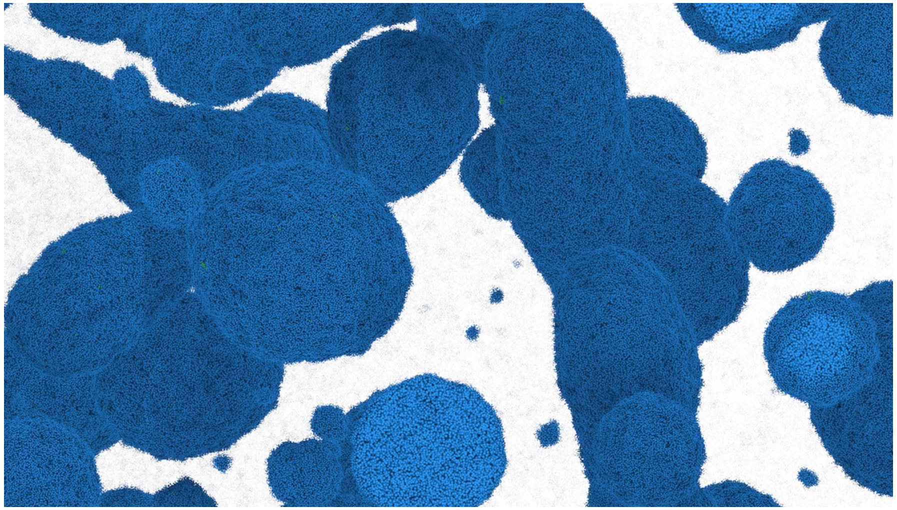
SAND2022-16052C

Exceptional service in the national interest



Large-Scale Atomistic Simulations of Molten Metal Expansion

Stan Moore, Mitchell Wood,
Kyle Cochrane, Aidan Thompson
Sandia National Laboratories

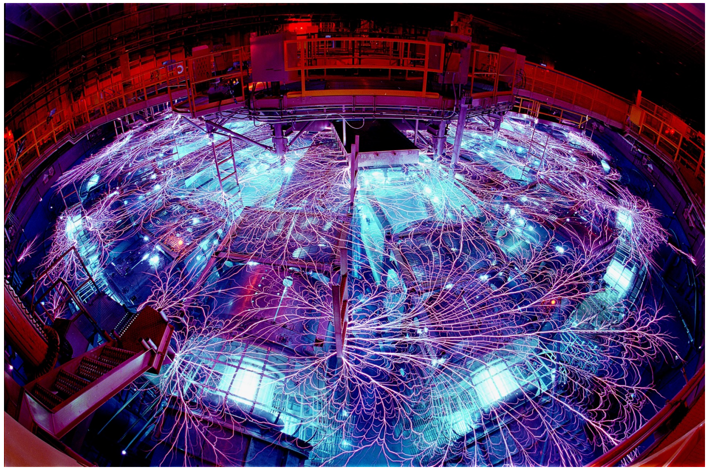
SC22 Conference

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Introduction

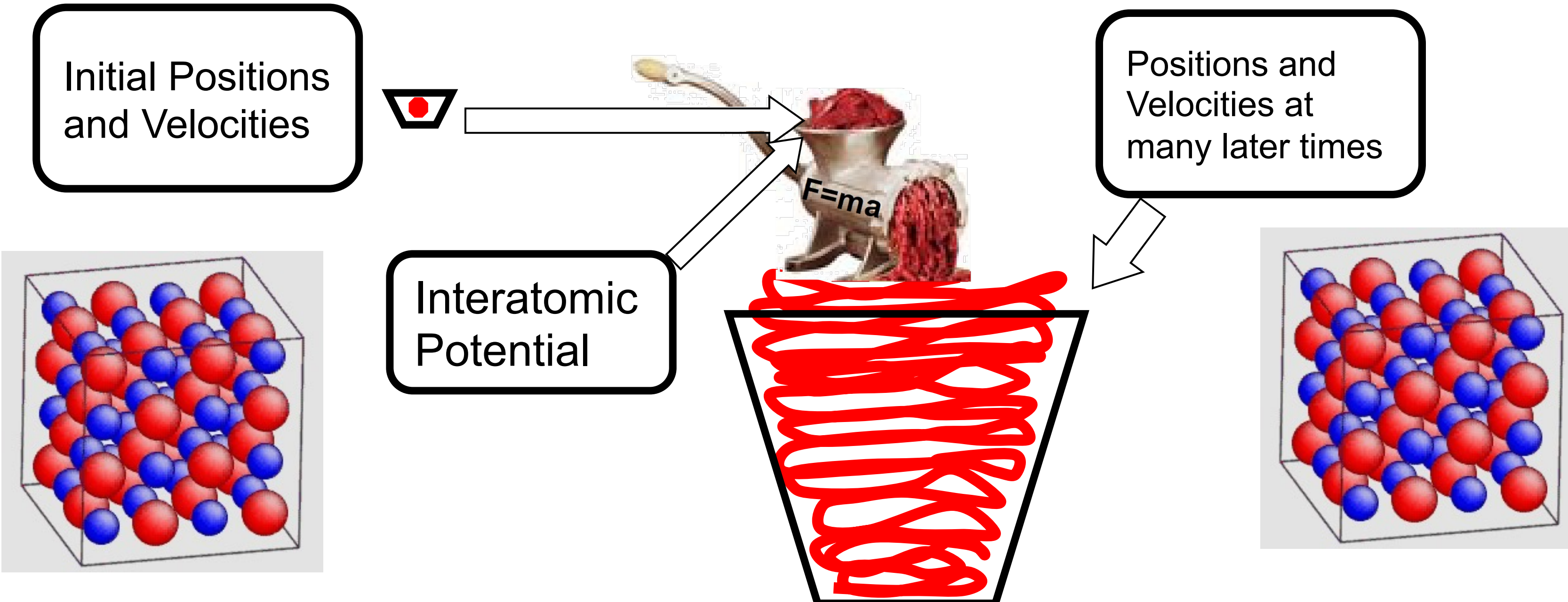
- For some flyers and wire vaporization experiments (e.g. Sandia's Z Machine) the expanding material enters the **liquid-vapor coexistence region**
- Most continuum hydrodynamics codes use **equilibrium equations of state**: assumes phase transformation kinetics are short compared to the dynamics of the simulation
- However, if **liquid-vapor transformation kinetics are long** compared to the simulation dynamics, then once material enters these two-phase regions, the simulation is **no longer valid**



Why Atomistic?

- **Atomistic simulations** (e.g. molecular dynamics) avoid **explicit assumptions** about the material behavior in the liquid-vapor coexistence region
- Accurately capture droplet formation, coalescence, break-up, surface tension, heat transfer, etc., **without approximations** commonly required for continuum models
- The goal of this work is to help **provide a basis for two-phase equations-of-state models in hydrocode simulations** of free expansion (e.g. exploding wires)
- Disadvantages of MD over continuum models: **computationally expensive, smaller length and time scales** (but gap can be partially closed with large supercomputers)

Molecular Dynamics: What is it?



Mathematical Formulation

- Classical Mechanics
- Atoms are Point Masses: $r_1, r_2, \ldots\ldots r_N$
- Positions, Velocities, Forces: r_i, v_i, F_i
- Potential Energy Function = $V(r^N)$
- $6N$ coupled ODEs

Newton's Equations:

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i$$

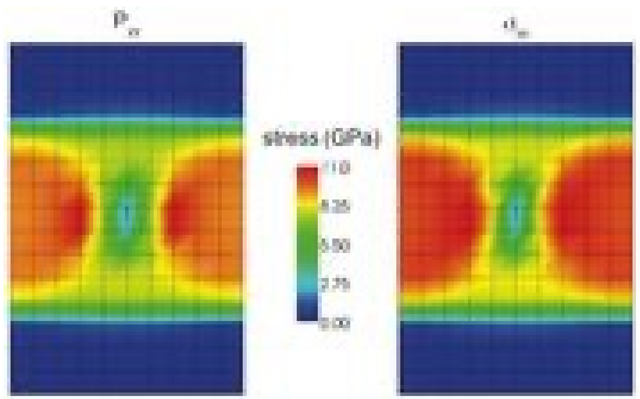
$$\frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{F}_i}{m_i}$$

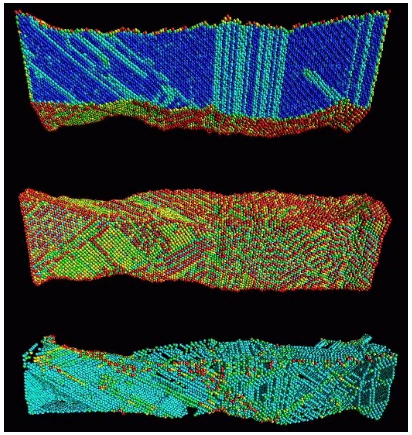
$$\mathbf{F}_i = -\frac{d}{d\mathbf{r}_i} V\left(\mathbf{r}^N\right)$$

Sandia National Laboratories

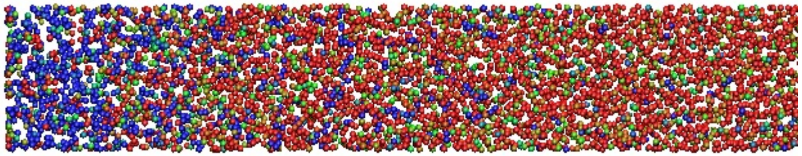
MD Versatility

Coupling to Solid Mechanics



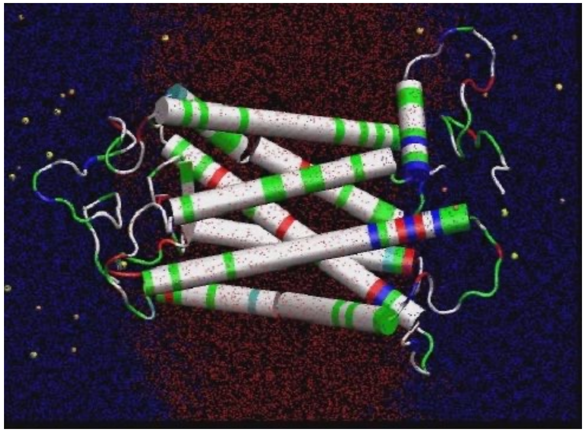


Materials Science: metals, polymers, etc.

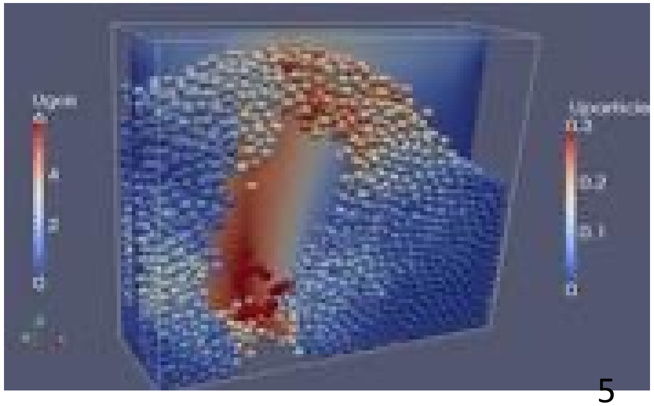


Chemistry

Biophysics

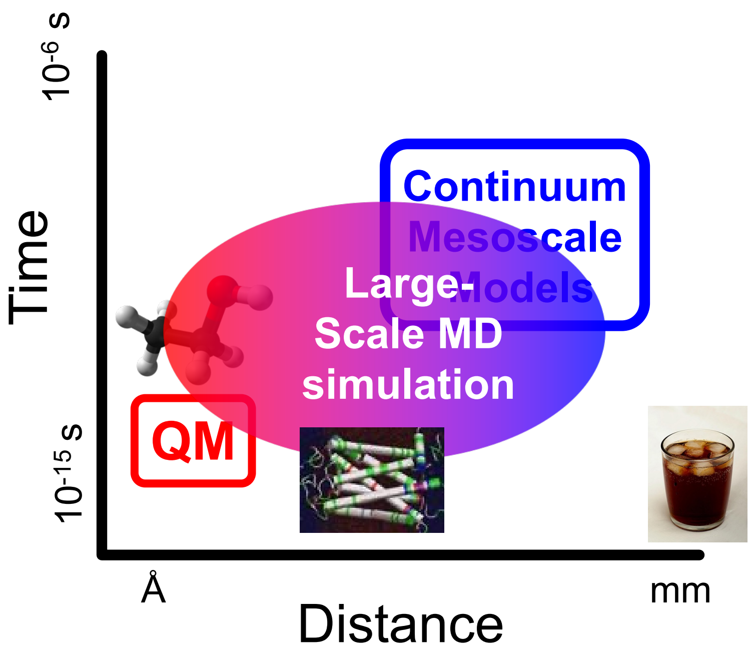


Granular Flow



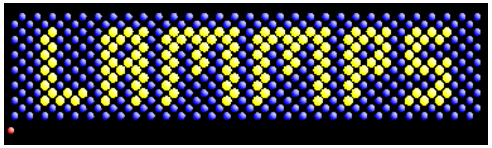
MD Time & Length Scales

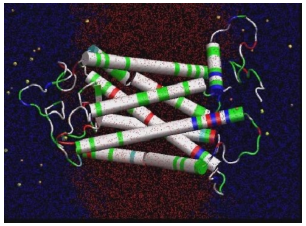
- Quantum mechanical electronic structure calculations (QM) provide accurate description of mechanical and chemical changes on the atom-scale, but limited to ~1000 atoms
- Atom-scale phenomena drive a lot of interesting physics, chemistry, materials science, mechanics, biology…but it usually plays out on a much larger scale
- Mesoscale: much bigger than an atom, much smaller than a glass of soda
- QM and continuum/mesoscale models (CM) can not be directly compared—**large scale MD can bridge gap**

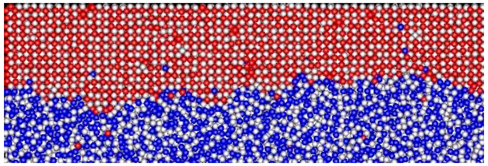


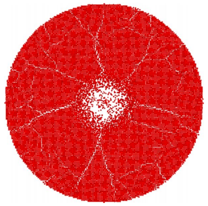
LAMMPS Code Overview

- Large-scale Atomic/Molecular Massively Parallel Simulator
- https://lammps.org
  - Open source, C++ code
  - Bio, materials, mesoscale





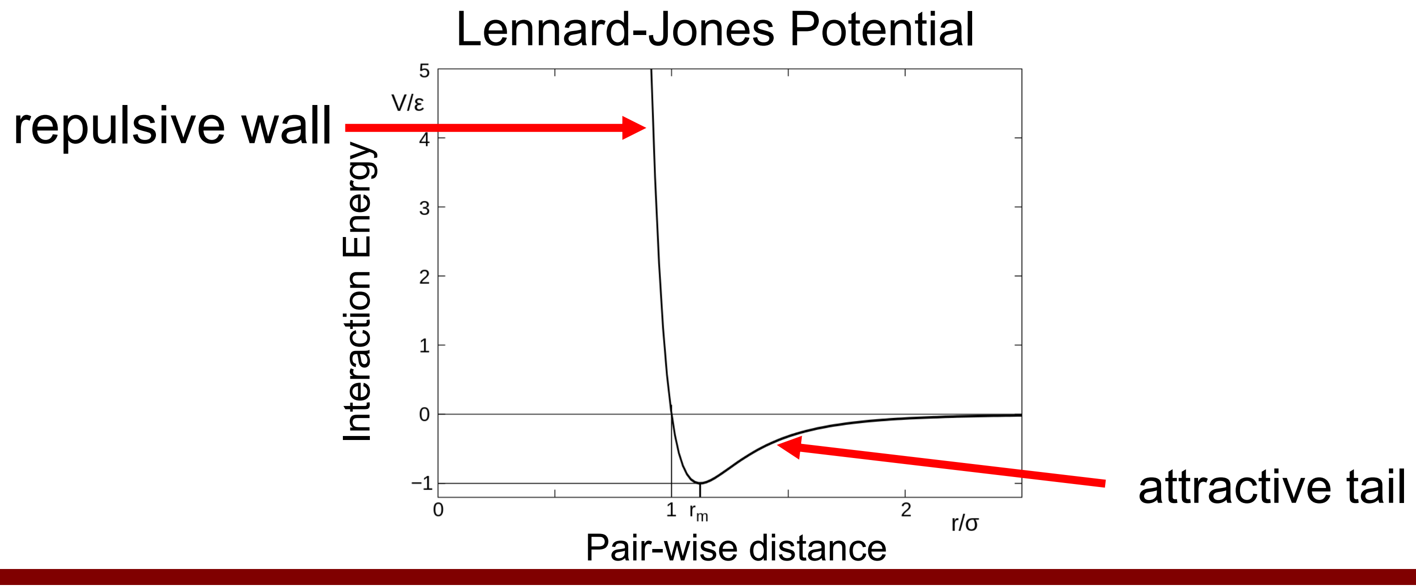




  - Particle simulator at varying length and time scales
    - Electrons → atomistic → coarse-grained → continuum
  - Spatial-decomposition of simulation domain for parallelism
  - Energy minimization, dynamics, non-equilibrium MD
  - GPU and OpenMP enhanced, Kokkos enabled
  - Can be coupled to other scales: QM, kMC, FE, CFD, …

Interatomic Potentials

- Quantum chemistry: solves Schrödinger equation (electron interactions) to get forces on atoms. Accurate but very computationally expensive and only feasible for small systems: ~1000 atoms
- Molecular dynamics: uses empirical force fields, sometimes fit to quantum data. Not as accurate but **much** faster
- MD typically only considers pair-wise or three-body interactions, scales as *O*(N) (billion atom simulations are considered huge)



SNAP Training Workflow

https://github.com/FitSNAP/FitSNAP

Model Form

- Energy of atom i expressed as a basis expansion over K components of the bispectrum (B_k^i)

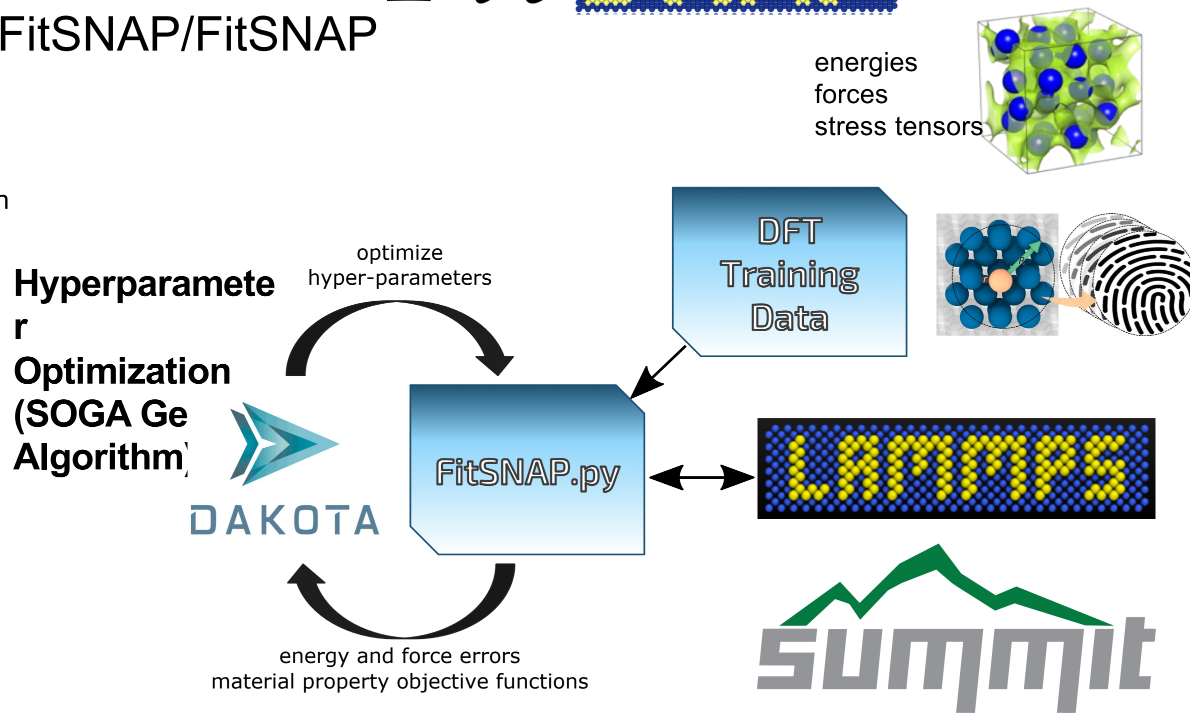
$$E_{SNAP}^i = \boldsymbol{\beta} \cdot \mathbf{B}^i + \frac{1}{2}(\mathbf{B}^i)^T \cdot \boldsymbol{\alpha} \cdot \mathbf{B}^i$$

Regression Method

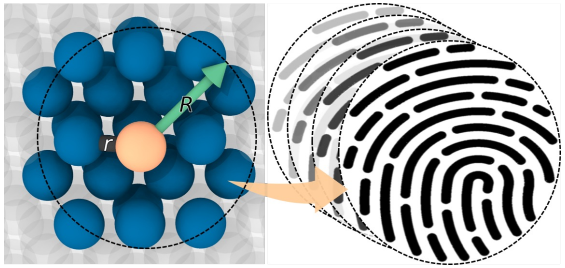
- β vector fully describes a SNAP potential
- Decouples MD speed from training set size

$$\min(||\mathbf{w} \cdot D\boldsymbol{\beta} - T||^2 - \gamma_n \, ||\boldsymbol{\beta}||^n)$$

Weights Set of Descriptors DFT Training Regularization Penalty



SNAP Bispectrum Components



- Neighbors of each atom are mapped onto unit sphere in 4D

$$3D\ Ball: (r, \theta, \phi), r < R_{cut} \implies 4D\ Sphere: (\theta_0, \theta, \phi), \theta_0 = \frac{r}{R_{cut}}\pi$$

- Expand density around each atom in a basis of 4D hyperspherical ***harmonics***,

$$\rho_i(\mathbf{r}) = \delta(\mathbf{0}) + \sum_{r_{i'} < R_{cut}} f_c(r_{i'}) w_{i'} \delta(\mathbf{r}_{i'})$$

- Bispectrum components of the 4D hyperspherical harmonic expansion are used as the geometric descriptors of the local environment
  - Preserves universal physical symmetries
  - Rotation, translation, permutation
  - Size-consistent (extensible)

$$u^j_{m,m'} = U^j_{m,m'}(0,0,0) + \sum_{r_{ii'} < R_{cut}} f_c(r_{ii'}) w_i U^j_{m,m'}(\theta_0, \theta, \phi)$$

$$B_{j_1,j_2,j} = \sum_{m_1,m_1'=-j_1}^{j_1} \sum_{m_2,m_2'=-j_2}^{j_2} \sum_{m,m'=-j}^{j} (u^j_{m,m'})^* H^{jmm'}_{\substack{j_1 m_1 m_1' \\ j_2 m_2 m_2'}} u^{j_1}_{m_1,m_1'} u^{j_2}_{m_2,m_2'}$$

- Deeply nested loops
- Loop structure not regular
- Loop sizes <= 14

Team for LAMMPS/SNAP GPU Optimizations

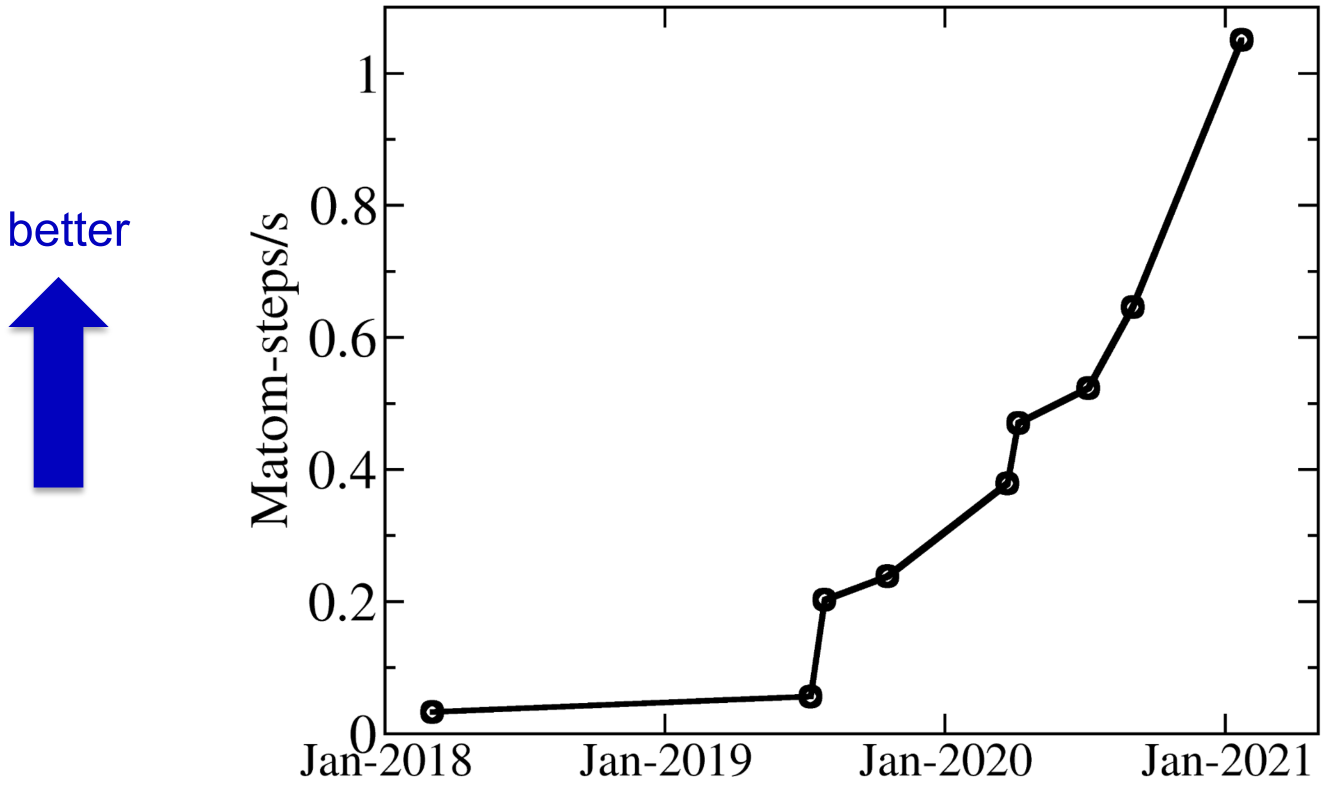
- **Stan Moore** (SNL): LAMMPS Kokkos lead developer, lynchpin for integrating Kokkos improvements into public LAMMPS (ported and reviewed code), benchmarked LAMMPS on pre-exascale testbeds
- **Aidan Thompson** (SNL), **Nick Lubbers** (LANL): algorithm redesign
- **Evan Weinberg** (NVIDIA): Major performance improvements on GPUs
- **Rahul Gayatri** (NERSC) and **Neil Mehta** (NERSC): performance improvements, support for TestSNAP and LAMMPS on pre-exascale testbeds, developing Kokkos OpenMPTarget backend
- **Nick Curtis** (AMD): Profiling SNAP on MI250X, Kokkos HIP backend improvements, investigating SNAP performance
- **Chris Knight** (ALCF) and **Yasi Ghadar** (ALCF): support for TestSNAP and LAMMPS on pre-Aurora testbeds
- **Daniel Arndt** (ORNL): developing Kokkos SYCL backend, helped tune TestSNAP performance on Arcticus

SNAP Improvements

- **Adjoint refactor**: algorithmic redesign that reduced the computational complexity and memory footprint by large factor
- **Flattened jagged multi-dimensional arrays:** reduced memory use
- **Major kernel refactor:** Broke one large kernel into many smaller kernels, reordered loop structure
- **Changed the memory data layout** of an array between kernels via transpose operations
- **Refactored loop indices and data structures** to use complex numbers and multi-dimensional arrays instead of arrays of structs
- Refactored some kernels to **avoid thread atomics** and use of **global memory**
- Judiciously used **Kokkos hierarchical parallelism** and **GPU shared memory**
- **Fused** a few selected **kernels**, which helped eliminate intermediate data structures and reduced memory use
- Added an AoSoA **memory data layout** inspired by Cabana code, which enforced perfect coalescing and load balancing in one of the kernels
- **Symmetrized data layouts** of certain matrices, which reduced memory overhead and use of thread atomics on GPUs (also improved CPU performance)
- Large refactor of Wigner matrices + derivatives to **use AoSoA data layout**
- **Pack several 32-bit integers** for Clebsch-Gordon coefficient lookup tables **into 128-bit int4 structs** and use 128-bit load/store to **reduce memory transactions**

SNAP Performance on V100

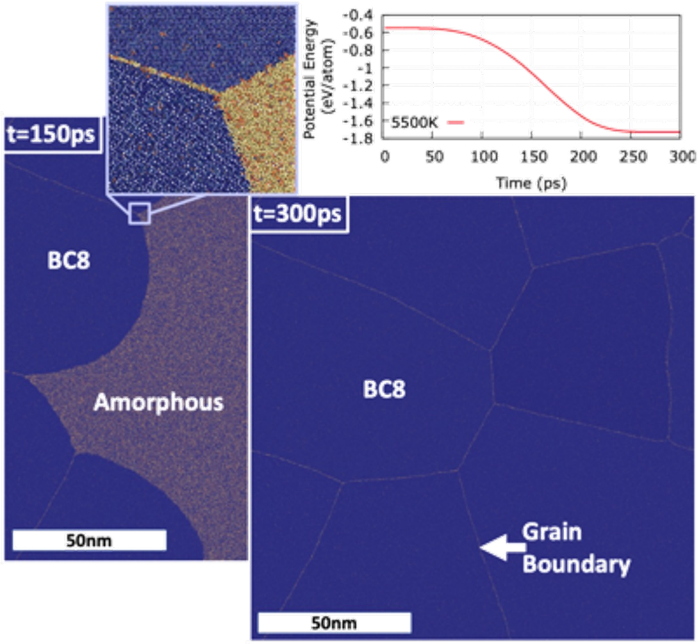
- Over 30x speedup since 2018!



- A few additional % speedup from recent improvement not shown

2021 ACM Gordon-Bell Award Finalist

- **"Billion atom molecular dynamics simulations of carbon at extreme conditions and experimental time and length scales"**
- SNAP model of carbon
- Team members from Sandia, U of S. Florida, NVIDIA, NERSC, and KTH
- Ran SNAP carbon model on full OLCF Summit (27,900 GPUs)
- Achieved **50.0 PFLOPs: 24.9% of Summit theoretical peak**, 33.6% of measured LINPACK benchmark
- SNAP MD simulation rate **22.9x higher** than DeepMD (2020 Gordon-Bell award for quantum-accurate MD)

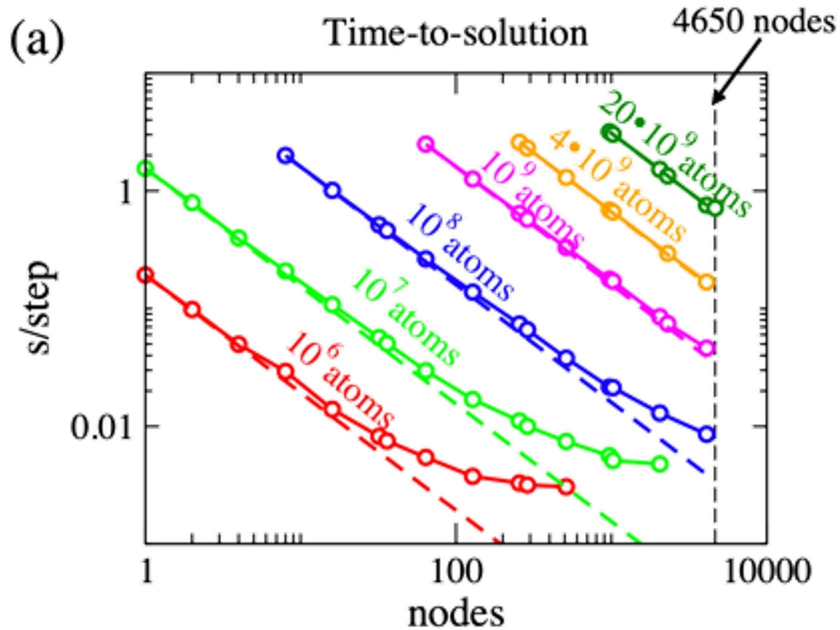


K. N. Cong, J. T. Willman, S. G. Moore, A. B. Belonoshko, R. Gayatri, E. Weinberg, M. A. Wood, A. P. Thompson, I. I. Oleynik, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '21), Article 4, 1–12, 2022.

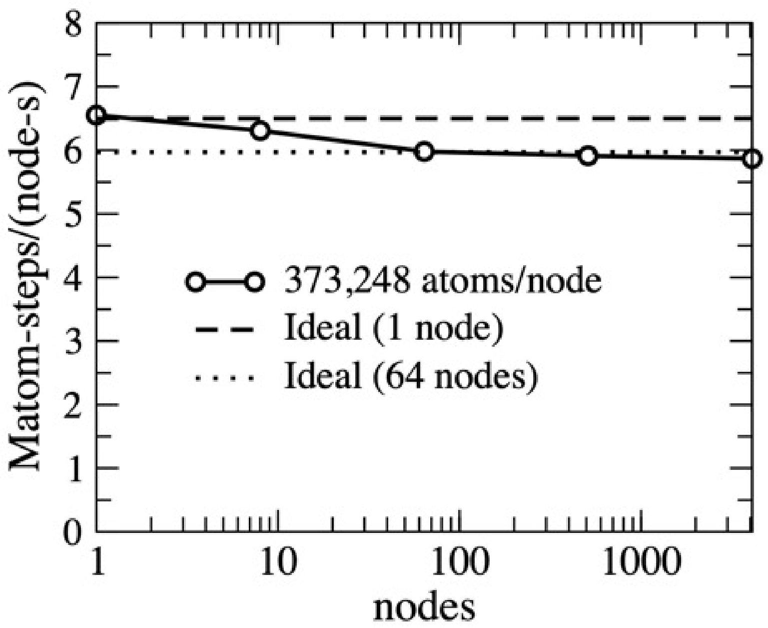
OLCF Summit Scaling Results

Benchmarked up to 20 billion atoms (amorphous carbon sample)

strong scaling



weak scaling



K. N. Cong, J. T. Willman, S. G. Moore, A. B. Belonoshko, R. Gayatri, E. Weinberg, M. A. Wood, A. P. Thompson, I. I. Oleynik, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '21), Article 4, 1–12, 2022.

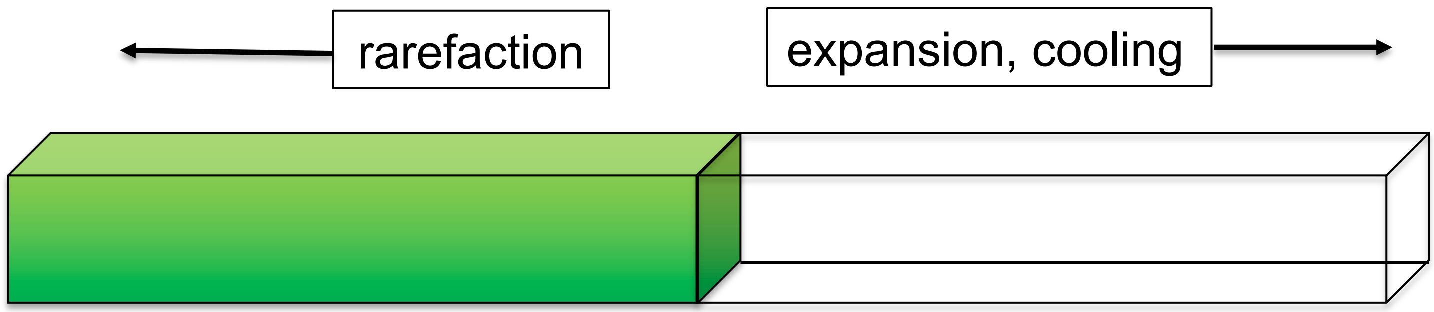
NNSA's ATS-2 Sierra Supercomputer

Sandia National Laboratories

- Hardware similar to OLCF Summit, but fewer GPUs per node
- 4320 nodes, 4 V100-16GB GPUs per node, IBM Power 9 CPUs
- At one point was #3 on the TOP500 supercomputer list, now #6 (as of November 2022)
- Located at Lawrence Livermore National Laboratory in California

Problem: Free Expansion

- Supercritical fluid expands into vacuum
- Supercritical means the material is so hot that there is no longer a clear distinction between the liquid and vapor phases
- When the supercritical fluid expands, the temperature drops below the critical temperature, and the fluid rapidly phase-separates into liquid droplets and vapor bubbles
- Rarefaction wave travels in opposite direction of expansion, limits maximum timescale of simulation



Early Work: Lennard-Jones

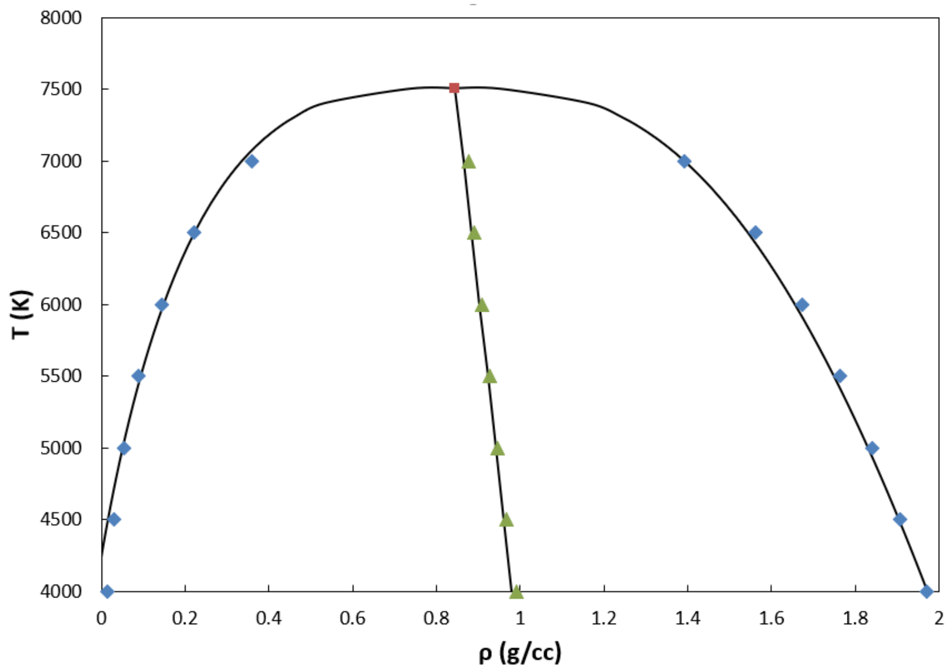
- The Lennard-Jones (LJ) interatomic potential is a simple empirical model that still captures many relevant physics phenomena of materials (~argon)
- LJ enables rapid throughput with large atom counts
- Investigated free expansion by running up to **~24 billion atoms** on 8192 GPUs on Sierra
- LJ is computationally very cheap: simulation size is limited by GPU memory on Sierra
- However, need a realistic model for metal: **develop SNAP machine learning potential for aluminum**
- SNAP model much more expensive: simulation size is limited more by time stepping throughput (i.e. number of compute days allocated on full machine)

Training the Al SNAP Model

- Density function theory (DFT) used as “ground truth” training data
- Normand Modine (SNL) generated DFT data using VASP code
- Training set included ~800,000 configurations!
- Bulk Al structures at a range of densities and temperatures (1.2–3.0 g/cc, 933–10,000 K)
- Freely expanding Al slabs at the same range of temperatures
- Ember Sikorski (SNL) optimized SNAP hyperparameters and generated model candidates using DAKOTA and FitSNAP

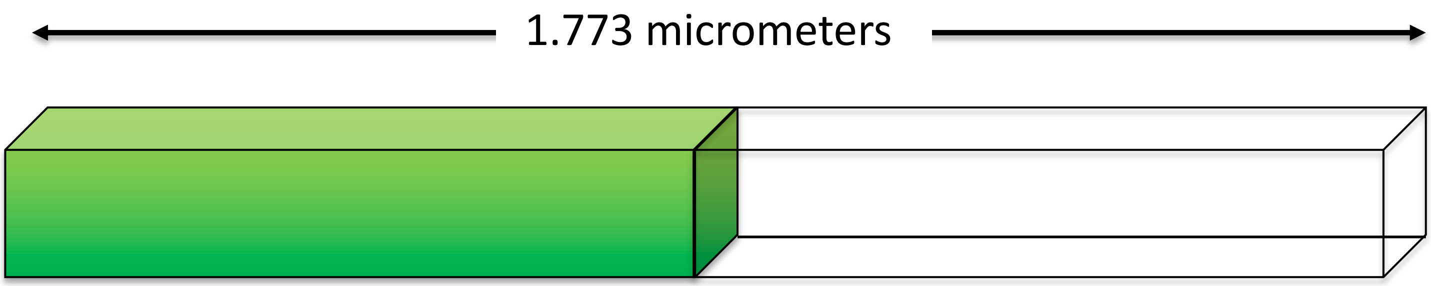
Evaluating Model Candidates

- Multiple SNAP model candidates were generated
- Candidates were evaluated by running small (~30k atom) simulations at different temperatures to map out the liquid-vapor coexistence region
- Critical temperatures and densities were fit using the universal Ising critical exponent β ≈ 0.326 and law of rectilinear diameter
- The predicted critical point was compared to values from experiment and theory
- One of the best candidates was then selected to run at large scale on Sierra



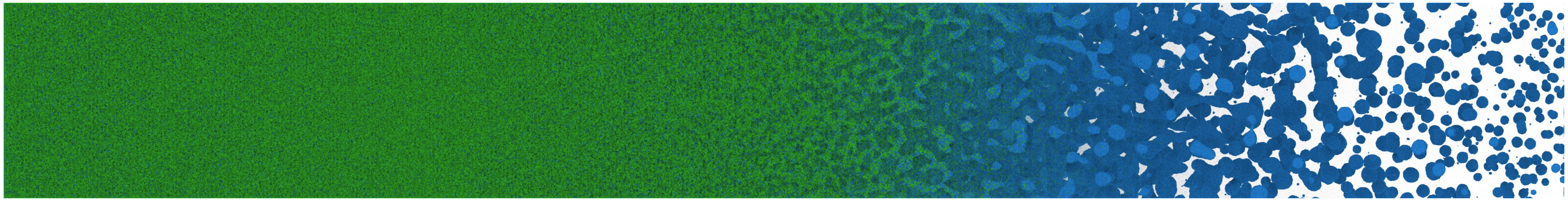
Aluminum Free Expansion

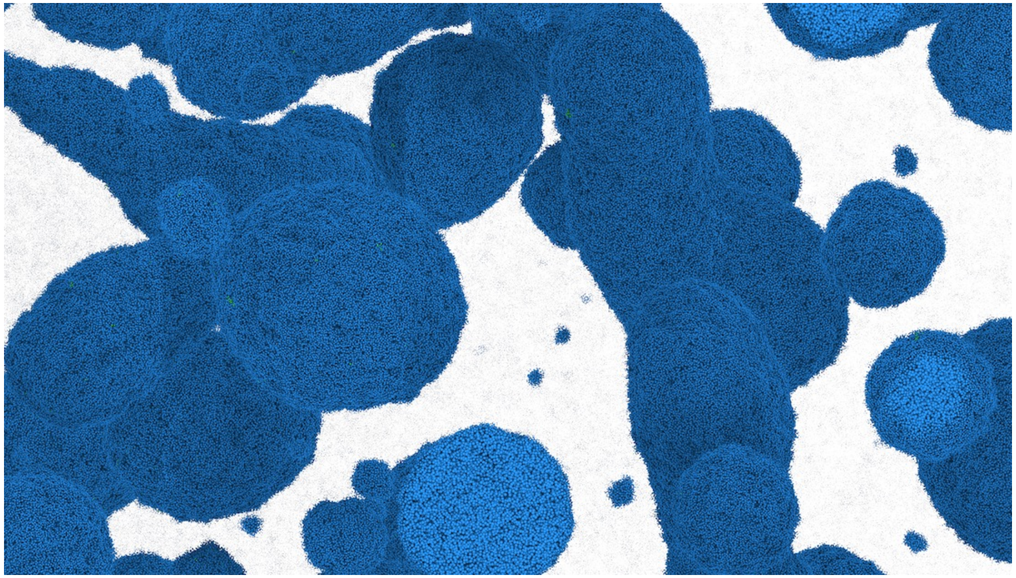
- ~1.5 billion atoms on 8192 GPUs (2048 nodes, ~47% of full Sierra), using KOKKOS package in LAMMPS
- Simulation is 1.773 micrometers long in the *x*-dimension, 8x smaller in the *y-z* dimensions
- Infinite periodic boundary conditions in *y-z* as well
- Simulation starts at a temperature of 9000 K and a density of 1.5 gm/cc (aluminum $T_c \approx 6500$ K)
- Supercritical fluid initially fills half the cell (~0.9 micron long) and then expands out
- 1 femtosecond timestep, ran for 0.56 nanoseconds total physical time

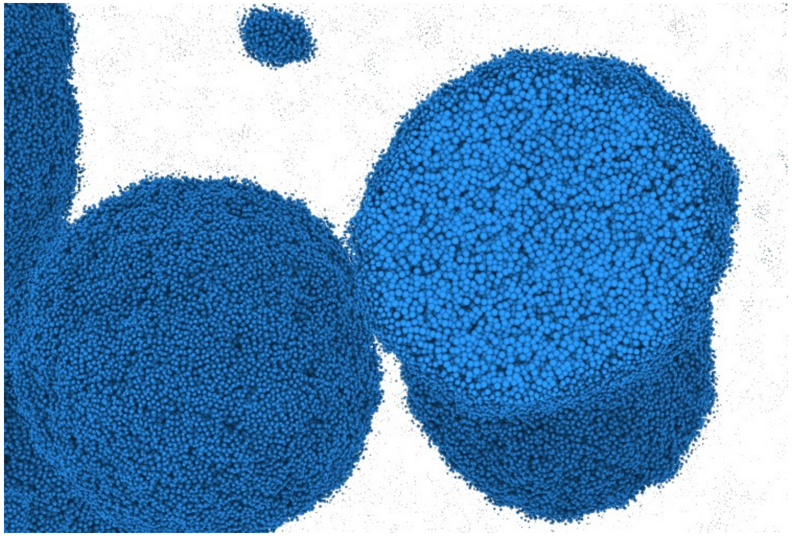


Simulation Visualization

- Particle size is rendered proportional to local density: effectively removes the vapor phase and leaves only liquid droplets
- Coloring: green = supercritical fluid, blue = subcritical liquid (approximate)







Visualization with OVITO

OVITO Advantages

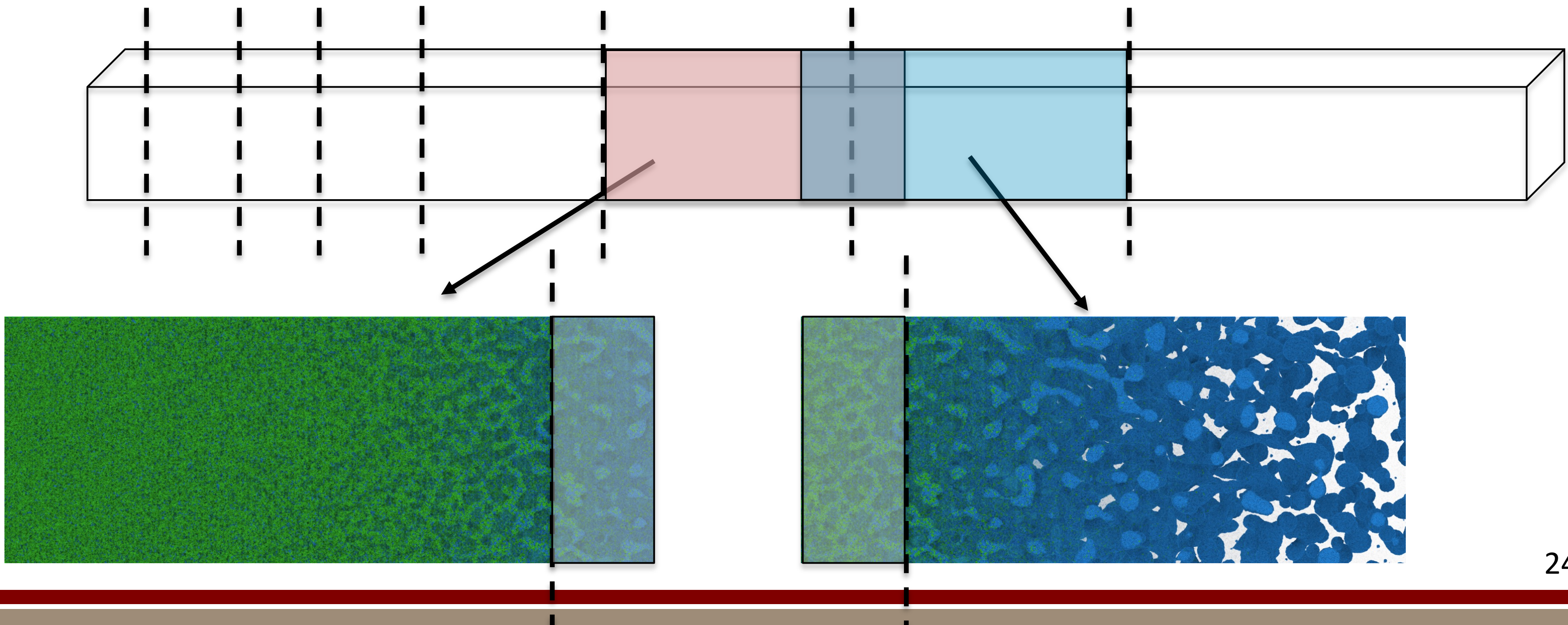
- Domain specific: highly optimized for particle simulations, has direct support for LAMMPS dump files
- Produces high quality visualizations with ray tracing, ambient occlusion, etc.
- Highly scriptable with Python and useful for data post-processing and analysis in general (in addition to rendering images)

OVITO Disadvantages

- Only runs on a single node with multithreading and shared memory, no MPI parallelization
- Can only visualize up to ~2 billion particles at a time (assuming unlimited memory)

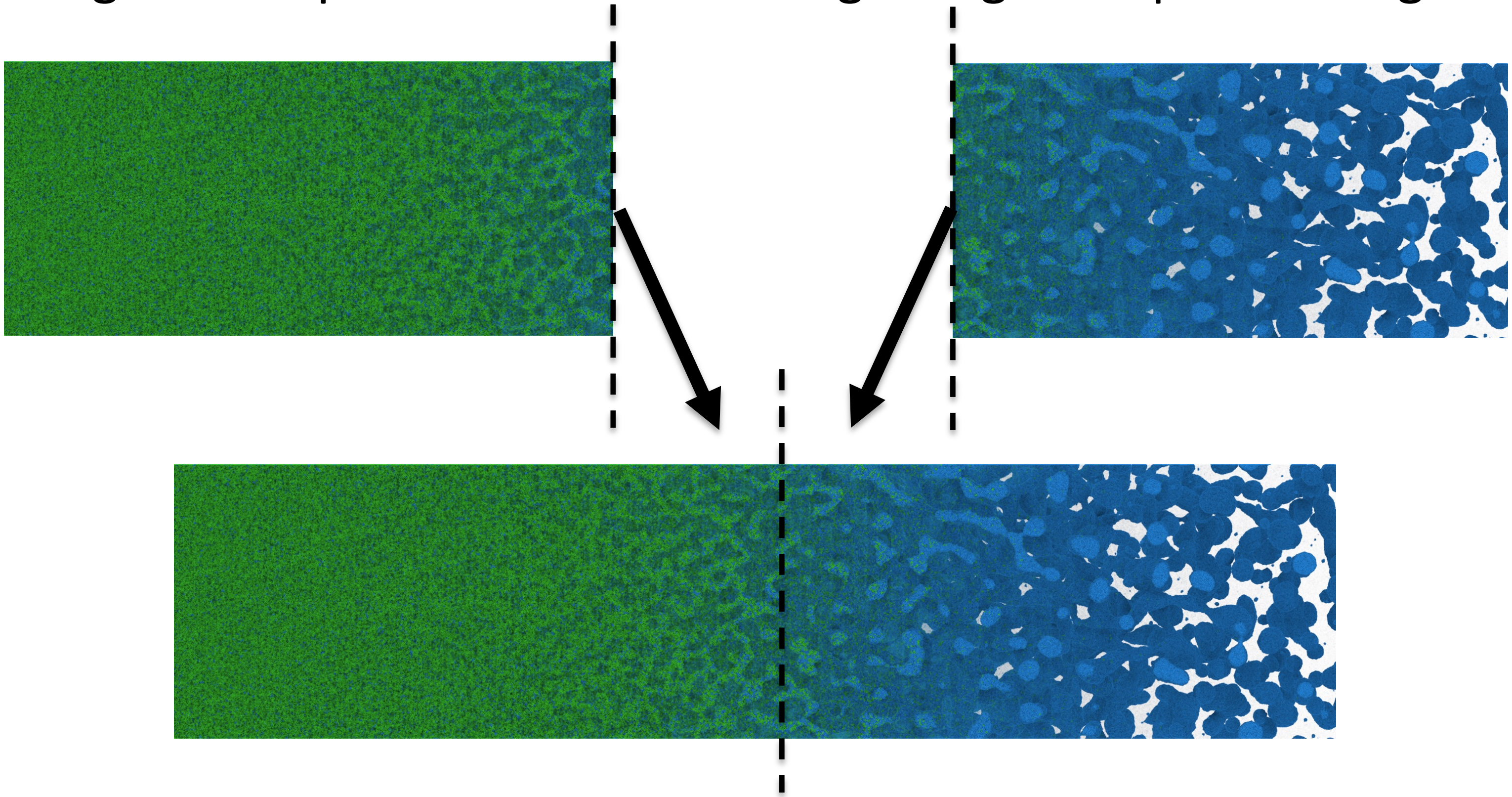
OVITO Parallelism Workaround

- LAMMPS reshuffles atom data between ranks so each rank has a "slice" of the simulation data in the x-direction
- Each rank outputs to a separate file (e.g. 8192 files total)
- MPI driver program launches separate instances of OVITO on many nodes
- Each OVITO instance loads atom data from "owned" slices, along with neighboring "ghost" slice data to create a buffer zone to reduce visual edge artifacts



OVITO Parallelism Workaround (cont.)

- OVITO renders an image of the slice, including buffer zone, thenthe buffer region is cropped off
- Another MPI driver program stiches all the small slice images together in parallel to create a single large composite image



OVITO Parallelism Workaround (cont.)

Advantages:

- Highly scalable: large images are rendered in an (almost) embarrassingly parallel manner
- Can render more than 2 billion atoms

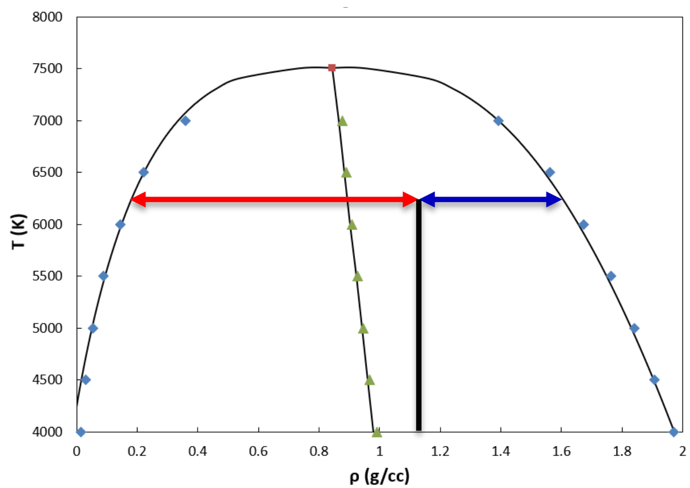
Disadvantages:

- Minor artifacts in lighting/shadows, but overall produces nice, usable images in parallel
- Can only visualize a single face straight on (so everything lines up), no 3D perspective views

- Would like to also try Paraview in the future (less domain specific, but MPI-enabled so requires less workarounds)

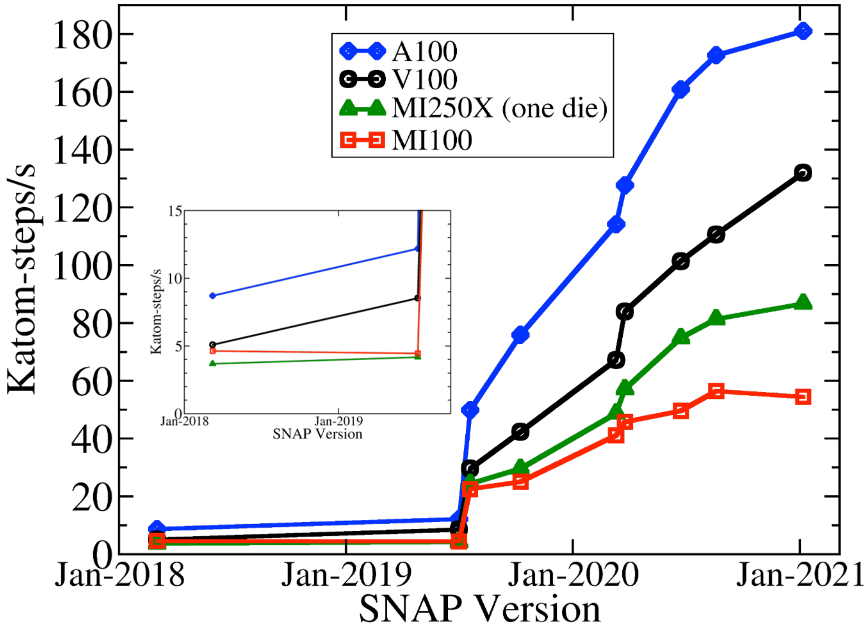
Free Expansion: Next Steps

- Performed some data analysis of earlier LJ simulations, but need to analyze new data from SNAP runs
- Investigate how well each slice in the simulation matches the phase fractions predicted from the equilibrium liquid-vapor tie line
- Can also analyze droplet size and growth rate, etc.
- Also started running a 4x longer simulation with SNAP aluminum = ~6 billion atoms: allows for a longer timescale before the rarefaction wave reaches the front edge of the box



Towards Exascale

- Ran older versions of SNAP on OLCF Crusher MI250X: NVIDIA improvements also helped MI250X (V100 was an excellent proxy for MI250X)
- ~25x performance improvement comparing original vs latest code for both V100 and MI250X (see inset), different benchmark than earlier slide
- Additional few % improvement using latest Kokkos, ROCm, and LAMMPS on MI250X beyond latest data point on plot



Measured MI250X/V100 FLOP ratio = ~2.4x, but performance is only 0.7x, so we lost 3.4x, why?

SNAP Kernels and Limiters

V100:

- **ComputeYi**: 66% of runtime, L1 cache bandwidth bound
- **ComputeFusedDeidrj**: 27% of runtime, FP64 compute bound
- **ComputeUi**: 3% of runtime, FP64 atomic-add bound

MI250X:

- **ComputeYi**: 63% of runtime, currently VALU (int32) bound
- **ComputeFusedDeidrj**: 29% of runtime
- **ComputeUi**: 5% of runtime

INT32 Throughput Issue

- Nick Curtis (AMD) profiled ComputeYi (largest kernel in SNAP): **90% VALU int32 bound** on MI250X, only waiting on memory 10% of the time
- Kokkos SNAP uses up to **4D arrays** in deeply nested loops: large int32 computation to index Kokkos views
- ROCm 5.3.0 has improvement to AMD compiler to generate IMAD (integer fused multiply/add) operations[1], but no improvement in practice for SNAP
- Another compiler optimization in ROCm 5.4.0 is expected to help Kokkos MDRangePolicy (potential speedup unknown)
- V100 has **independent parallel integer and floating-point data paths**, so the Volta SM is efficient on workloads with a mix of computation and addressing calculations[2]

[1] https://reviews.llvm.org/D127253
[2] https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf, page 7

Frontier L1 Cache Size

AMD MI250X GPU (one die):

- 24 FP64 TFLOPS peak theoretical[1], **18.3 FP64 TFLOPS** measured with Kokkos Bytes & FLOPS benchmark, i.e. with limited power and cooling, using AMD internal optimizations for Kokkos[2], **18.9 TFLOPS** for raw HIP with same benchmark
- **L1-cache/SM = 16 KB** (fixed)

NVIDIA V100 GPU:

- 7.8 FP64 TFLOPS peak theoretical, **7.8 FP64 TFLOPS** measured **L1-cache/SM = 96 KB** (typical but can be changed)

- ComputeYi L1 cache hit rate: **~60% on MI250X, ~90% on V100**
- V100 performance **highly sensitive** to reducing L1 cache size (i.e. using *cudaFuncCachePreferShared* instead of *PreferL1*)

[1] https://www.amd.com/system/files/documents/amd-cdna2-white-paper.pdf
[2] https://github.com/kokkos/kokkos/pull/4755, AMD is working through how to implement these optimizations in public Kokkos

Conclusions

- Atomistic simulations can generate **unprecedented insight into phase change kinetics and fluid microstructure evolution** during free expansion
- Provide a basis for **improving two-phase equation-of-state models** in **hydrocode** simulations
- **Machine learning is a powerful tool** but still **requires humans in the loop** to evaluate model candidates and interpret the results
- SNAP machine learning potential in LAMMPS is **highly optimized for NVIDIA GPUs**
- Need **more profiling** to better understand and potentially mitigate AMD MPI250X performance bottlenecks (L1 cache size and int32 throughput) for **OLCF Frontier exascale supercomputer**

Thank you

Questions?