A Distributed Mixed-Precision N-Body Engine

Scaling Behavior and the Precision Limits of a Direct-summation Kernel

Tenicka Norwood, Michael Flanders, Naw Safrin Sattar

Computer Science and Data Science Department, School of Applied Computational Sciences, Meharry Medical College, Nashville, Tennessee, USA.

Email: tenicka.norwood@mmc.edu, michael.flanders@mmc.edu, nawsafrin.sattar@mmc.edu

57x
strong-scaling speedup at 96 ranks, N = 32768

99%
parallel efficiency through 8 ranks

2.9x
throughput gain from native fp32 at 96 ranks

~100x
error penalty for rounding the accumulator

0.5%
initial-condition energy vs analytic Plummer

The accumulator, not the terms, sets the precision floor. Rounding the running sum as well as each term costs 92-125x more error at tf32, fp16 and bf16 than rounding the terms alone.

Motivation

- Direct summation is the accuracy reference for N-body [1]: every pair, no approximation, unlike a tree code [2].
- $O(N^2)$ arithmetic on O(N) data, so the kernel is compute-bound, not bandwidth-bound.
- Low precision promises throughput [3]. What does it cost the physics?

Problem

$$a_i = G\sum_j m_j \frac{r_j - r_i}{\left(\|r_j - r_i\|^2 + \varepsilon^2\right)^{3/2}}$$

- **RQ1.** Where does the one collective per step stop being free?
- **RQ2.** How far can precision fall before the physics degrades, and is the limit the terms or the accumulator?

Methods

- **Leapfrog is symplectic [4].** Energy error stays bounded, so any secular drift is arithmetic, not the integrator.
- **Plummer has a known answer [5].** Exact total energy $-3\pi/64 = -0.147262$ gives a free correctness check.
- **Softening kills the branch.** The self term is exactly zero, so the inner loop stays one vectorizable reduction.

Validated: initial conditions reproduce the analytic energy to 0.5% at N = 16384, with seed spread falling from 4.4×10^{-3} to 5.1×10^{-4} as N grows: $1/\sqrt{N}$ sampling noise, not systematic error.

Parallel Design

- **N/p bodies per rank, N fixed.** Speedup is efficiency alone.
- **One MPI_Allgatherv per step.** Positions packed into one buffer, not three collectives.
- **Timers inside the step.** An outer timer cannot separate the kernel from the network.
- **Identical state everywhere.** Each rank builds the full IC from a seed and keeps its slice, so state is bit-identical at every rank count.

Per-Step Pipeline

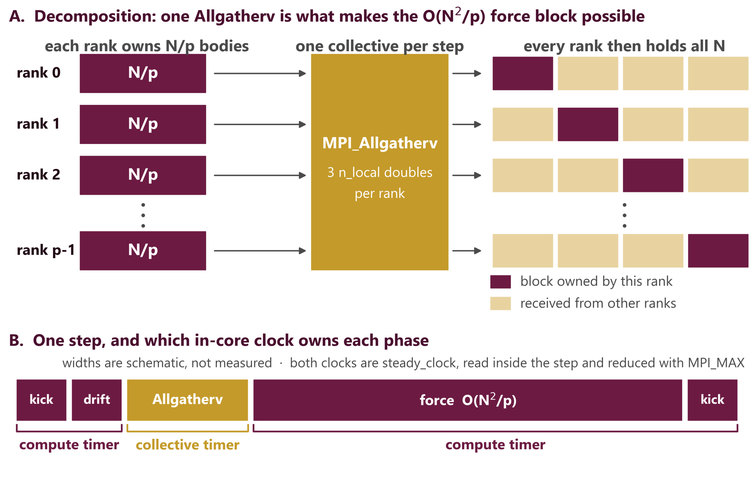


Fig 1. Per-step decomposition and the two in-core clocks.

Results

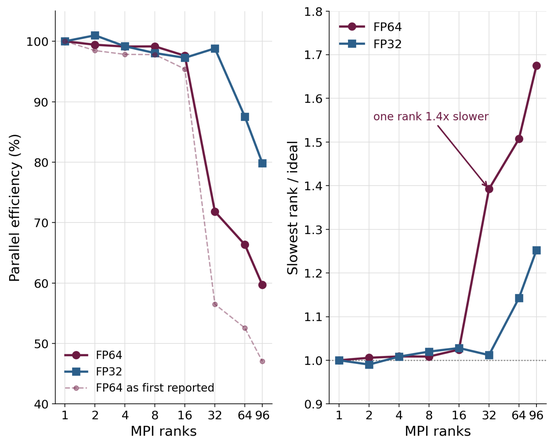


Fig 2. Strong scaling, N = 32768. Left: efficiency (dashed: as first reported). Right: slowest rank vs ideal.

- **Near-ideal to 16 ranks.** 99% efficiency through 8, 98% at 16.
- **A straggler, not the collective.** From 32 ranks one FP64 rank ran 1.4x slower. The others' wait at the Allgatherv looked like a 21% collective; transfer is under 1% of a step.

FP32 had no straggler. It held 80% efficiency at 96 ranks against 60% for FP64, 2.9x faster. Step time is the slowest rank's compute, which matched measured step time within 1% in a follow-up run.

Precision Model

- **Native vs emulated.** FP64 and FP32 run on hardware. TF32 through FP8 [6] are emulated [7], ~40× slower; their timings are void.
- **Two accumulator arms.** Round the terms, or the running sum too. The second admits stagnation [8]: once the sum exceeds a term by 1/u, the term vanishes.
- **Related work.** GPU N-body codes use FP32 or double-single [9, 10]; compensated summation [8, 11] and fixed-point force accumulation [12] repair the sum.

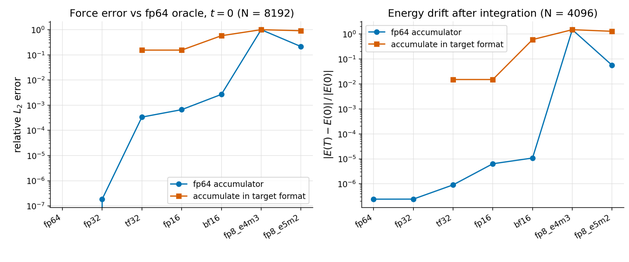


Fig 3. Precision ladder against the FP64 oracle. Left: relative L2 force error at t = 0. Right: relative energy drift.

Setup

- **Hardware.** ACES: Sapphire Rapids, 96 cores/node, NDR200, Lustre.
- **Toolchain.** intel/2023a via icpx, Intel MPI 2021.9, srun --mpi=pmi2, -fp-model=precise to keep the FP64 oracle honest.
- **Protocol.** Barriered windows per point, MPI_MAX reduced, provenance recorded per run.

Conclusions

- **Per-rank timers found the real limit.** Past 16 ranks one FP64 rank ran 1.4x slower; the collective itself costs under 1% of a step.
- **Precision is limited by the reduction, not the operands.** Keep the accumulator in FP64 [8] and spend low precision on the terms.
- **Chaos rules out trajectory error.** Late-time divergence measures Lyapunov growth, not rounding. Force error at t = 0 and symplectic energy drift isolate the arithmetic.

Future Work

- Find the straggling core; 1024 ranks across 11 nodes; pairwise and stochastic-rounding accumulators.
- Reduced-precision hardware arm; non-blocking Allgatherv to overlap the collective; distributed tree code [2].

References. [1] Aarseth, Gravitational N-Body Simulations, CUP (2003) [2] Barnes & Hut, Nature 324:446 (1986) [3] Higham & Mary, Acta Numer. 31:347 (2022) [4] Hairer, Lubich & Wanner, Acta Numer. 12:399 (2003) [5] Plummer, MNRAS 71:460 (1911) [6] Micikevicius et al., arXiv:2209.05433 (2022) [7] Higham & Pranesh, SIAM J. Sci. Comput. 41:C585 (2019) [8] Higham, Accuracy and Stability of Numerical Algorithms, SIAM (2002) [9] Nyland, Harris & Prins, GPU Gems 3, ch. 31 (2007) [10] Gaburov, Harfst & Portegies Zwart, New Astron. 14:630 (2009) [11] Kahan, Commun. ACM 8:40 (1965) [12] Le Grand, Götz & Walker, Comput. Phys. Commun. 184:374 (2013).

Acknowledgements. Advisor: Dr. Naw Safrin Sattar (Meharry). Compute: ACCESS/ACES CIS260987 (TAMU HPRC); NSF #2138259, #2138286, #2138307, #2137603, #2138296. AI-assisted tooling

Contributions. Norwood: engine, experiment design, analysis, manuscript. Flanders: ACES bring-up and toolchain, Lustre-safe result I/O, launcher robustness, provenance, code review.