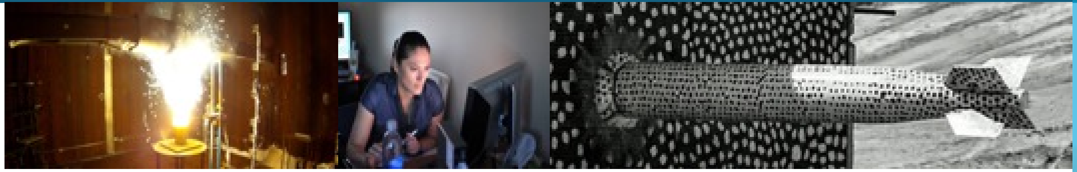
SAND2021-9785C

Sandia National Laboratories

The State of the LAMMPS KOKKOS Package



Stan Moore

Virtual LAMMPS Workshop and Symposium 2021

Supercomputer Hardware Trends

Currently, 6 of the top ten supercomputers use NVIDIA GPUs according to the June 2021 Top500 List (https://www.top500.org)

Future exascale supercomputers will have accelerators: ANL Aurora—Intel GPUs, ORNL Frontier—AMD GPUs)

Special code (beyond regular C++ and MPI in LAMMPS) is required to run well on GPUs (e.g. CUDA, HIP, SYCL); likely true for future hardware as well

Hardware and corresponding programming languages are ever changing, how to keep LAMMPS up to date?

http://www.nvidia.com/object/tesla-p100.html

Kokkos Performance Portability Library

Kokkos is an abstraction layer between programmer and next-generation platforms

Allows the same LAMMPS C++ code to run on multiple hardwares (GPU, Xeon Phi, etc.)

Kokkos consists of two main parts:

1. Parallel dispatch—threaded kernels are launched and mapped onto backend languages such as CUDA or OpenMP
2. Kokkos views—polymorphic memory layouts that can be optimized for a specific hardware

Used on top of existing MPI parallelization (MPI + X)

Open-source, can be downloaded at https://github.com/kokkos/kokkos

In a nutshell, the goal of Kokkos is to future-proof LAMMPS to allow it to run on future hardware without total re-write (i.e. add Kokkos backend for new hardware, no major changes to LAMMMPS)

LAMMPS KOKKOS Package

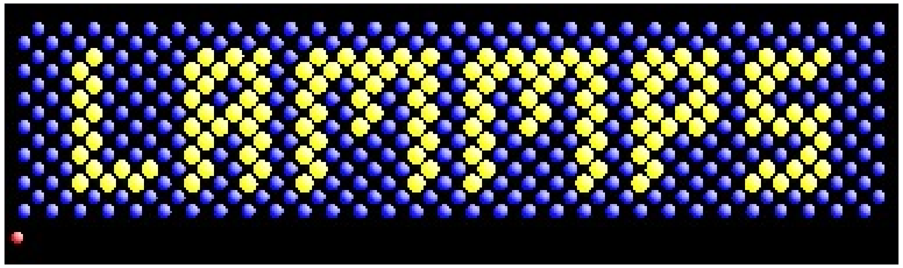
Optional add-on package in LAMMPS

Developed by Stan Moore, Christian Trott, and others

Goal is that everything in LAMMPS (pair, fixes, computes, etc.) runs on the GPU, with minimal data transfer from GPU to CPU if possible

Supports CUDA-aware MPI

Different than the GPU package, which only runs the pair-style and a few other computations on the GPU



LAMMPS KOKKOS Package

9 atom styles: angle, atomic, bond, charge, dpd, full, molecular, spin, sphere (along with hybrid)

44 pair styles: buck/coul/cut, buck/coul/long, buck, coul/cut, coul/debye, coul/dsf, coul/long, coul/wolf, dpd/fdt/energy, eam/alloy, eam/fs, eam, exp6/rx, gran/hooke/history, hybrid, hybrid/overlay, lj/charmm/coul/charmm/implicit, lj/charmm/coul/charmm, lj/charmm/coul/long, lj/class2/coul/cut, lj/class2/coul/long, lj/class2, lj/cut/coul/cut, lj/cut/coul/debye, lj/cut/coul/dsf, lj/cut/coul/long, lj/cut, lj/expand, lj/gromacs/coul/gromacs, lj/gromacs, lj/sdk, morse, multi/lucy/rx, reaxff, snap, sw, table, table/rx, tersoff, tersoff/mod, tersoff/zbl, vashishta, yukawa, zbl

23 fix styles: deform, dpd/energy, enforce2d, eos/table/rx, freeze, gravity, langevin, momentum, neigh/history, nph, npt, nve, nve/sphere, nvt, property/atom, qeq/reaxff, reaxff/bonds, reaxff/species, rx, setforce, shake, shardlow, wall/lj93, wall/reflect

3 compute styles: coord,/atom orientorder/atom, temp

3 bond styles: class2, fene, harmonic

4 angle styles: charmm, class2, cosine, harmonic

4 dihedral styles: charmm, class2, harmonic, opls

2 improper style: class2, harmonic

1 kspace style: pppm

Running KOKKOS Package

Kokkos library is already included with LAMMPS, no need to download

- Can compile with Makefiles or CMake
- Must use a c++14 compatible compiler (e.g. gcc 5.3.0 or higher, intel 17.0 or higher, CUDA 9.0 or higher)

No changes to input script needed, just add a few command line args:

- Run with 4 MPI tasks and 4 GPUs: "mpiexec -np 4 ./lmp_exe -in in.lj **-k on g 4 -sf kk**"
- Run with 4 OpenMP threads: "./lmp_exe -in in.lj **-k on t 4 -sf kk**"

See Kokkos docs: https://www.lammps.org/doc/Speed_kokkos.html, https://www.lammps.org/doc/package.html

Highlight: ReaxFF Reactive Potential

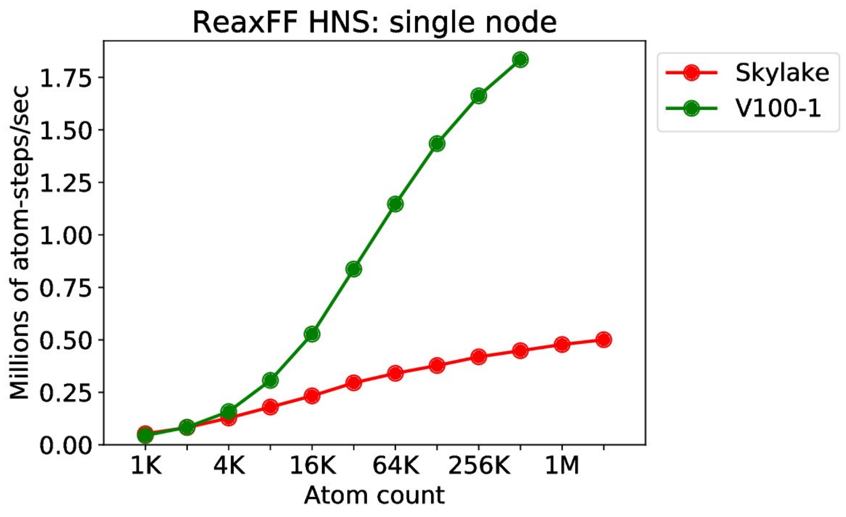
3 versions in LAMMPS:

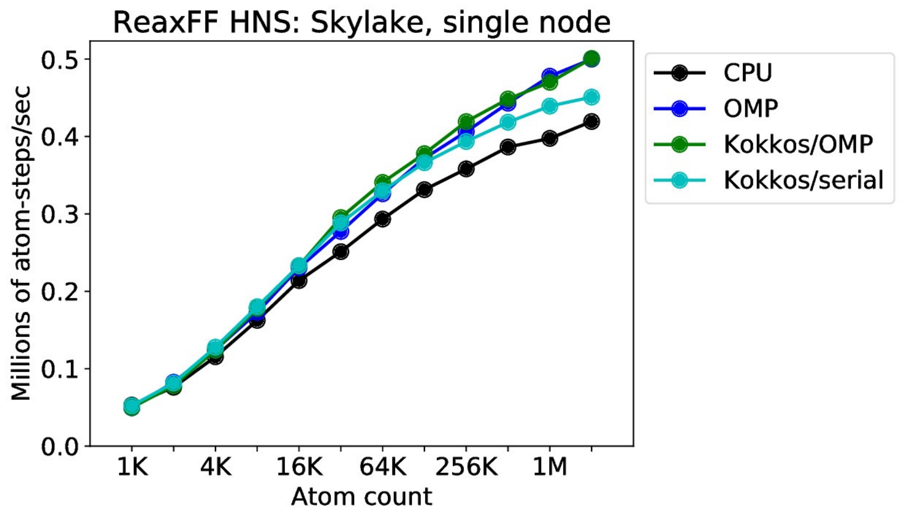
- REAXFF package
- KOKKOS package
- OPENMP package

KOKKOS version more memory robust, should be used if getting memory errors, or with fix GCMC

KOKKOS serial version faster than base REAXFF package, at least in some cases

KOKKOS CUDA version can run on NVIDIA GPUs

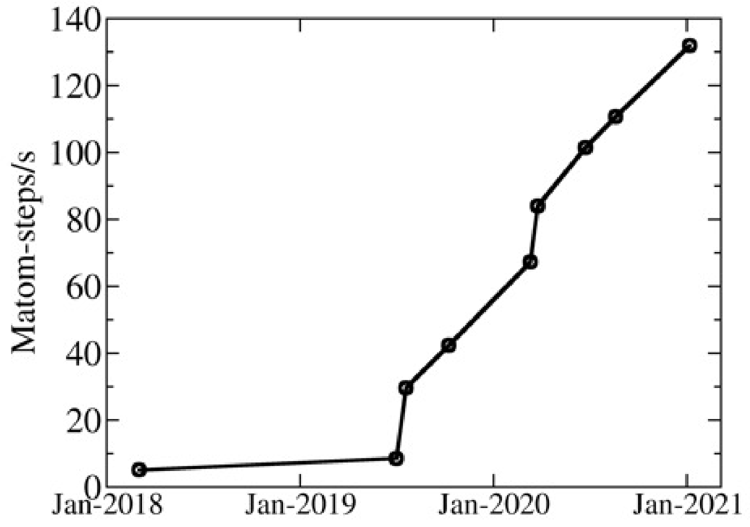


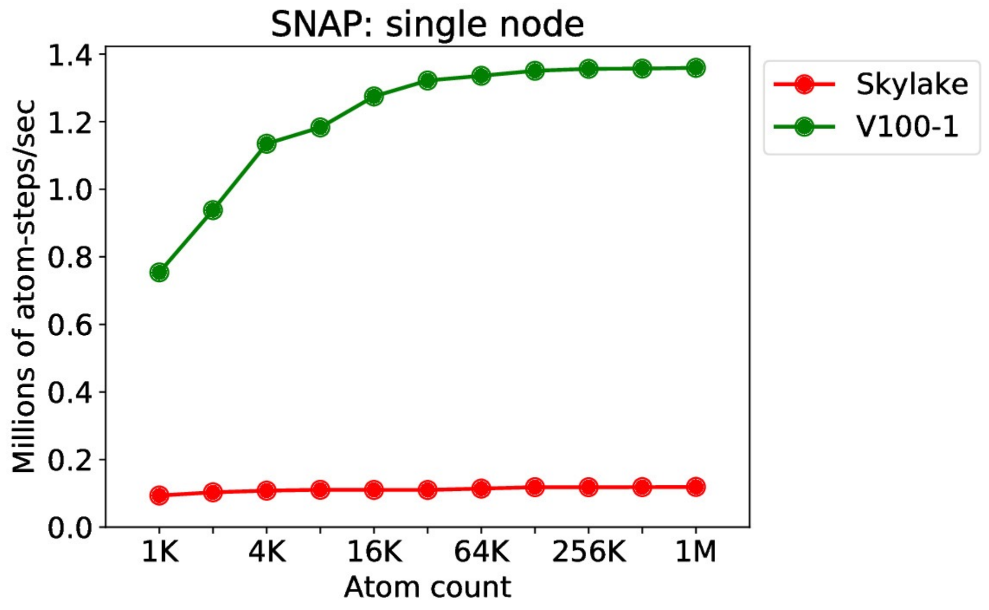


Highlight: SNAP Machine-Learning Potential

SNAP can achieve quantum-chemistry accuracy while still maintaining linear scaling of molecular dynamics

Kokkos CUDA version of SNAP has been extremely optimized, over 30x performance improvement on V100 GPU since 2018 for tungsten benchmark





Other Improvements

Recently added to Kokkos:

- **Fix Shake**—applies bond and angle constraints to small clusters of atoms (e.g. water) to allow larger timestep
- **"Hash" style atom-map**: needed for large parallel simulations with molecules to avoid running out of memory
- **Pair/only option**: easily mimics GPU package by only running pair-style on GPU

In progress:

- **New benchmarking page**: shows performance of all accelerator packages, as well as commands needed to duplicate performance on your own machine
- **Fix Rigid**: rigid bodies

Limitations of the Kokkos package

If a style isn't in the KOKKOS package, it won't be accelerated. Also may need to transfer atom data back and forth between CPU and GPU every timestep, which reduces performance

USER-INTEL, USER-OMP, and OPT packages often give better vectorization on Intel hardware leading to better performance

GPU and USER-INTEL packages support single and mixed precision, KOKKOS package only supports double precision (but planning to fix this in the future)

Conclusions

Computer hardware is becoming more complicated, requiring special code to run well

Kokkos library: goal is performance portability for current and future hardware

LAMMPS KOKKOS package allows LAMMPS to run on NVIDIA, AMD, and Intel GPUs

Give KOKKOS package a try

Post questions, issues, or feature requests to the LAMMPS mail list, happy to help