Course Project #4: Implementation of Dictionary Codec

Due date: November 11

1. Introduction

The objective of this project is to implement a dictionary codec. As discussed in class, dictionary encoding is being widely used in real-world data analytics systems to compress data with relatively low cardinality and speed up search/scan operations. In essence, a dictionary encoder scans the to-be-compressed data to build the dictionary that consists of all the unique data items and replaces each data item with its dictionary ID. To accelerate dictionary look-up, one may use certain indexing data structure such as hash-table or B-tree to better manage the dictionary. In addition to reducing the data footprint, dictionary encoding makes it possible to apply SIMD instructions to significantly speed up the search/scan operations.

2. Requirement

Your implementation should support the following operations:

(1) Encoding: given a file consisting of raw column data, carry out dictionary encoding and generate an encoded column file consisting of both dictionary and encoded data column. Your code must support multi-threaded implementation of dictionary encoding. For the encoded data column, you must use integer compression to further reduce the footprint of the encoded data column on SSD (you may Google existing open-source integer compression libraries).

(2) Query: enable users to query an existing encoded column file. Your implementation should allow users to (i) check whether one data item exists in the column, if it exists, return the indices of all the matching entries in the column; (ii) given a prefix, search and return all the unique matching data and their indices. Your implementation must support the use of SIMD instructions to speed up the search/scan.

(3) Your code should also support the vanilla column search/scan (i.e., without using dictionary encoding), which will be used as a baseline for the speed performance comparison

In addition to the source code, your Github site should contain

(1) Readme that clearly explains the structure/usage of your code, and how your code utilizes multi-threading and SIMD to speed up

(2) Comprehensive experimental results that show the performance of your implementation (both encoding performance and query performance). When measuring the performance, do not count the time of loading file to memory and writing file to SSD. The performance results must contain: (i) encoding speed performance under different number of threads, (ii) single data item search speed performance of your vanilla baseline, dictionary without using SIMD, and dictionary with SIMD, (iii) prefix scan speed performance of your vanilla baseline, dictionary without using SIMD, and dictionary with SIMD.

(3) Show the effectiveness of dictionary coding on reducing the column data size both in memory and on SSD (use integer compression to further reduce on-SSD data footprint)

(4) Analysis and conclusion

Note: Use the raw column data file at the following address for your experiments and evaluation
https://drive.google.com/file/d/195XTg8HWDTILc1JlsGX6_jJ5PUhi9KDG/view?usp=sharing