PERNIX: High-Throughput SIMD Floating-Point Compression

Block quantization and 64-byte packed blocks on CPUs

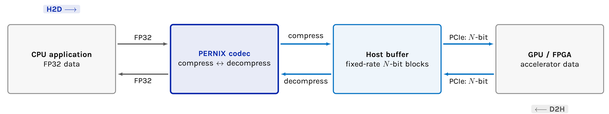
Felix Sternberg[1], Robert Schade[2], and Christian Plessl[1,2]
[1] Department of Computer Science, Paderborn University, Germany
[2] Paderborn Center for Parallel Computing (PC2), Paderborn University, Germany

Approximate floating-point compression with minimal overhead

Problem and Scope

Data movement between CPU applications, host memory, and GPU or FPGA accelerators is the motivating bottleneck. Approximate compression reduces the volume transferred and stored, but only helps when the compression codec is faster than the resulting data path.

Host-to-device (H2D) and device-to-host (D2H) paths are shown below. This work focuses on compression and decompression on the CPU; accelerator-side codec kernels are outside this evaluation.



Design Target

Make compression fast enough that moving the compressed stream determines throughput.

CPU Backends

x86: AVX2, AVX2+BMI2, and AVX-512. ARM NEON and SVE2 are under active development.

Poster Focus

AVX2 and AVX-512 across in-core, host-memory, H2D, and D2H measurements.

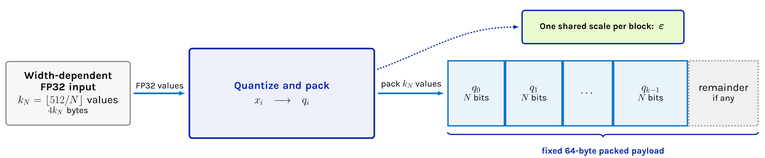
Motivating use case Electron repulsion integrals in hybrid density-functional calculations with CP2K[5].

Quantization and Block Format

PERNIX[1] applies established block-wise quantization[3, 4] with N bits per FP32 value x_i. The maximum magnitude $b_{\max}$ in a block determines its scale ε:

Block-Wise Quantization

Block maximum	Quantization step	Integer code	Reconstruction
$b_{\max} = \max_i \lvert x_i \rvert$	$\varepsilon = \dfrac{b_{\max}}{2^{N-1}-1}$	$q_i = \operatorname{round}\left(\dfrac{x_i}{\varepsilon}\right)$	$\tilde{x}_i = \varepsilon q_i$
maximum magnitude in the block	one shared scale per block	quantize FP32 values to signed N-bit integers	round-to-nearest for in-range values



64-Byte Packed Payload

Each packed block represents

$$k_N = \left\lfloor \frac{512}{N} \right\rfloor, \qquad S_{\text{input}} = 4k_N \text{ bytes}$$

of FP32 input as signed N-bit integers. One scale ε is associated with every independently decodable block.

Reconstruction Error

Round-to-nearest gives

$$|x_i - \tilde{x}_i| \le \frac{\varepsilon}{2},$$

provided values remain inside the represented range.

Fixed Ratio
Fixed-width values give predictable rate.

Independent
Blocks decode independently and can be parallelized.

Fixed Packed Format
Kernels traverse the related values in ISA-width groups.

Packed size is not SIMD work size. A block is always 64 bytes, but represents $k_N = \lfloor 512/N \rfloor$ FP32 values. Therefore the number of blocks is width-dependent. For $N = 12$, 42 FP32 values (168 bytes) become one packed block: AVX2 processes $5 \times 8 + 2$ values; AVX-512-VBMI processes $2 \times 16 + 8 + 2$.

SIMD Design and Contributions

The quantization method is established prior work[4]. PERNIX contributes its high-throughput CPU realization: regular SIMD data motion for both byte-aligned and non-byte-aligned fields.

One Packed Block / Call

One 64-byte block per benchmark kernel call; AVX2 traverses groups of 8 values and AVX-512 groups of 16.

Precomputed Control

Masks, shuffle and permutation indices, and shifts are selected by bit width.

Branch-Free Loop

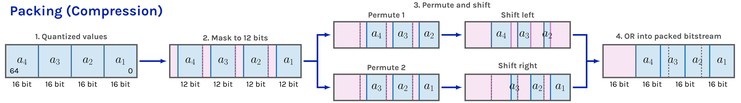
No per-value branches or scalar field extraction occur in the processing loop.

Same process, ISA-specific realization. AVX2, AVX2+BMI2, and AVX-512 use different instructions and vector widths while following the same conceptual packing pipeline.

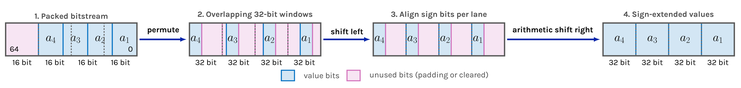
SIMD Bit Packing and Unpacking

Packing uses permutations, shifts, masks, and bitwise combinations. Unpacking uses permutations and shifts to isolate fields and perform vectorized sign extension. This is schematically shown for $N = 12$ and 128-bit SIMD in the following:

Packing (Compression)

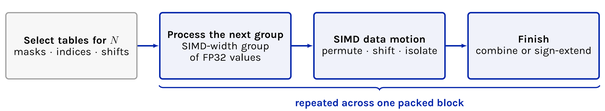


Unpacking (Decompression)



Compression Arrange each N-bit field, shift partial fields into place, then combine them into the packed payload.

Decompression Replicate overlapping windows, align and isolate each field, then sign-extend to full-width integers.



Performance Evaluation

1. In-Core Performance

Codec kernels without input/output traffic; measured versus the LLVM-MCA throughput estimate.

2. Host Memory

Complete loads and stores across L1, L2, L3, and DRAM.

3. End-To-End

Compression codec plus H2D/D2H transfers; compressed versus uncompressed.

Backends: AVX2, AVX2+BMI2, and AVX-512-VBMI where applicable. Dashed curves show static LLVM-MCA throughput estimates; solid curves are measured using the PERNIX benchmark suite[2]. The estimates are comparisons, not upper bounds.

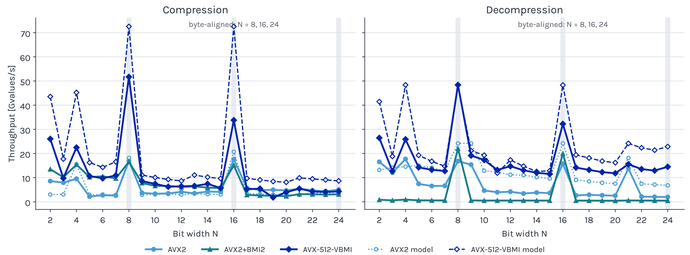
Experimental Setup

Host: 2× AMD EPYC 9655 (192 cores total, SMT off), one pinned thread.
Caches/core: 48 KiB L1D, 1 MiB L2; 32 MiB L3 per 8-core CCD.
Build: benchmarks use GCC 14.3, `-O3`; PCIe uses GCC 15.2, `-O3 -march=x86-64-v3` (AVX2/BMI2) or `-march=x86-64-v4 -mavx512vbmi`.
Protocol: five repetitions after ≥ 0.05 s warm-up; medians reported. LLVM-MCA estimates use a measured frequency of ≈ 4.53 GHz.
PCIe: NVIDIA A40, Gen4×16; 1 GiB original FP32 payload, pinned host memory, asynchronous copies, one stream, event synchronization per transfer, and no codec/copy overlap.

Not measured: accelerator-side codec kernels or a concurrent multicore codec/copy pipeline. Multicore curves assume ideal per-core scaling.

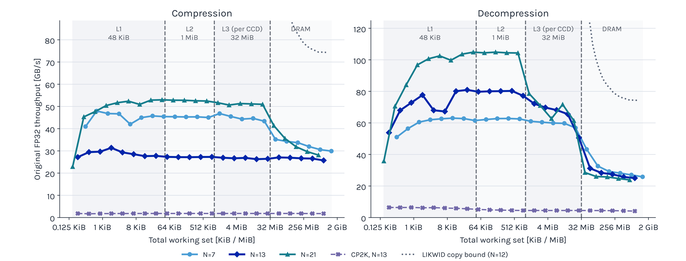
Results

In-Core Throughput



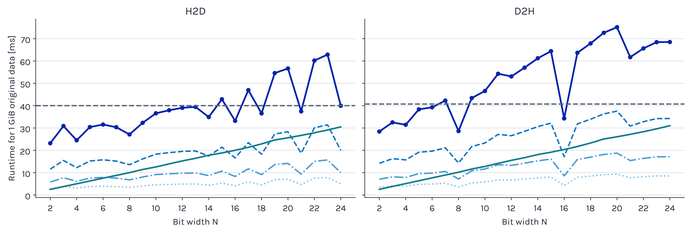
Finding: General kernels cover $N = 2 \ldots 24$. ISA choice and bit layout produce strong width-dependent variation; byte-aligned widths are shown as reference cases.

Host-Memory Throughput



Finding: The $N = 7, 13, 21$ kernels expose the cache-to-DRAM transition. At DRAM scale, $N = 13$ exceeds CP2K by $> 10\times$ for compression and $> 5\times$ for decompression.

End-to-End PCIe



Finding: With ideal scaling, four codec cores make PCIe limiting from $N \geq 7$ for H2D and $N \geq 8$ for D2H; eight cores do so from $N \geq 4$ in both directions. One-core codec and PCIe times are measured.

Impact and Next Steps

8 cores
PCIe-limited from $N \geq 4$ (projected)

$> 10\times$
$N = 13$ compression vs. CP2K

$> 5\times$
$N = 13$ decompression vs. CP2K

Measured: one-core codec + PCIe. **Projected:** ideal 2/4/8-core scaling; no codec/copy overlap.

Application impact not evaluated. Results report codec throughput; the effect of lossy quantization on CP2K simulation outputs has not been measured.

Next: integrate PERNIX in CP2K and validate simulation accuracy; optimize additional CPU platforms (ARM NEON/SVE2 already in development).

References

(1) **PERNIX source code**, https://github.com/pc2/pernix.
(2) **PERNIX benchmark source code**, https : / / github . com / pc2 / pernix-benchmark.
(3) X. Wu, T. Kenter, R. Schade, T. D. Kühne and C. Plessl, 2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2023, pp. 162–173.
(4) M. Guidon, F. Schiffmann, J. Hutter and J. VandeVondele, "Ab Initio Molecular Dynamics Using Hybrid Density Functionals", *The Journal of Chemical Physics*, 2008, **128**, 214104.
(5) T. D. Kühne et al., "CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations", *The Journal of Chemical Physics*, 2020, **152**, 194103.
(6) A. S. Charif-Rubial, E. Oseret, J. Noudohouenou, W. Jalby and G. Lartigue, 2014 21st International Conference on High Performance Computing (HiPC), 2014 21st International Conference on High Performance Computing (HiPC), 2014, pp. 1–10.

PERNIX source

Benchmarks