequence, thus their performances depend directly on the presence of such uniform segments. Their effectiveness varies for different organizations of the database tuples, since ordering of tuples affect uniform segments in the columns. In this work, we study how to reorder tuples of a database to achieve higher compression rates. Our techniques are used as a preprocessing step before compression, only to improve the performance, without affecting algorithms used for compression and querying. We state this tuple reordering problem as a combinatorial optimization problem, and propose heuristics for effective solutions for this NP-Complete problem. We show a reduction of the tuple reordering problem to the traveling salesperson problem, which is a well-studied combinatorial optimization problem. However, given the enormous sizes of the databases, we are only restricted to memory and time efficient heuristics, which takes away the applicability of most frequently used techniques such as simulated annealing. In this work, we propose Gray code sorting to order the rows of a bitmap table for larger segments of uniform 1's. Our algorithm is linear, in the size of the database, and an in-place algorithm, which means it does not require any auxiliary memory allocation. Theoretically, we prove that our algorithm is optimal, when all cells of a bitmap table are full. In practice, our experiments on scientific data showed significant improvements in compression rates. In many instances, compressed file size for the reordered file less than half the compressed size of the original file. We have also observed a 9.60 times reduction in compressed file size on data set `HEP3`, bitmap table for which has 110 columns and 2,000,000 rows.