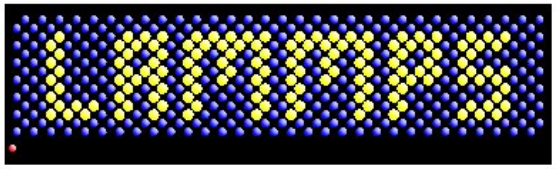
Lecture #5a – How LAMMPS works internally

Steve Plimpton
Sandia National Labs
sjplimp@sandia.gov

7th LAMMPS Workshop Tutorial
Virtual meeting – August 2021



Center for Computing Research

Sandia National Laboratories

Sandia National Laboratories is a multi-mission 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-NA0003525.

National Nuclear Security Administration

Goals for this 10 minute lecture

Describe ...

- Basic data structures used within LAMMPS
- Basic algorithms used by LAMMPS
- LAMMPS can be a standalone code or library

Understanding the basics of how LAMMPS works
can help you use it more effectively

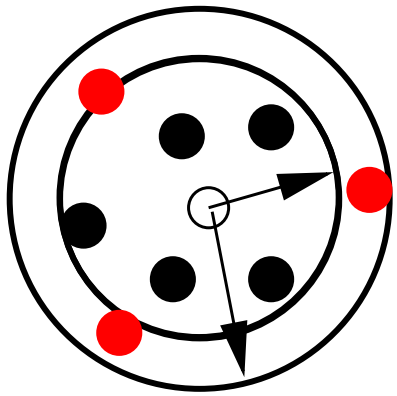
Per-atom data

- Choice of atom style ⇒ properties stored by atoms
  - vectors: type, charge, atom ID, molecule ID, etc
  - arrays: coords, forces, dipole moments, etc
- See atom_style doc page

angle	bonds and angles	bead-spring polymers with stiffness
atomic	only the default values	coarse-grain liquids, solids, metals
body	mass, inertia moments, quaternion, angular momentum	arbitrary bodies
bond	bonds	bead-spring polymers
charge	charge	atomic system with charges
dielectric	dipole, area, curvature	system with surface polarization
dipole	charge and dipole moment	system with dipolar particles
dpd	internal temperature and internal energies	DPD particles

Neighbor lists

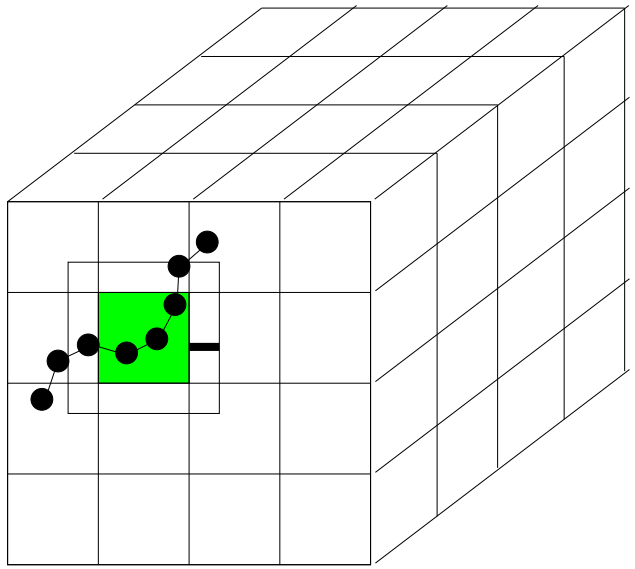
- Each atom stores vector of indices of M nearby atoms
  - within a spherical cutoff distance



- All pair styles use one or more neighbor lists (half or full)
- Fixes and computes can request additional neighbor lists
  - compute centro/atom: local crystal struct with full list
  - compute RDF: list with longer cutoff than pair style
- Logfile enumerates neighbor lists being used
- Largest memory footprint in typical LAMMPS simulation

Spatial parallelization in LAMMPS

- Simulation box divided into 3d bricks
- One brick per processor (MPI task)

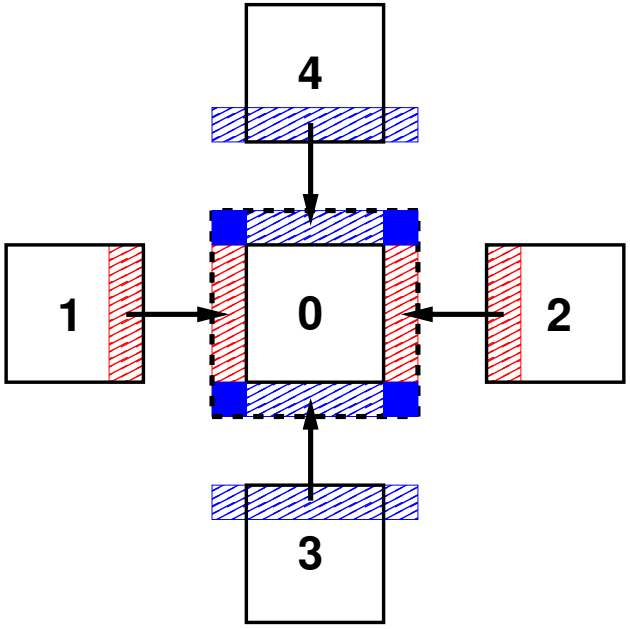


- Per-atom data and neigh lists distributed same way
- Proc only stores owned atoms + ghost atoms +
  neighbor list for owned+ghost atoms
- Atoms carry properties & molecule topology as they migrate

Spatial decomposition ⇒ parallel communication

Communication of ghost atoms within cutoff:

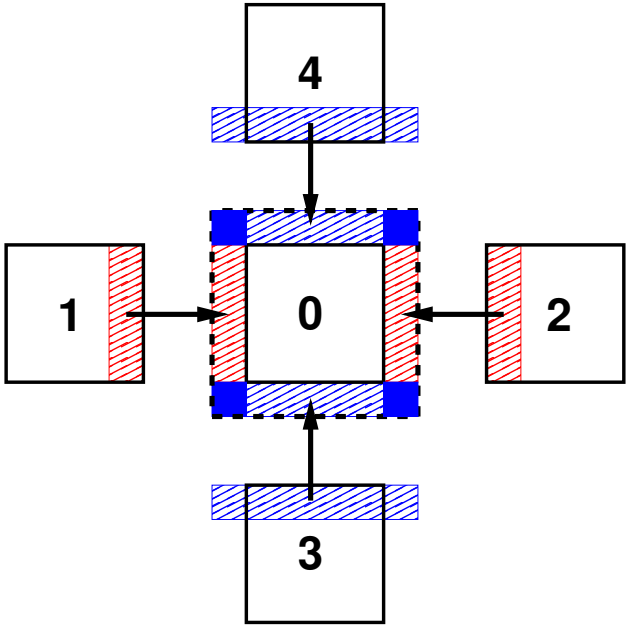
- 6-way stencil (4-way in 2d)
- forward/reverse comm of coords/forces
- also used by pair/fix/compute styles



Spatial decomposition ⇒ parallel communication

Communication of ghost atoms within cutoff:

- 6-way stencil (4-way in 2d)
- forward/reverse comm of coords/forces
- also used by pair/fix/compute styles

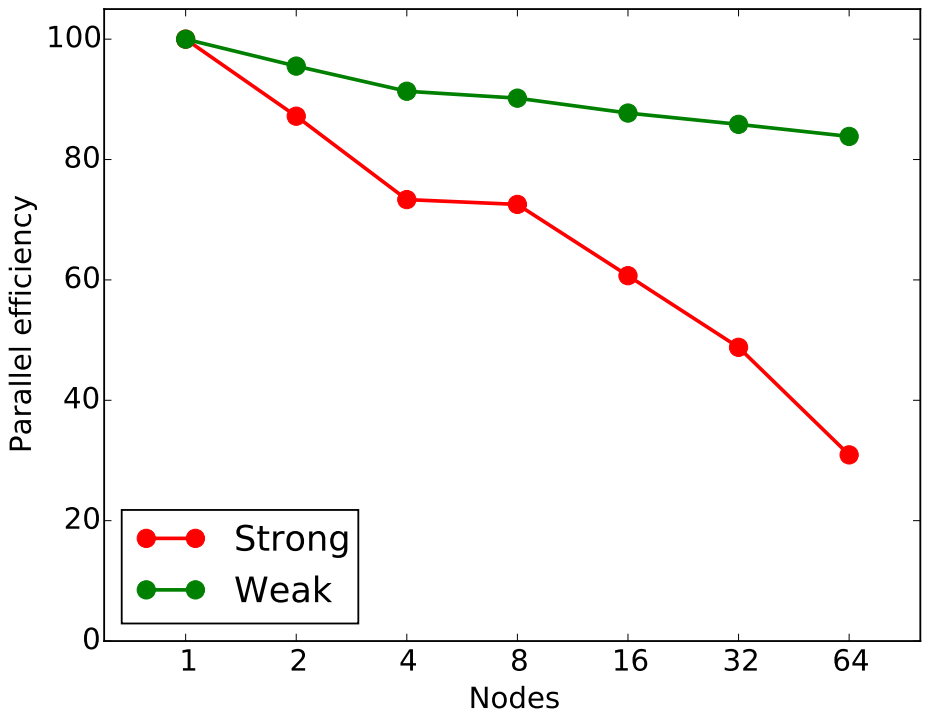


Models with short-range forces

- Computation:
  - forces scale as $O(N/P)$
  - neigh list scales as $O(N/P)$
- Communication:
  - scales as $O(N/P)^{2/3}$

Parallel performance

- Strong scaling: Same 128K atoms as nodes increase
- Weak scaling: Same 128K atoms/node as nodes increase
- Intel Skylake CPU cluster: 1 node = 36 cores



- Plotted so perfect efficiency would be horizontal line
- Strong scaling degrades below few 100 atoms/proc
- Possible to weak scale to many 1000s of nodes

LAMMPS is also a library

- Most people use it as a stand-alone code
- Using it as a library can also be useful

1. Enables direct coupling to another code
   - multiphysics or multiscale simulations
2. Enables use of LAMMPS in a workflow with other codes
   - increasingly common in machine learning applications
3. Enables launching & using LAMMPS from a Python script
   - use mpi4py module to run Python in parallel

LAMMPS is also a library

- Most people use it as a stand-alone code
- Using it as a library can also be useful

1. Enables direct coupling to another code
   - multiphysics or multiscale simulations
2. Enables use of LAMMPS in a workflow with other codes
   - increasingly common in machine learning applications
3. Enables launching & using LAMMPS from a Python script
   - use mpi4py module to run Python in parallel

See Section 2 of Programmers Guide
Richard Berger talk in Fri developer sesson on
variety of ways to use LAMMPS and Python together