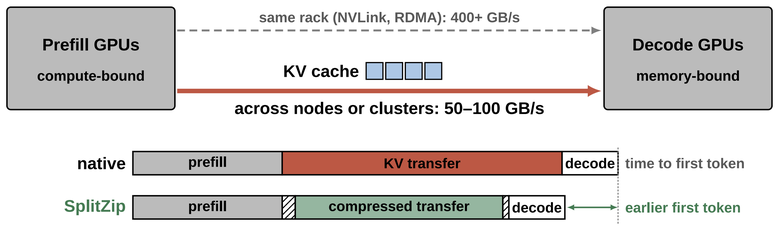
SplitZip: Ultra Fast Lossless KV Compression for Disaggregated LLM Serving

Yipin Guo · **Siddharth Joshi**
University of Notre Dame · yguo23@nd.edu · sjoshi2@nd.edu

Intelligent Microsystems Lab

Disaggregated serving must move each request's KV cache to the decode GPUs before the first token. SplitZip codes the redundant BF16 exponents with fixed 4-bit codes plus a sparse escape stream: **1.32× fewer bytes**, bit-exact, with GPU encode at **613 GB/s** and decode at **2182 GB/s**. On long prompts, time to first token improves by up to **1.30×**.

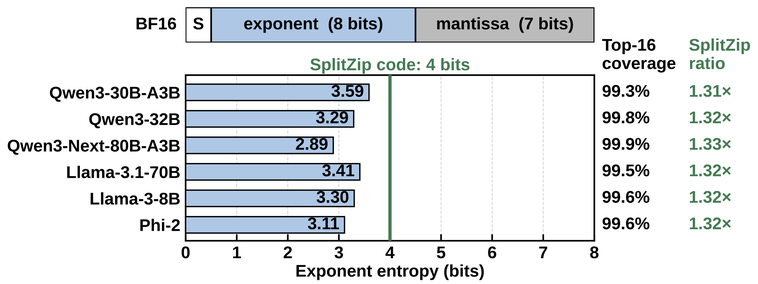
1 KV transfer delays the first token



- Disaggregated serving places compute-bound prefill and memory-bound decode on separate GPUs, so every request's KV cache crosses the network before decoding starts.
- Across nodes or clusters, links carry 50–100 GB/s. For Qwen3-32B at 16 × 64K tokens, moving the raw KV cache over RoCE 4×200G takes **1.75 s**.
- Lossy KV compression changes model outputs. Lossless entropy coders (DFloat11, ZipNN) encode at **0.004–1.2 GB/s**, slower than the link itself.

Goal: lossless KV compression whose GPU encode **and** decode both outrun the link.

2 KV exponents are highly redundant



Across dense, MoE and hybrid models, exponent entropy is 2.9–3.6 bits, and the 16 most frequent exponents cover **≥ 99.3%** of values. SplitZip codes only the exponent and stores the mantissa exactly, reaching **1.31–1.33×** on every model against a 16/12 = **4/3** bound.

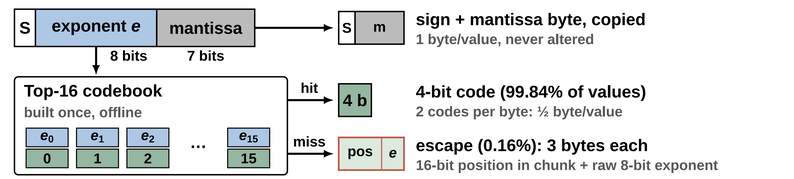
3 Bit-exact, extends to FP8, open source

- Decoding restores every BF16 bit: on Qwen3-32B the generated text matches native serving, with a maximum logit difference of **0** from 128 to 2,840 tokens.
- On FP8 E5M2 KV caches the same Top-16 code gives **1.14×** beyond FP8 (0.16% escapes).

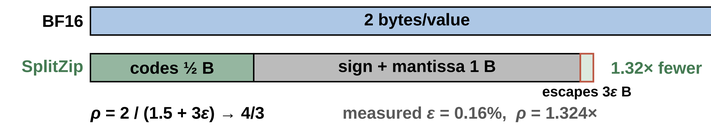
Triton codec, benchmarks and Splash path
github.com/Intelligent-Microsystems-Lab/SplitZip
Splash integration: PR #630

4 SplitZip: fixed 4-bit codes + sparse escapes

1 Encode: split each value, code its exponent

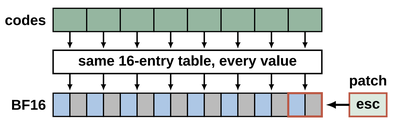


2 Send 1.5 + 3ε bytes per value instead of 2



$\rho = 2 / (1.5 + 3\varepsilon) \rightarrow 4/3$ measured $\varepsilon = 0.16\%$, $\rho = 1.324\times$

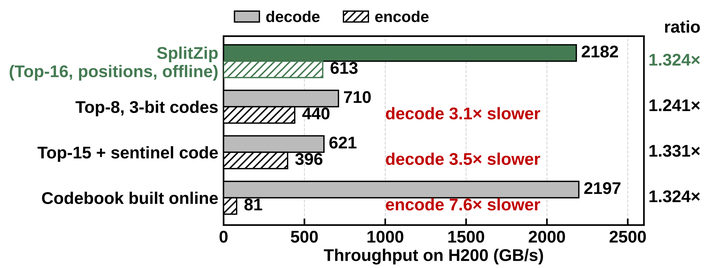
3 Decode: one table lookup per value, then patch



Why not Huffman?
Huffman (DFloat11, ZipNN): variable length
code *i* starts after codes 0…*i*−1: serial
SplitZip: fixed 4 bits
code *i* at bit 4*i*: every thread independent

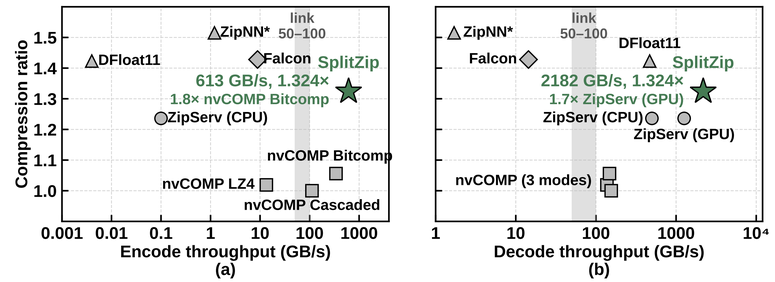
- A one-time offline calibration fixes the codebook, so encoding never builds a histogram. A codebook calibrated on WikiText-2 keeps coverage above 99% on HumanEval, GSM8K, MMLU and PTB.
- Escapes store 16-bit positions within 1024-value chunks, and every value decodes through the same table.

5 Each design choice buys throughput



- **Top-16 vs. Top-8:** escapes drop from 7.89% to 0.16% of values, so decode runs 3.1× faster.
- **Positions vs. a sentinel code:** 3.5× faster decode for a 0.5% lower ratio.
- **Two-pass encode** sizes the escape buffer exactly (125 MiB vs. 48 GiB at 131K tokens).

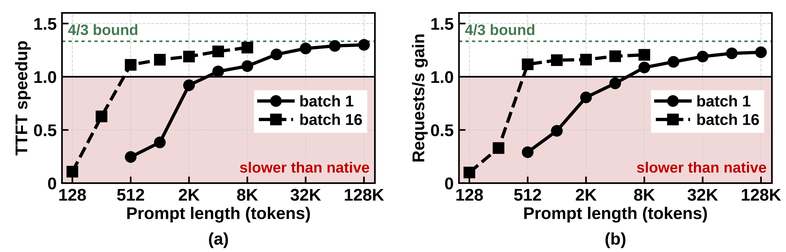
6 Fastest lossless codec on the transfer path



H200, Qwen3-32B BF16 KV. Gray band: 50–100 GB/s link; a codec left of it is slower than sending raw bytes. *ZipNN: throughput from its paper.

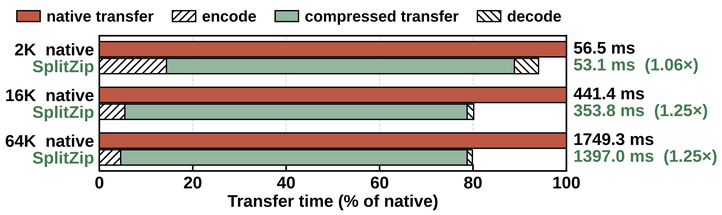
- SplitZip encodes **1.8×** faster than the fastest GPU encoder (nvCOMP Bitcomp) and decodes **1.7×** faster than ZipServ's GPU kernel. Its 1.324× ratio sits 7.1% above ZipServ and 7.0% below DFloat11, whose encoder is 150,000× slower.
- From A100 to B200, encode stays 1.8–3.0× ahead of Bitcomp and decode 1.07–1.75× ahead of ZipServ's GPU kernel, reaching 2607 GB/s on B200.

7 Earlier first token on long prompts



SGLang with Mooncake KV transfer, Qwen3-32B. Speedup over native BF16 transfer.

- TTFT improves from 4K tokens at batch 1 (512 at batch 16) and reaches **1.30×**; request throughput reaches **1.23×**. Shorter prompts pay a fixed codec cost.
- Mooncake KV transfer alone (Llama-3-8B, Qwen3-30B-A3B): up to **1.32×** faster.



Qwen3-32B, batch 16, RoCE 4×200G (700 Gb/s). At 64K tokens, encode and decode take 7.1% of SplitZip's time.

This material is based upon work supported by the National Science Foundation under CAREER Award No. CCF-2340799. Yipin Guo was supported in part by an Arthur J. Schmitt Presidential Leadership Fellowship at the University of Notre Dame.