Delving deeper into LAMMPS

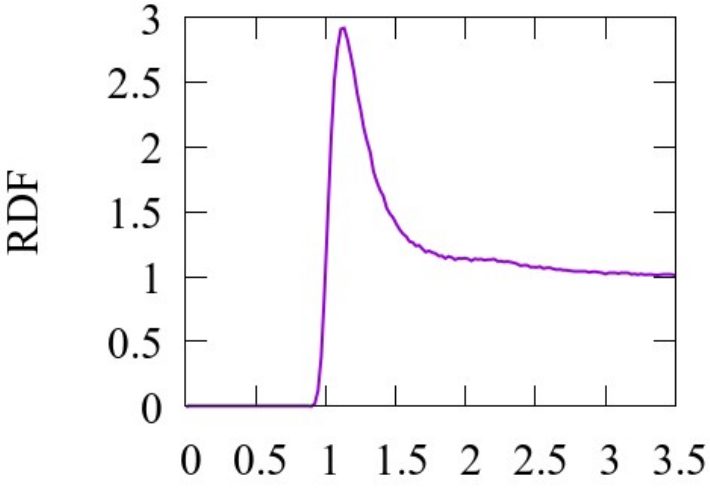
Reusing this material

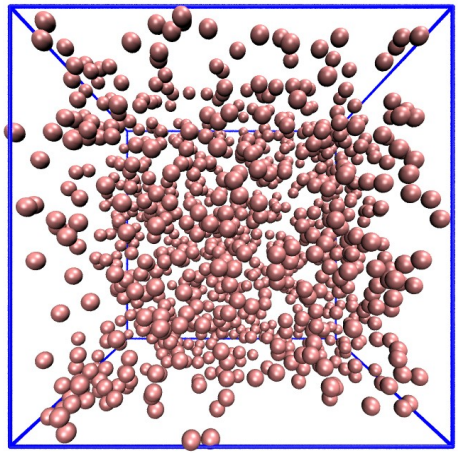
Overview of this session

- Excercise problem ‘solution’
- Running LAMMPS on ARCHER
- Domain decomposition
  - balance command
  - comm_style command
- Compiling LAMMPS on ARCHER
  - Using make & cmake
  - A quick word on GPU compilation
- Short break
- Altering LAMMPS source code
- Any questions

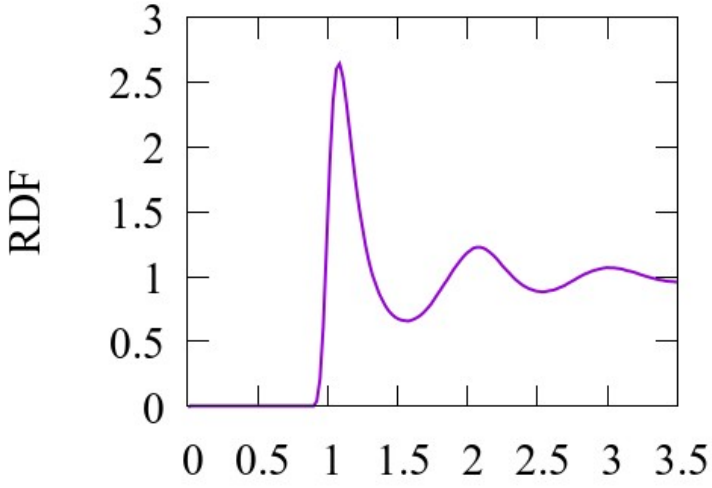
1) Exercise ‘solutions’

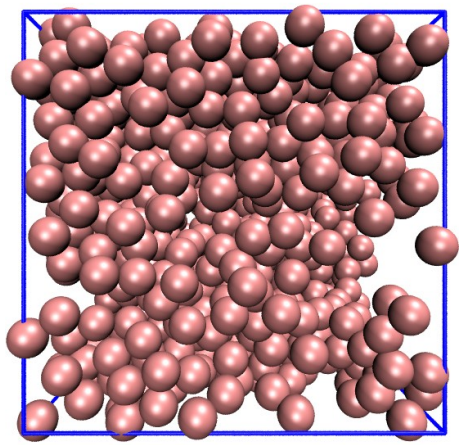
ρ = 0.05



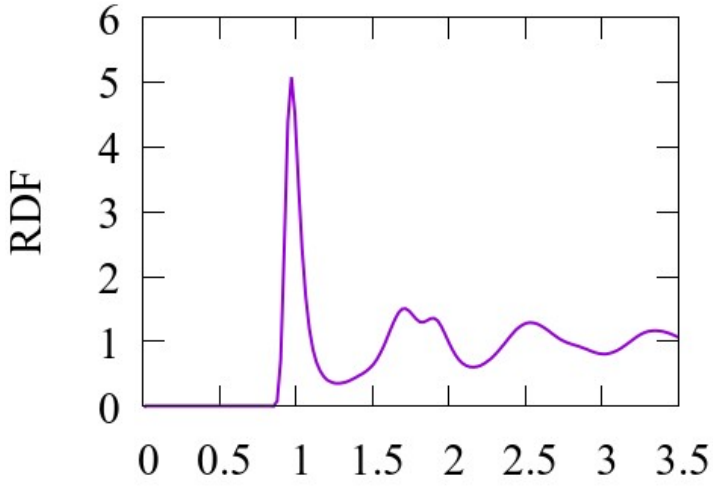


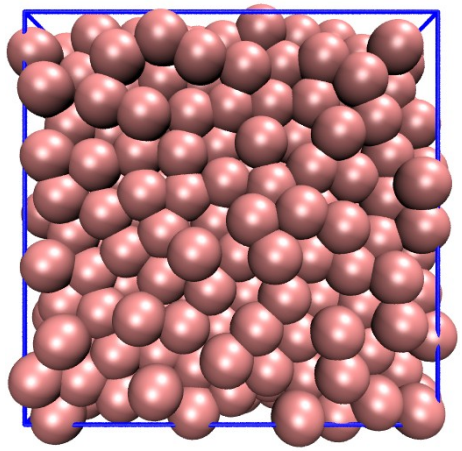
ρ = 0.8





ρ = 1.4





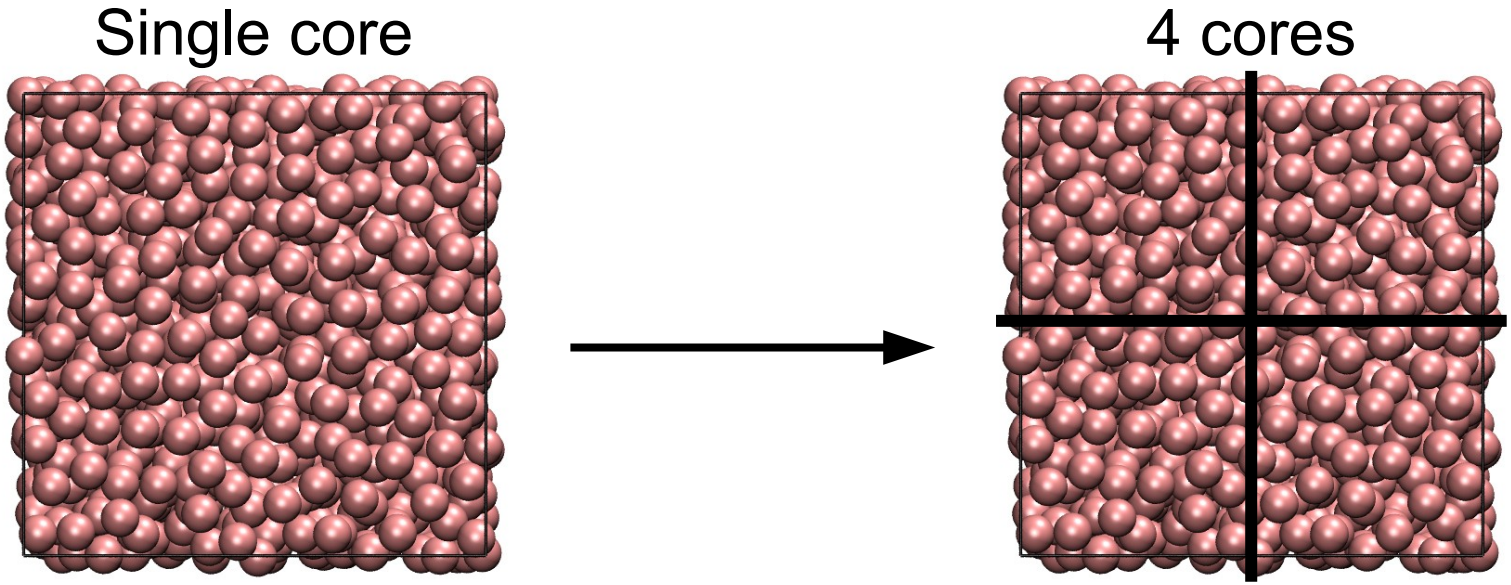
epcc

THE UNIVERSITY OF EDINBURGH

2) LAMMPS on ARCHER

- ARCHER uses modular packages
  - ARCHER has pre-installed versions of LAMMPS as a module
- Multi-processor jobs on ARCHER should:
  - Have the required modules loaded
  - Be submitted to backend nodes *via* submission script
  - Have a number of 24-processor nodes assigned
  - Be assigned a run time

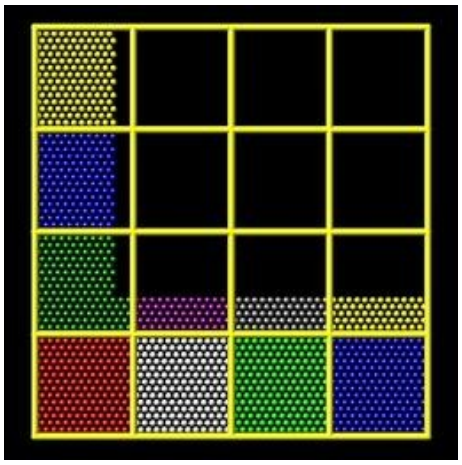
3) Domain decomposition in MD



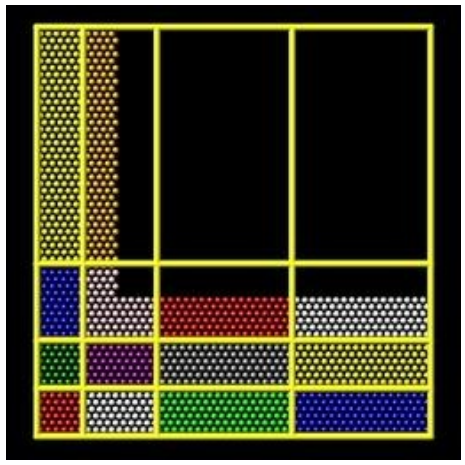
- Easiest to divide box spatially
  - ARCHER nodes have 24 cores = 4x3x2 split
- Every timestep, core communicates with neighbour when updating positions/energies/forces/velocities
- Communication can be expensive!

3a) ‘balance’ command

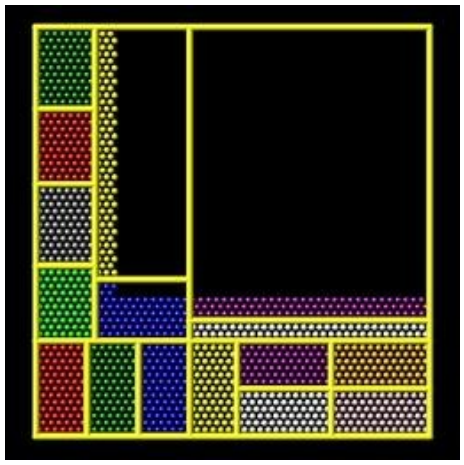
No balance command



‘Shift’ balance ‘grid-type’ command



‘RCB’ balance ‘tiling-type command



- Two methods of balancing
  - ‘balance’ command – sets domain when invoked only
  - ‘fix balance’ command – periodically re-updates domain sizes
- Two types of domains:
  - Grid/block style (default) – domain decomposed into regular 3D blocks
  - Tiling style – domain decomposed into non-overlapping blocks of various shapes/sizes
- For tiling, need to set comm_style appropriately!

4) Compiling LAMMPS on ARCHER

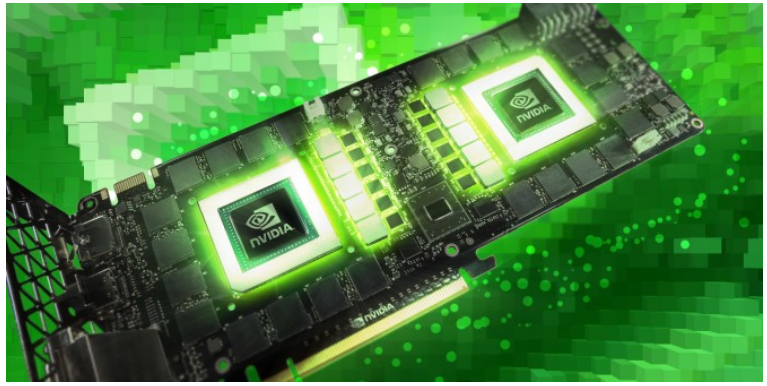
- Necessary modules:
  - module switch PrgEnv-cray PrgEnv-intel
  - module switch intel intel/17.0.3.191
  - module switch gcc gcc/6.3.0
  - module load fftw
- git clone https://github.com/lammps/lammps
- Using ‘make’:
  - Copy Cray Makefile (in /work/y14/shared/LAMMPS_WORKSHOP/day_2_material) to lammps/src/MAKE/MACHINE
  - In lammps/src directory, make xc30
- Using ‘cmake’ (differences from using ‘make’):
  - module load cmake
  - mkdir lammps/build; cd lammps/build
  - cmake -DCMAKE_CXX_COMPILER=CC -DBUILD_MPI=ON ../cmake/
  - make

4a) Compiling packages

- List of packages:
  https://lammps.sandia.gov/doc/Packages.html
- List of ‘difficult’ packages:
  https://lammps.sandia.gov/doc/Build_package.html
- Adding packages using ‘make’:
  - Install with – ‘make yes-PACKAGE_NAME’
  - Uninstall with – ‘make no-PACKAGE_NAME’
- Adding packages using ‘cmake’:
  - Add option ‘-D PKG_NAME=yes’ in cmake command

4b) Compiling LAMMPS for GPUs

- GPU package:
  - Allows for pair styles and long-ranged electrostatic calculations to be done on GPUs
  - Needs NVIDIA GPU and NVIDIA CUDA drivers
  - Need to compile LAMMPS GPU library in lammps/lib/
    - In Makefile, change CUDA_ARCH
    - Make file using:
      make -j N -f Makefile.linux.single
      where N is the machine core-number



GPU	sm_$	CUDA
c2070	sm_20	max 8.0
GTX 470	sm_20	max 8.0
GT 610	sm_21	max 8.0
GTX 770	sm_30	min 8.0
GTX 1050Ti	sm_61	min 8.0
GTX 1080Ti	sm_61	min 8.0
Titan Xp	sm_61	min 8.0
Titan V	sm_70	min 9.0

5) Altering LAMMPS source code

- Required changes in .h file:
  - ‘Style’ command
  - LMP_NAME_OF_COMMAND
  - Class name (must match style name)
- In .cpp file:
  - All instances of ‘style’ class
  - .h file called
  - Any calculations that you want added/removed

5a) Useful links for developers

- Modify & extend LAMMPS:
  https://lammps.sandia.gov/doc/Modify.html
- LAMMPS Developer Manual:
  https://lammps.sandia.gov/doc/Developer.pdf

6) ‘Certificate of Attendance’ exercise

- Modify in.lj_certification
  - This is exactly the same as in.lj from Day 1
  - Can find file on course website or on ARCHER in /work/y14/shared/LAMMPS_WORKSHOP/day_2_material
- Currently, the code runs with fixed particle number, volume, and temperature
  - Exercise is to change this to run with fixed particle number, **pressure**, and temperature
- Please email log.lammps files to training@epcc.ed.ac.uk

Thanks for your time. Any questions?

Thanks also to:

- Clair Barrass for running the show behind the scenes
- Rui Apostolo for his input on post-processing software and tips on compiling LAMMPS on GPUs
- Ranga Radhakrishnan and Yang Cui for their inputs on post-processing software

EPSRC
Pioneering research and skills
epcc

CRAY
THE SUPERCOMPUTER COMPANY

NERC SCIENCE OF THE ENVIRONMENT

archer

THE UNIVERSITY OF EDINBURGH