Elf: Erasing-Based Lossless Floating-Point Compression @VLDB 2023

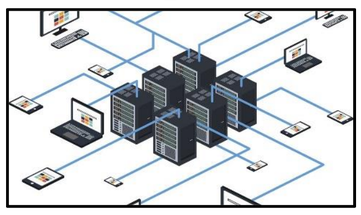
Ruiyuan Li [1], Zheng Li [1], Yi Wu [1], Chao Chen [1], Yu Zheng [2]

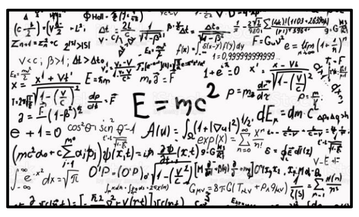
[1] *Chongqing University, China* [2] *JD iCity, JD Technology, China*

Background

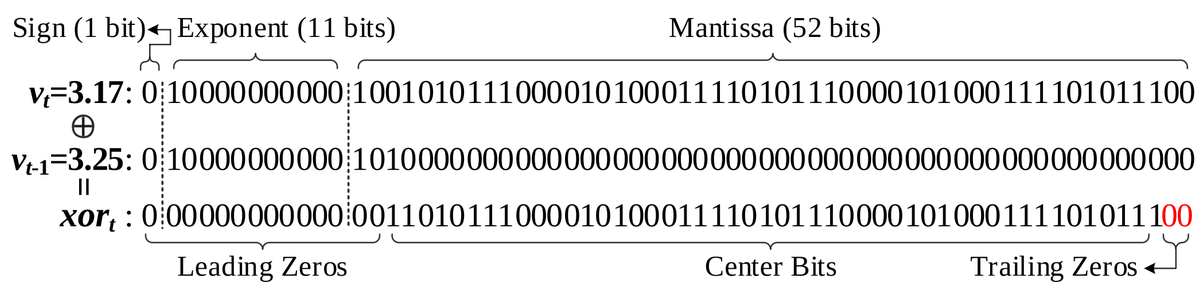
- **Lossless Floating-Point Compression is Rather Vital**

Network Transmission

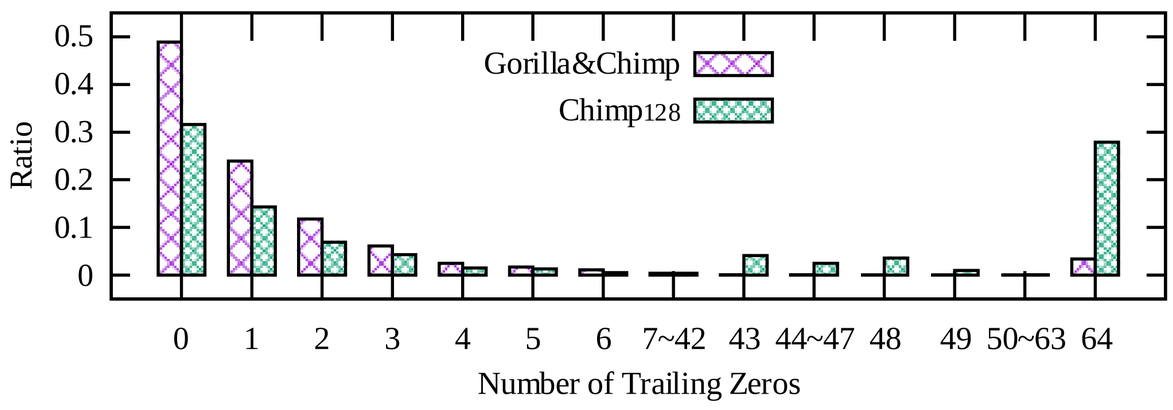

Data Management


Scientific Calculation

- **XORing-Based Time Series Compression Method**

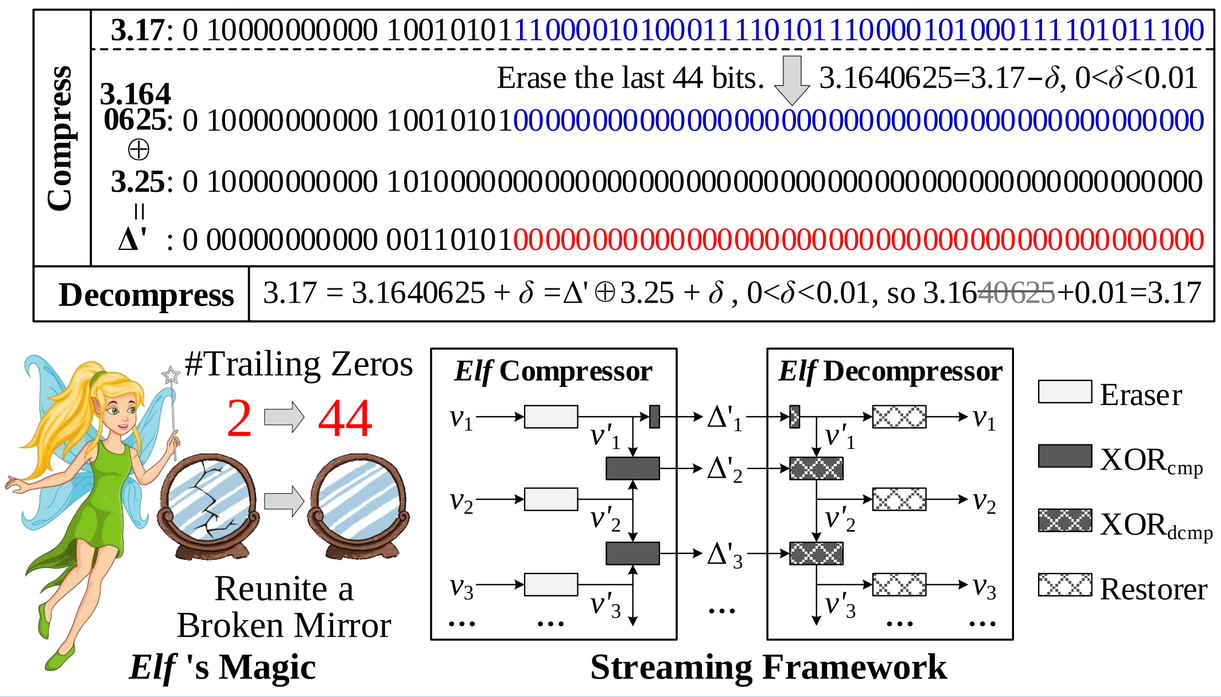


- **Few Trailing Zeros Lead to Unsatisfactory Effects**



Intuition

- **Erasing Last Few Bits (i.e., Setting Zeros) of Mantissa**
- **Restoring the Mantissa with a Leave-Out Operation**

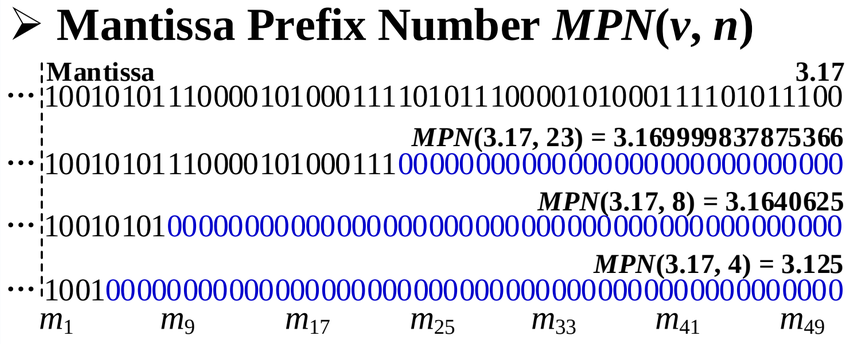


Challenges

- How to quickly determine the erased mantissa bits?
- How to compactly encode the erased floating-point data?
- How to losslessly restore the original floating-point data?

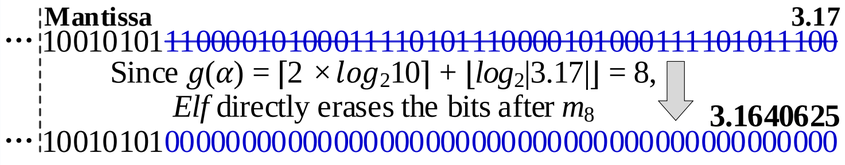
Determine Erased Bits

- **Decimal Place Count $\alpha = DP(v)$**
  - $DP(3.17) = 2$, $DP(-0.0317) = 4$, $DP(317.0) = 1$
- **Mantissa Prefix Number $MPN(v, n)$**

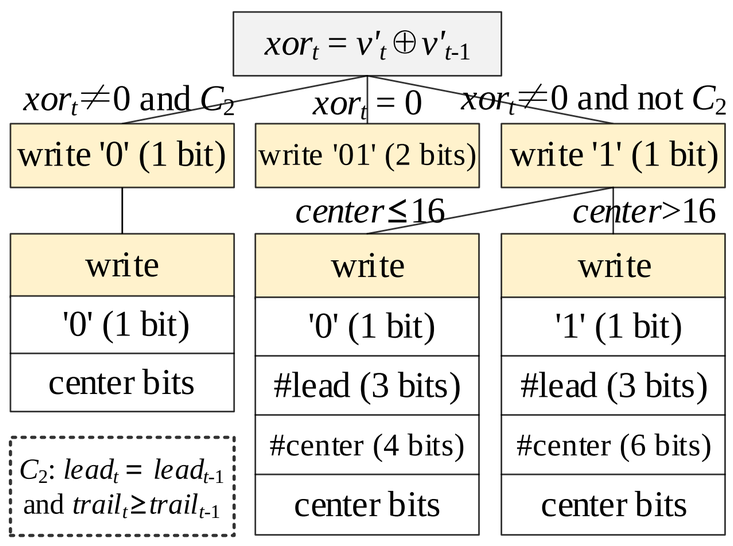


- **Find an *MPN* with Max #Trailing Zeros**
  - $v' = MPN(v, n)$ satisfying $\delta = v - v' \in (0, 10^{-\alpha})$
  - **Basic Method**: Enumerate n from 1 to 52. $O(52)$
  - ***Elf*** : Erase the bits after $m_{g(\alpha)}$ directly. $O(1)$

$$g(\alpha) = \lceil \alpha \times log_2 10 \rceil + \lfloor log_2 |v| \rfloor$$



Encode Erased Data



- ***xor_t* has both many leading zeros and trailing zeros, so *#center*≤16 mostly**
  - 4 bits or 6 bits for *#center*
- ***#lead* is severely unbalanced**
  - 3 bits for *#lead*, Chimp's idea (VLDB 2022)
  - Approximation: (0, 8, 12, 16, 18, 20, 22, 24)

Restore Original Data

- **Step 1: Decode xor_t**
- **Step 2: Recover $v'_t = xor_t \oplus v'_{t-1}$**
- **Step 3: Restore v_t from v'_t**

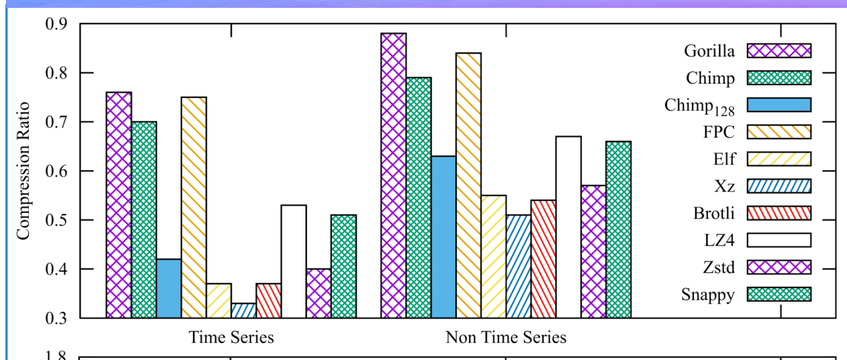
$$v_t = d_{h-1}d_{h-2}...d_0.d_{-1}...d_{-\alpha}\cancel{d_{-(\alpha+1)}...d_l} + 10^{-\alpha}$$

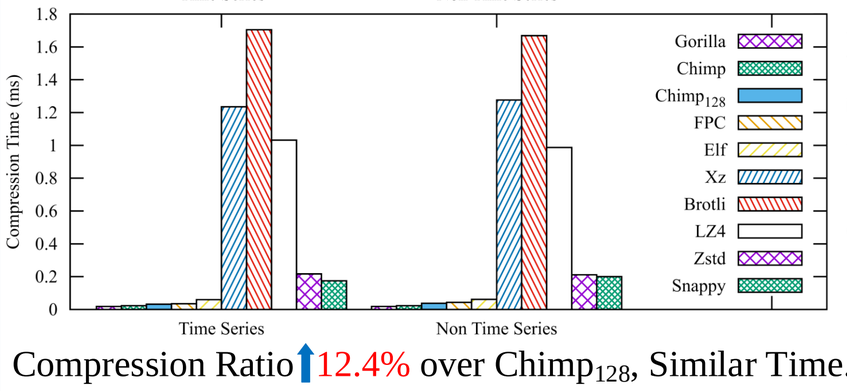
e.g., $3.17 = 3.16\cancel{40625} + 10^{-2}$

- **How to store $\alpha = DP(v)$?**
  - **Basic Method**: α_{max}=324, $\lceil log_2 \alpha_{max} \rceil$ = 9 bits
  - ***Elf***: Store β instead of α with only 4 bits
- **Decimal Significand Count $\beta = DS(v)$**
  - $DS(3.17) = 3$, $DP(-0.0317) = 3$, $DP(317.0) = 4$
  - $\beta \in \{0,1,...,17\}$, but we give up erasing if $\beta \geqslant 16$
- **Start Decimal Significand Pos $SP(v)$**
  - $SP(3.17) = 0$, $DP(-0.0317) = -2$, $DP(317.0) = 2$
- **Calculate α with β and $SP(v')$**

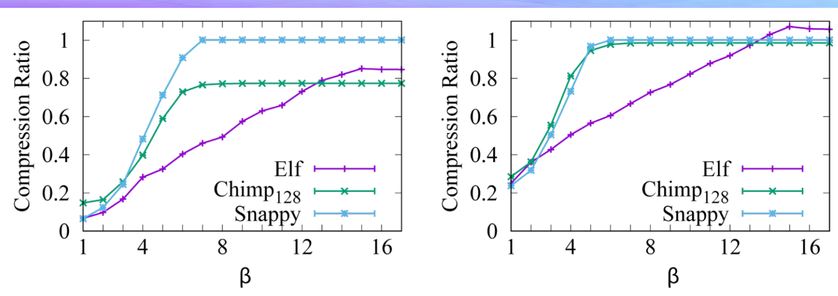
$$\alpha = \begin{cases} \beta - (SP(v') + 1) & v \neq 10^{-i}, i > 0 \\ \beta - (SP(v') + 2) & v = 10^{-i}, i > 0 \end{cases}$$

Experiments (Datasets: 14 Time Series, 8 Non-Time Series. https://github.com/Spatio-Temporal-Lab/elf)

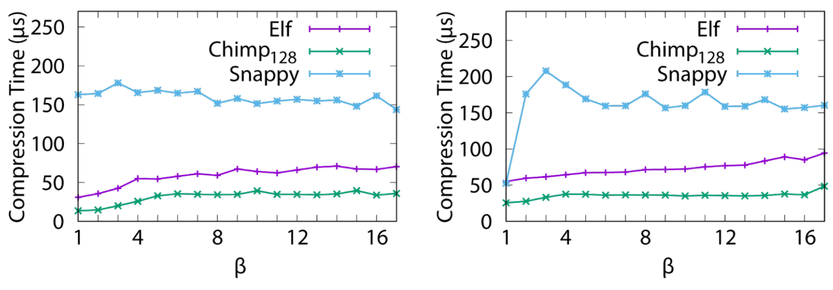




Compression Ratio ⬆12.4% over $Chimp_{128}$, Similar Time.
Competitive Compression Ratio to Xz, Only 4.86% Time.

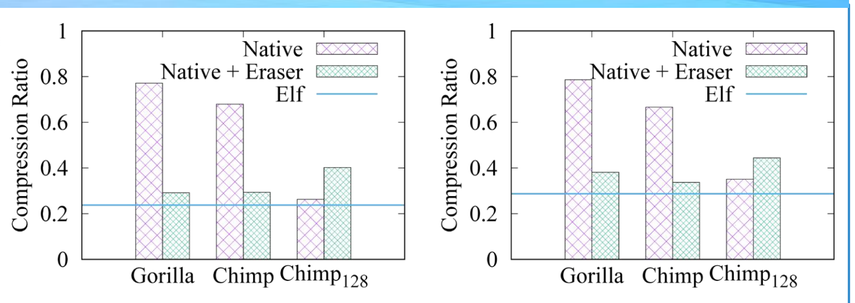


(a) Compression Ratio in AS. (b) Compression Ratio in PLon.

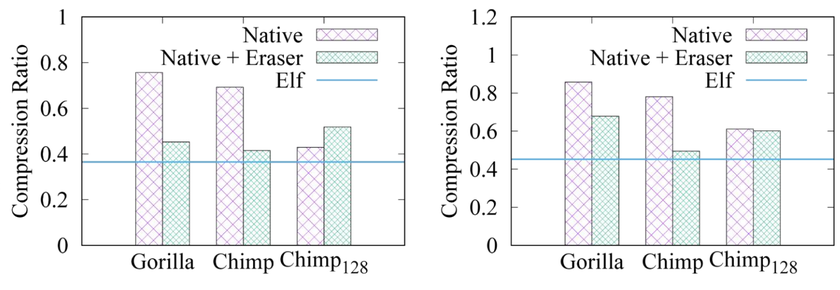


(c) Compression Time in AS. (d) Compression Time in PLon.

Smaller β, Better Compression Ratio, Stable Time.
$51 - \beta log_2 10 <$ #Erased Bits $< 53 - (\beta - 1) log_2 10$.



(a) Time Series with Small β. (b) Non Time Series with Small β.



(c) Time Series with Medium β. (d) Non Time Series with Medium β.

Erasing Technique can Enhance Gorilla and Chimp.
Smaller β, Greater Improvement.