PERNIX: High-Throughput SIMD Floating-Point Compression

Felix Sternberg*, Robert Schade†, and Christian Plessl*†
*Department of Computer Science, Paderborn University, Germany
†Paderborn Center for Parallel Computing (PC2), Paderborn University, Germany

I. Motivation

The growing disparity between processor throughput and data transfer bandwidth makes data movement a central performance constraint in scientific computing. Electron repulsion integrals (ERIs) in hybrid density-functional calculations with CP2K are one example: CPUs, GPUs, or FPGAs produce large intermediate data sets that must be transferred over the memory hierarchy, an accelerator interconnect, or host memory for reuse[1], [2]. Reducing their representation can save bandwidth and capacity, but compression is beneficial only when the CPU codec is faster than the critical data path.

PERNIX[3] targets this requirement with a maintainable C++ library and C, C++, and Fortran interfaces. It implements lossy compression for workloads in which a controlled reduction in numerical accuracy is acceptable. The packed format is compatible with the 512-bit blocks used by a prior FPGA implementation for ERIs[4]. This supports heterogeneous CPU-accelerator pipelines and reuse of compressed data.

II. Quantization and Packed Format

PERNIX applies block-wise, fixed-rate quantization to FP32 values. For a block with maximum magnitude $b_{\max}$ and a target width of N bits, the scale is

$$\varepsilon = \frac{b_{\max}}{2^{N-1} - 1} \tag{1}$$

Each value x_i is converted to a signed integer $q_i = \mathrm{round}(x_i/\varepsilon)$ and reconstructed as $x_i \approx \tilde{x}_i = \varepsilon q_i$. Round-to-nearest bounds the reconstruction error by $|x_i - \tilde{x}_i| \leq \varepsilon/2$, provided that the values remain within the represented range.

Each independently decodable block has a 64-byte packed payload and an associated scale. It contains $k_N = \lfloor 512/N \rfloor$ signed N-bit values and therefore represents $4k_N$ bytes of original FP32 data. This fixed layout offers consistent compression ratios and supports widths ranging from 2 to 24 bits. However, the packed size should not be confused with the SIMD work size. For instance, a 12-bit block contains 42 values, which the kernels traverse in groups determined by the instruction set width.

III. SIMD Implementation

PERNIX's contribution is a high-throughput CPU realisation of this quantisation method. Three x86 backends, AVX2, AVX2+BMI2, and AVX-512-VBMI, follow the same conceptual pipeline but use ISA-specific vector widths and data-motion instructions. AVX2 processes groups of eight values, while AVX-512 processes groups of sixteen. Control data, including masks, shifts, and shuffle or permutation indices, is precomputed for each bit width. Consequently, the processing loop contains neither per-value branches nor scalar field extraction.

Compression involves arranging quantized fields, shifting partial fields into position and combining them into the packed payload. Decompression involves replicating overlapping input windows, aligning and isolating fields, performing vectorized sign extension and applying the block scale. This replaces explicit per-bit work with regular SIMD permutations, masks, shifts and logical operations. Byte-aligned widths serve as efficient reference cases, whereas non-byte-aligned widths reveal the cost and effectiveness of the general packing machinery. ARM NEON and SVE2 backends are under development, but are outside the scope of this evaluation.

IV. Evaluation and Results

The evaluation separates codec execution from data movement. In-core benchmarks omit input and output traffic, comparing measurements from the PERNIX benchmark suite[5] with static LLVM-MCA throughput estimates. These estimates are comparative models rather than strict upper bounds. Host-memory benchmarks include complete loads and stores, scaling the working set across L1, L2, L3 and DRAM. Throughput is reported as either the original FP32 values or the number of bytes per second, taking into account the number of values represented by each packed block, which depends on its width.

For $N = 2 \ldots 24$, the choice of ISA and bit layout cause variation in in-core throughput depending on the width. Host-memory measurements for N = 7, 13 and 21 demonstrate a transition from codec- or cache-limited execution to DRAM-limited behaviour. For DRAM-scale working sets, the 13-bit PERNIX kernels outperform the existing implementation in CP2K by more than 10x for compression and 5x for decompression. Therefore, the

optimized codec can reach the data-movement limit with fewer CPU resources, leaving more cores available for the scientific calculation.

V. Summary

PERNIX uses fixed-rate block quantization and ISA-specific SIMD packing to compress FP32 data with bit widths ranging from 2 to 24. Its AVX2, AVX2+BMI2, and AVX-512-VBMI kernels deliver compression and decompression throughputs that are more than ten and five times higher, respectively, than those of the existing CP2K implementation for 13-bit data at DRAM-scale working sets. This shifts the primary constraint from codec execution to data movement, leaving more CPU resources available for scientific computation. Future work will evaluate complete accelerator transfers, overlap compression with computation, extend the measurements to additional x86 systems, and complete the ARM backends.

References

[1] 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, vol. 152, no. 19, p. 194 103, May 2020, issn: 0021-9606. doi: 10.1063/5.0007045. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/5.0007045/16718133/194103_1_online.pdf. [Online]. Available: https://doi.org/10.1063/5.0007045.

[2] M. Guidon, F. Schiffmann, J. Hutter, and J. VandeVondele, "Ab initio molecular dynamics using hybrid density functionals," The Journal of Chemical Physics, vol. 128, no. 21, p. 214 104, Jun. 4, 2008, issn: 0021-9606. doi: 10.1063/1.2931945. Accessed: Aug. 24, 2025. [Online]. Available: https://doi.org/10.1063/1.2931945.

[3] "Pernix source code. "[Online]. Available: https://github.com/pc2/pernix.

[4] X. Wu, T. Kenter, R. Schade, T. D. Kühne, and C. Plessl, "Computing and Compressing Electron Repulsion Integrals on FPGAs," in 2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), May 2023, pp. 162–173. doi: 10.1109/FCCM57271.2023.00026. Accessed: Aug. 14, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10171537.

[5] "Pernix benchmark source code. "[Online]. Available: https://github.com/pc2/pernix-benchmark.