OAK RIDGE National Laboratory

Cabana: a Kokkos+MPI library for particles and hybrid particle-grid applications

Sam Reeve

Oak Ridge National Lab

ORNL is managed by UT-Battelle LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Slattery et al., (2022). Cabana: A Performance Portable Library for Particle-Based Simulations. Journal of Open Source Software, 7(72), 4115, https://doi.org/10.21105/joss.04115

Physics applications

CabanaMD
Molecular dynamics

Application codes

Computational motifs

Cabana

*Performance portable, multi-node **particle, structured-grid,** and **particle-grid** motifs*

Distributed parallelism

MPI
Multi-node computation

ArborX
Geometric search

heFFTe
FFTs

hypre
Structured solvers

HDF5
Particle I/O

Threaded parallelism

Kokkos
On-node performance portability

Threaded backends

OpenMP | CUDA | HIP | SYCL

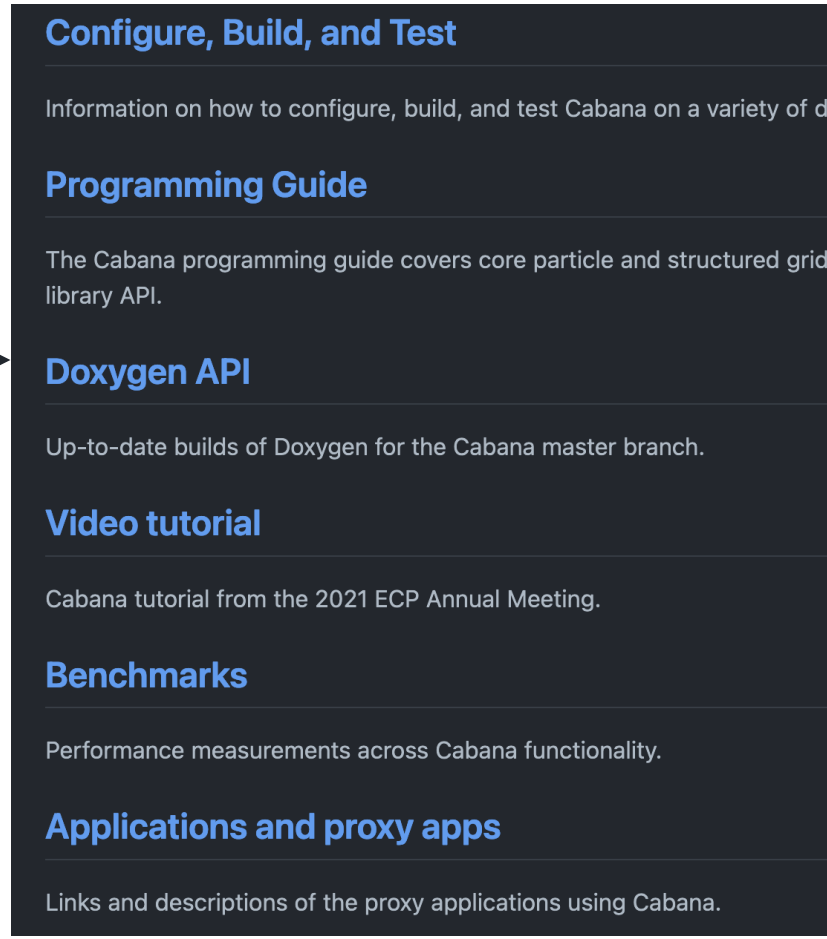
Cabana: created during ECP to support particle work

Lab leads

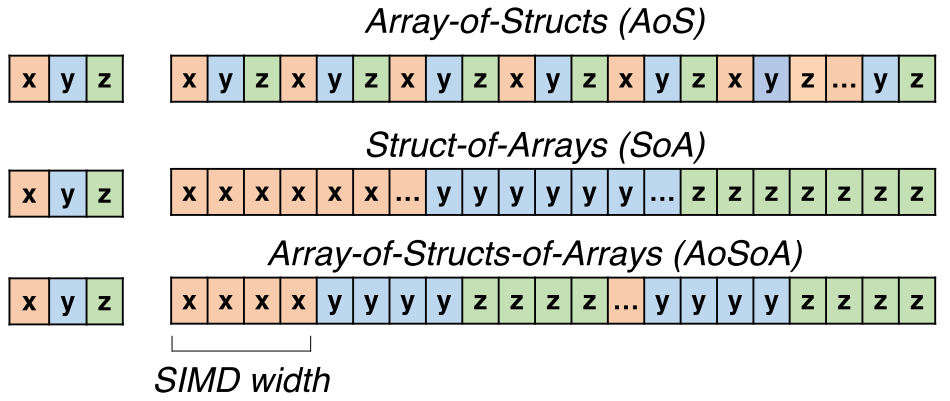
PI: Sue Mniszewski, LANL
Co-PI: Jim Belak, LLNL
Site Lead: C-S Chang, PPPL
Site Lead: Salman Habib, ANL
Site Lead: Steve Plimpton, SNL
Site Lead: Stuart Slattery, ORNL
FFTX Lead: Phil Collela, LBNL

Cabana resources

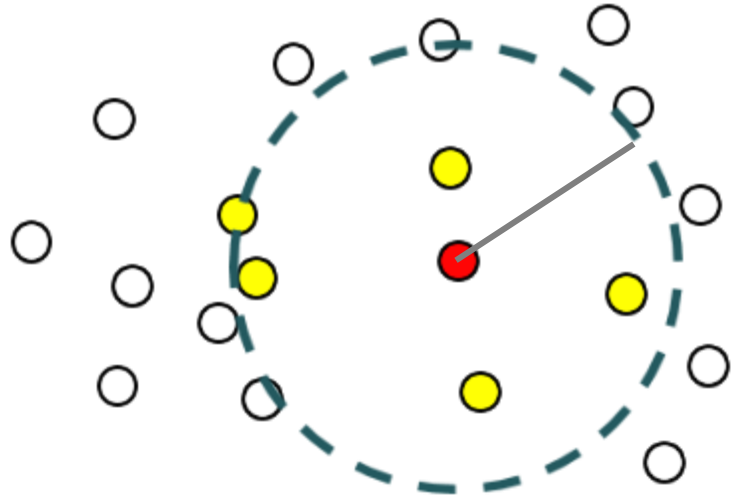
- GitHub: https://github.com/ECP-CoPA/Cabana
  - We welcome issues and pull requests
  - **Examples/tutorials in source code**
  - GitHub wiki
- YouTube tutorial (2021): https://youtu.be/VAS7JdprQqE
- Slack https://kokkosteam.slack.com **#cabana**
- Email (reevest@ornl.gov)



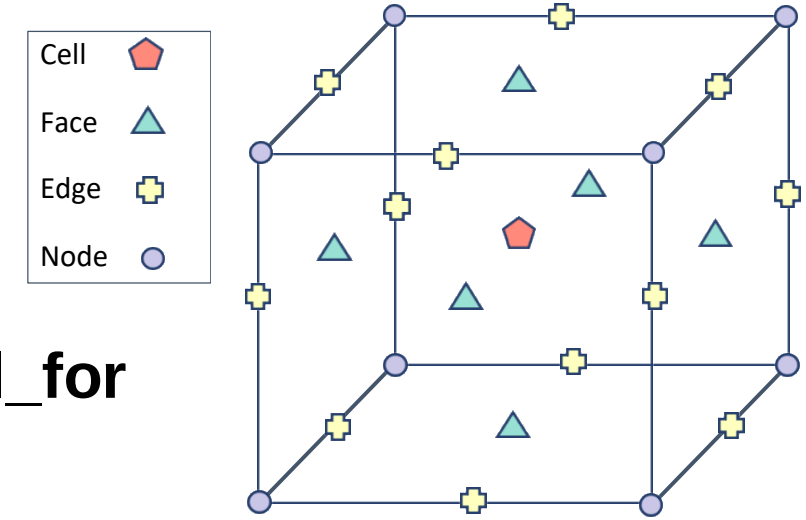
Primary Core library features



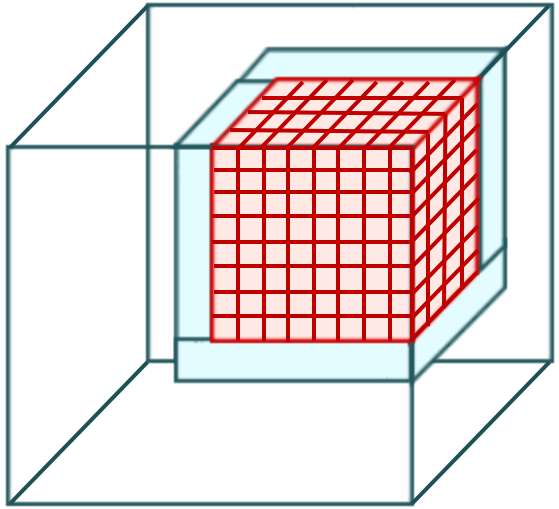
- Particle data structures (View extensions)
- Binning/sorting
  - Including spatial binning with **Cabana::LinkedCellList**
- Neighbor interactions
  - Neighbor lists and **Cabana::neighbor_parallel_for**
    - **Cabana::VerletList** (bin-accelerated N^2 search)
    - ArborX interface (tree based search)
    - **Cabana::LinkedCellList** (binning only)
- MPI Communication
  - Particle migration **Cabana::migrate**
  - Particle halo **Cabana::{Gather,Scatter}**



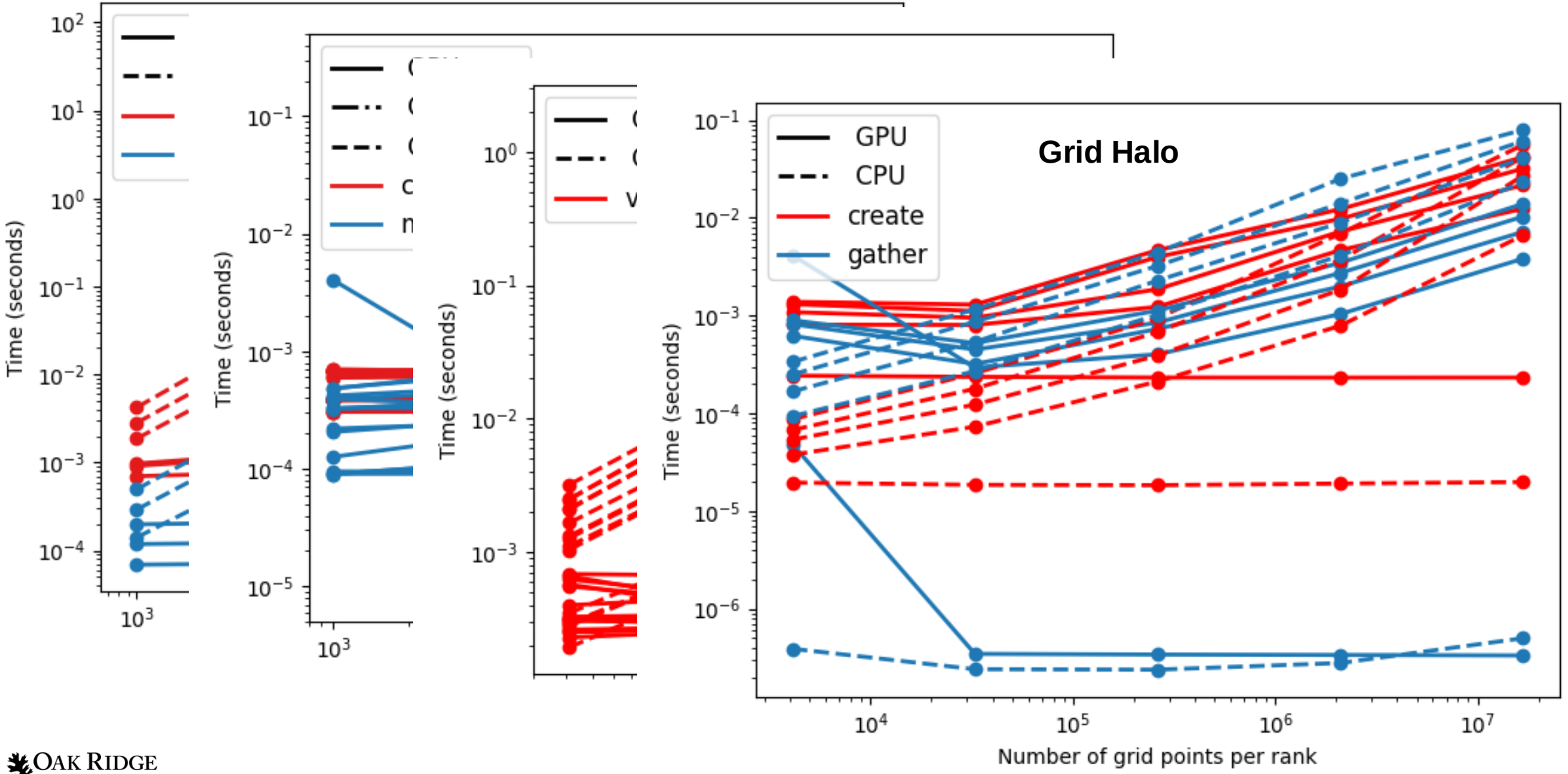
Primary Grid library features



- Domain decomposed structured grids
  - Grid objects: indices for **Cabana::Grid::grid_parallel_for**
  - Mesh objects: physical cell details
- Particle-grid interpolation: **Cabana::Grid::{p2g, g2p}**
- Algorithms
  - FFTs (heFFTe interface)
  - Solvers/preconditioners (HYPRE interface)
  - Load balancing (ALL interface)
- MPI Communication: **Cabana::Grid::Halo**
- Sparse grids (logically dense)
  - Only allocate the portion of the grid fields where particles exist



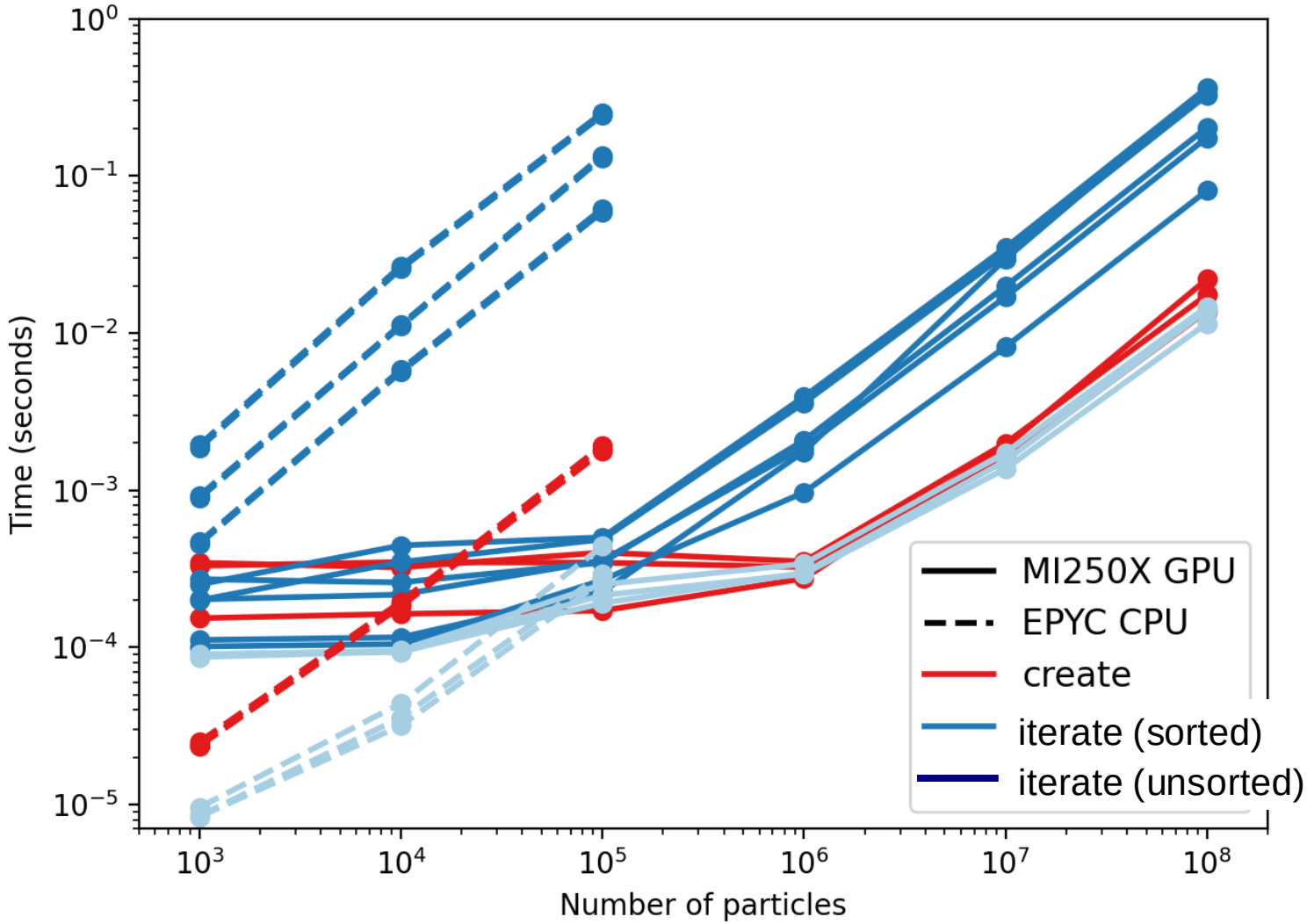
Cabana benchmarks (Frontier 1 GCD/node)



Cabana performance (Frontier)

Particle binning as a replacement for neighbor list storage.

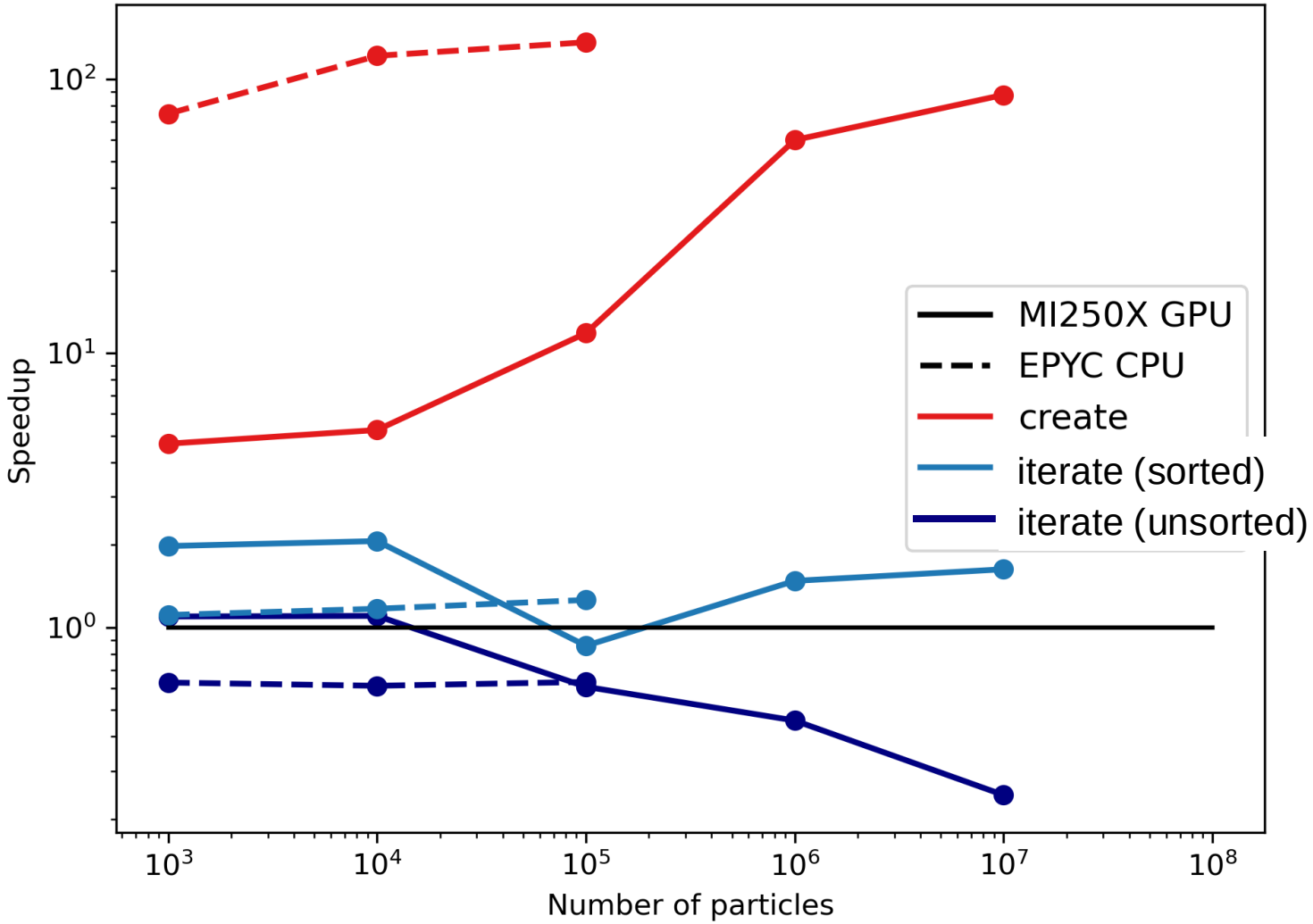
- Creation (binning)
- Parallel iteration with and without sorting



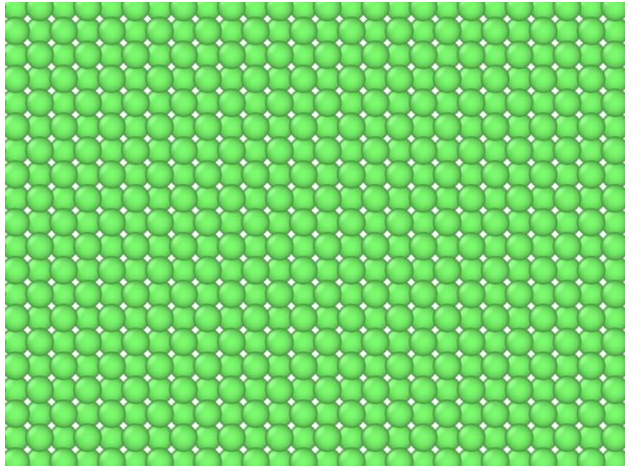
Cabana performance (Frontier)

Speedup from stored Verlet-style list and binning-only as a neighbor list

- Creation (binning)
- Parallel iteration with and without sorting



CabanaMD proxy app



Memory/ flop kernel

Communication

MD data

Single MD timestep

1. Halo exchange of ghost cells and redistribution

2. Construct neighbor lists *(Not done every timestep)*

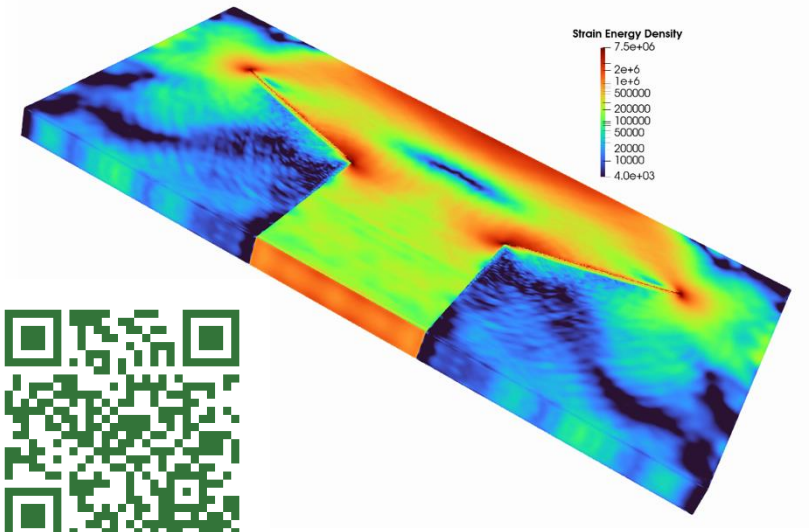
3. Compute forces on particles due to short-range neighbors

4. Integrate equations of motion (particle update)

5. Resorting of particles *(Not done every timestep)*

- Cabana::AoSoA, Kokkos::Views, and Cabana::Grid::UniformMesh
- **Cabana::migrate(Cabana::Distributor(…), AoSoA);**
  Cabana::gather(Cabana::Halo(…), x);
- **Cabana::VerletList(x, …);**
- **Cabana::neighbor_parallel_for(…); *// with x, f***
- **Kokkos::parallel_for(…); *// with x, v, f***
- **Cabana::permute(Cabana::LinkedCellList(x, AoSoA), AoSoA);**

CabanaPD: peridynamics timestep



0. Construct neighbor lists

1. Halo exchange of ghosts

2. Compute forces on particles due to short-range neighbors & bond-breaking

3. Integrate equations of motion (particle update)

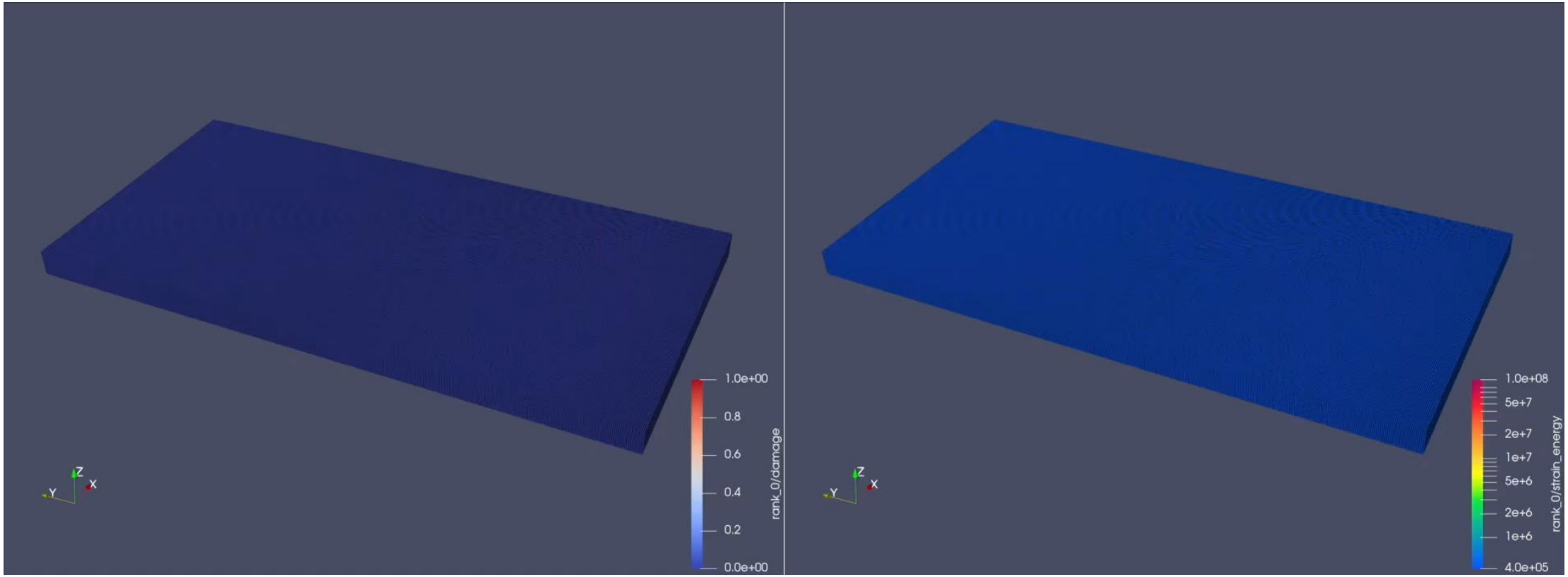
4. Apply boundary conditions

5. Compute energy
(Not done every step)

- **Cabana::VerletList(x, …);**
- **Cabana::Gather::apply( );**
- **Cabana::neighbor_parallel_for(…);** *// with u, x, f*
- **Kokkos::parallel_for(…);** *// with u, v, f*
- **Kokkos::parallel_for(…);** *// with f*
- **Cabana::neighbor_parallel_reduce(…);** *// with u, x, f*

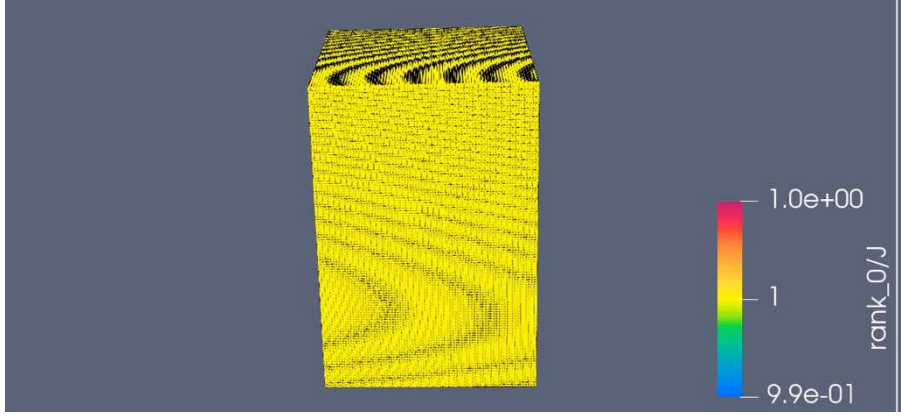
CabanaPD

- New fracture mechanics code using mesh-free peridynamics
- Current extension to contact problems and multi-physics



OAK RIDGE National Laboratory

ExaMPM proxy app



Memory/ flop kernel | **Communication**

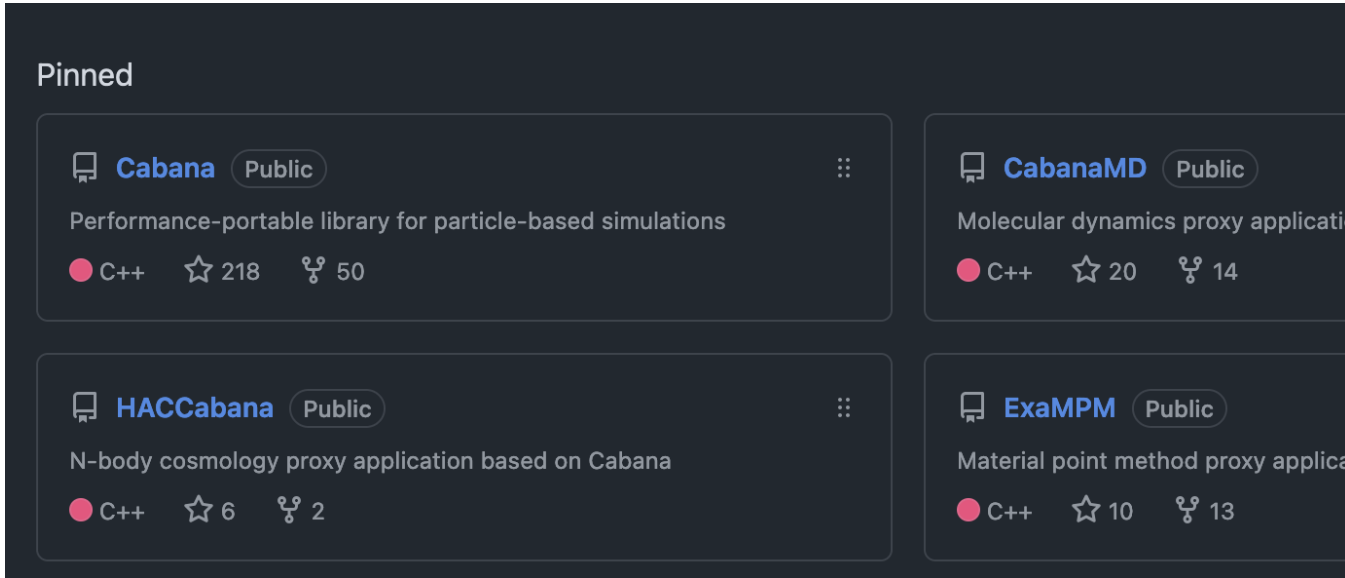
1. **Particle to grid**
2. **Field solve on the grid**
3. **Grid to particle**
4. **Apply boundary conditions and correct positions**

- **Kokkos::parallel_for** with **Cabana::Grid::P2G::{gradient, value}**
  - And **Cabana::Grid::Halo::gather**
- **Cabana::Grid::grid_parallel_for**
- **Kokkos::parallel_for** with **Cabana::Grid::G2P::{gradient, value}**
  - And **Cabana::Grid::Halo::scatter**
- **Kokkos::parallel_for** and **Cabana::Grid::grid_parallel_for**

** Could replace with **Cabana::Grid::{p2g, g2p}**, if fusing the kernels is not performance critical*

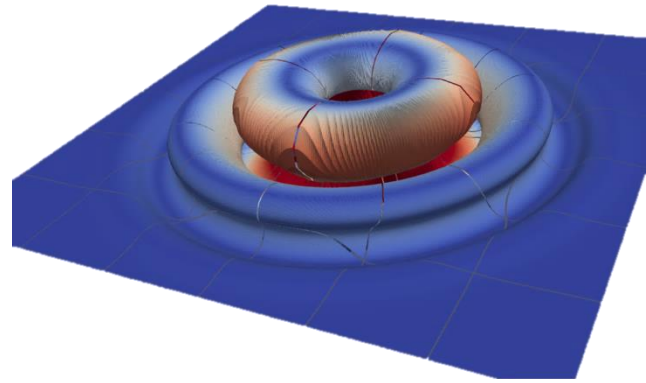
Cabana proxy applications (https://github.com/ECP-CoPA)

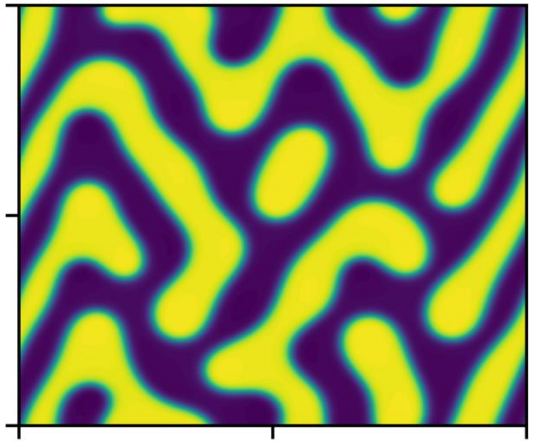
- CabanaMD (EXAALT) [LAMMPS]
  - Classical molecular dynamics
- HACCabana (ExaSky, ANL) [HACC]
  - Potential exploratory use alongside production HACC
- ExaMPM [PicassoMPM]
  - New material point method for additive manufacturing
- CabanaPIC (LANL) [VPIC]
  - Plasma PIC

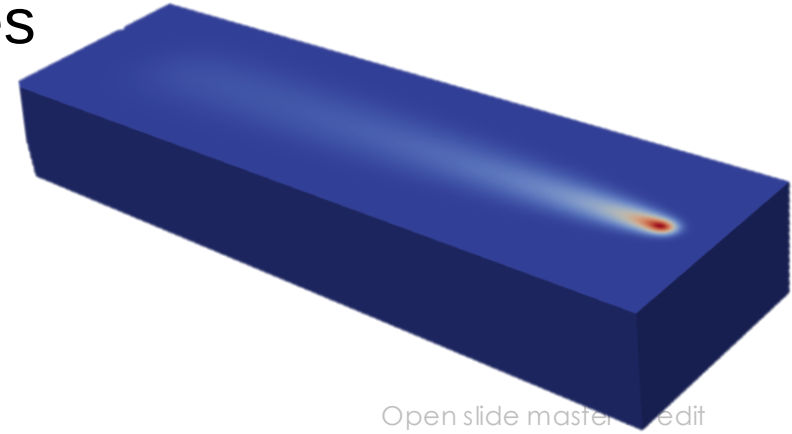


Cabana applications, continued

- **Beatnik (UNM PSAAP):** Z-model interface solver
- **CabanaPD (ORNL LDRD):** Peridynamics for fracture
- **CabanaPF (ORNL SULI):** Pseudospectral phase field
- **CabanaMPCD (Julich):** Multi-particle collision dynamics
- **Finch (ORNL MDF):** Heat transfer for additive manufacturing
- **Hyperion (LANL LDRD):** multi-physics hybrid PIC
- **MultiSim (UCLA):** Material point method for graphics
- **PUMI-PIC (RPI):** PIC library using Cabana data structures







Cabana interoperability

- Cabana has been built to connect and unify across disparate scientific communities
- We are completely intertwined with Kokkos and MPI
  - But we do support interfaces with other portability strategies
    - heFFTe & hypre vs ArborX & KokkosComm
- Primary current effort is better separation of data structure and algorithm
  - Accept **`Kokkos::View`** anywhere we use **`Cabana::slice`**
  - The particle data could be **`Unmanaged`** and therefore wrap user data without significant code intrusion

Current status and future possibilities

- ORNL science areas for Cabana-based applications
  - Fracture mechanics (fusion, materials science)
  - Materials and manufacturing (additive, etc.)
  - Plasma physics (fusion)
  - Mesh generation

- Develop at the application level
  - Generalize and move code to Cabana where appropriate
    - Generalize and move code to Kokkos{Comm, FFT} where appropriate
      - Generalize and move code to Kokkos where appropriate

Moving code upstream

- From ExaAM Picasso into Cabana
  - **Cabana::ParticleList** for user type-tagged particle fields
  - **Cabana::particleGridMigrate** to determine particle destinations based on mesh
  - **Cabana::Grid::createParticles(Cabana::{InitRandom, InitUniform}, …)**
- *In progress* from CabanaPD into Cabana
  - **Cabana::particleGrid{Gather, Scatter}**
- *Hypothetically* **Cabana::AoSoA** into Kokkos
  - View extension with extra dimension for SIMD/SIMT width
  - Supports packing multiple fields together
  - Associated execution policies and parallel_for

OAK RIDGE National Laboratory

Work performed under the auspices of the U.S. DOE under contract DE-AC05-00OR22725 for Oak Ridge National Laboratory.

This research was sponsored by the Oak Ridge National Laboratory Directed Research and Development program VERTEX Initiative.

Cabana references

- Slattery, S., Reeve, S. T., Junghans, C., Lebrun-Grandié, D., et al. (2022). Cabana: A performance portable library for particle-based simulations. *Journal of Open Source Software*, *7*(72), 4115. https://doi.org/10.21105/joss.04115
- Mniszewski, S. M., Belak, J., et al. (2021). Enabling particle applications for exascale computing platforms. *The International Journal of High Performance Computing Applications*, *35*(6), 572-597.
- Qiu, Y., Reeve, S. T., Li, M., Yang, Y., Slattery, S. R., & Jiang, C. (2023). A Sparse Distributed Gigascale Resolution Material Point Method. *ACM Transactions on Graphics*, *42*(2), 1-21.
- Desai, S., Reeve, S. T., & Belak, J. F. (2022). Implementing a neural network interatomic model with performance portability for emerging exascale architectures. *Computer Physics Communications*, *270*, 108156.