SBoost: Boosting Data Filtering in Columnar Store

Hao Jiang, Aaron J. Elmore

Abstract
Lightweight encodings are widely used in columnar stores. They are fast and allow in-situ queries [2]
- Bit-Packed Encoding [3]
- Delta Encoding [5]
- Run-Length Encoding
- Dictionary Encoding

We build SBoost, a columnar store based on Parquet [1] that uses innovative SIMD algorithms to speed up data filtering on encoded data by up to 100x.

Background

Columnar Store

Store similar data types consecutively

SMD

Instruction Pool

Process multiple data in one instruction

SBoost Architecture

Native Memory

JVM

SIMD Algorithm

filter

Bitmap

Decoded Data

Parquet File

Row Group 1

Col 1

Page 1

Page 2

Page K

Disk

Row Group 2

Mapped Page

Reference

2. D. Abadi, SIGMOD’06, 671-682
3. T. Willhalm, VLDB’ 09, 385-394
4. C. Binnig, SIGMOD’09, 283-296

Algorithms

Data Filter on Bit-Packed Encoded Data

Idea: store the comparison result in the most significant bit

Equality

Use \texttt{xor} to check equality

\[ x_1 \text{xor} x_2 = 0 \iff x_1 = x_2 \]

Most significant bit

\[ 1 \text{xor} 0 = 0 \]

\[ 0 \text{xor} 1 = 1 \]

\[ a \text{xor} b = c \]

\[ c \text{ has carry} \]

\[ x < a \iff \begin{cases} 1. & x = 0 \text{ and } a = 1 \\ 2. & x = 1 \text{ and } a = 0 \end{cases} \]

Data Filter on Dictionary Encoded Data

Use Order-Preserving Dictionary [4] to rewrite the query, then use bit-packed algorithm

Data Filter on Delta Encoded Data

Delta encoding stores delta between adjacent numbers. To decode, we need to compute the cumulative sum on an entry list.

SIMD \texttt{hadd} performs multiple add in parallel

Use \texttt{hadd} to compute cumulative sum

\[ b = \text{shift}256(b, bp) \]

\[ x_1 = \text{hadd}(b, bp) \]

\[ x_2 = \text{permute}(s1) \]

\[ x_3 = \text{hadd}(x_1, s2) \]

\[ x_4 = \text{permute}(s3) \]

result = add32(x_3, s4)

\[ decoder(\text{result}) \]

Conclusion

- SBoost provides innovative SIMD algorithms on Lightweight Encodings
- Speed up Data Filtering on Columnar Stores
- Fully Compatible with existing Datasets
- Next Step: Explore more encoding schemes (on Double Data Type)