Blitzcrank: Fast Semantic Compression for In-Memory Online Transaction Processing

Yiming Qiao, Yihan Gao, Huanchen Zhang

A stand-alone C++ Library for Compressing Row-store, In-memory Databases

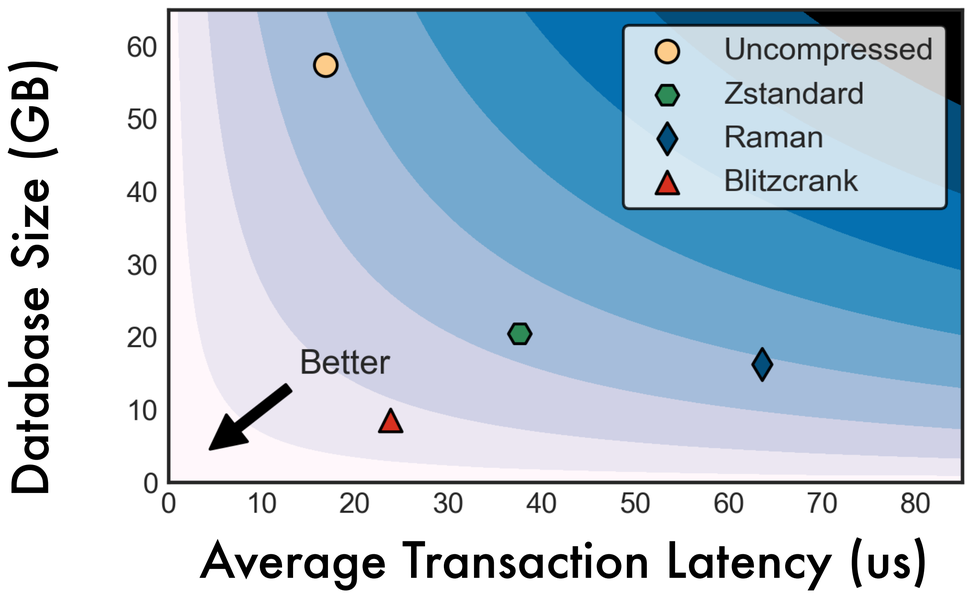
id	gender	balance
3	M	19.8

→ Mem: < 10 MB → **Blitzcrank** → Cpr. Factor: **5 – 10** → Compressed Tuple

Compression Latency: 2 us/tuple Decompression Latency: 1 us/tuple

Motivation for Row-store Compression

1. In-memory Databases are faster than on-disk databases, but the memory is a limited resource.
2. Block Compression has high random access latency.
3. Current Entropy Coding is not fast enough



Compression = Modeling (Semantic Modeling) + Encoding (Delayed Coding)

id	gender	name
1	F	Taylor
2	M	Alex
3	F	Alex
4	F	Taylor
5	F	Taylor

Semantic Model for name
P (name = Alex) = 0.4
P (name = Taylor) = 0.6

Semantic Model for gender
P (gender = F | name = Taylor) = 1
P (gender = F | name = Alex) = 0.5

Learned Semantic Models

Each Semantic Model has two functions:
Translate: Value → Probability Interval
Inv-Translate: Code → Value

For example,
Alex → [0, 0.4), Taylor → [0.4, 1)
If name = Taylor, F → [0, 1)
If name = Alex, F → [0, 0.5), M → [0.5, 1)

The Proposed Delayed Coding generates Fixed-length Codes (4 bits)

A tuple to Compress →

~~[1/16, 1/4)~~ ~~[1/8, 1/2)~~ ~~[3/16, 1/4)~~

[1, 4) [2, 8) [3, 4) 4-bit Integer Probability

Entropy: 2.4 bits 1.4 bits 4 bits

For an interval [L, R), any 4-bit integer in this interval can be used as the code, so…

Step 1:

Step 2: (1.4 bits, 2.6 bits)

$N = a_3b_6 = a \times 6 + b$

Resulting Codes: $(.0001\ 0101)_2$

It uses 8 bits to represent three intervals, with a total entropy of 7.8 bits

qiaoym21@mails.tsinghua.edu.cn github.com/YimingQiao/Blitzcrank