arnes
povezujemo znanje

EURO
SLING

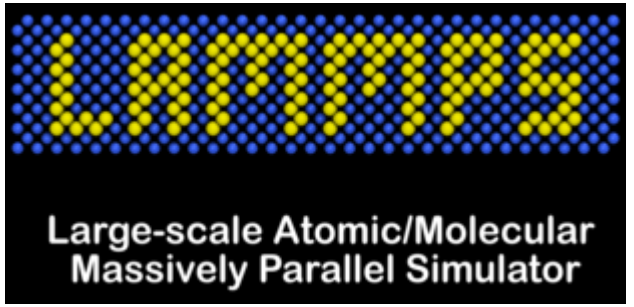
MREŽA ZNANJA
Ljubljana, 3.–5. december 2024

LAMMPS simulation package

Tilen Potisk
National Institute of Chemistry
Center of Excellence MultiXscale

LAMMPS – classical MD simulation engine

- Why choose LAMMPS?
- High-level overview of LAMMPS
- Neighborlists and Parallelization
- Getting started with LAMMPS
- OBMD in LAMMPS



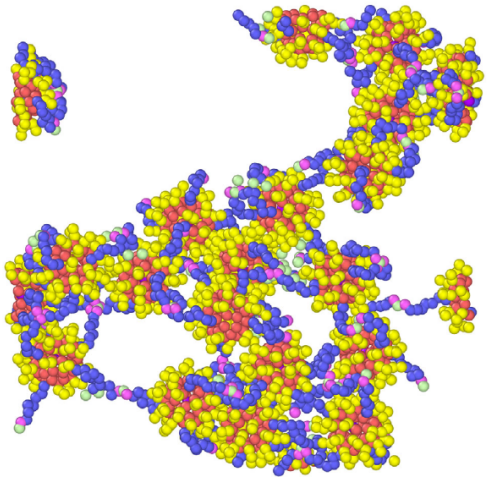
developed at:

Sandia National Laboratories

Why choose LAMMPS?

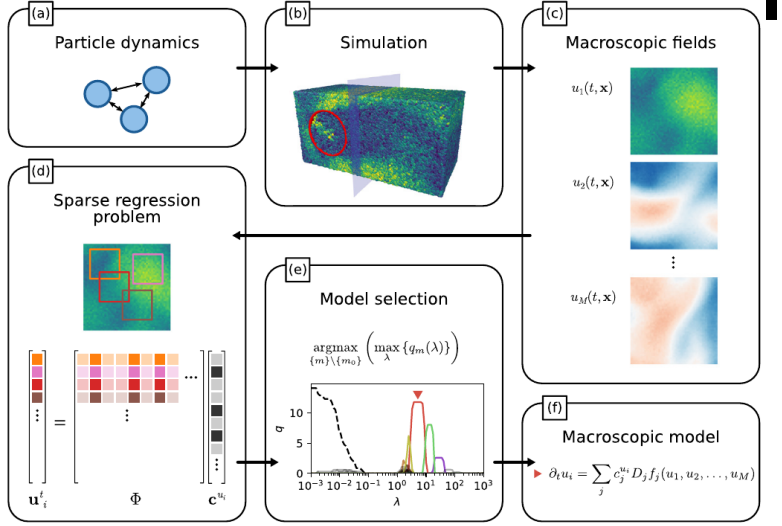
- Extremely versatile: **MD, DPD, SPH, LBM, ...**
- Excellent **scallability**
- Endless functionalities
- Easy to modify and extend
- Large community (matsci.org forum)
- Well documented https://docs.lammps.org/Manual.html

Biomedicine



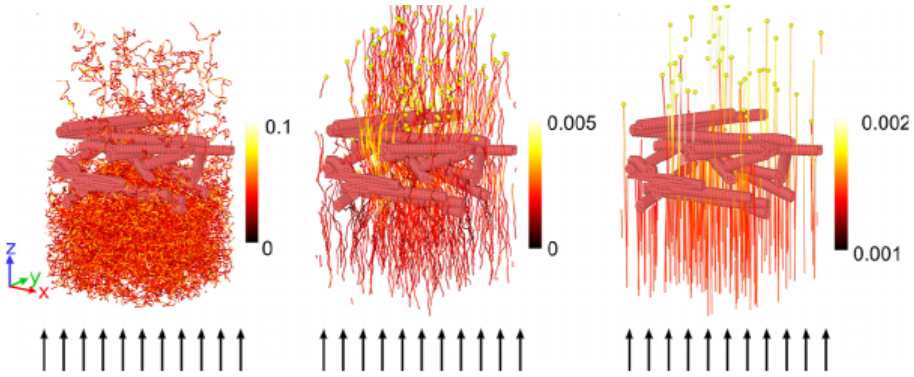
Potisk, et al. Adv. Theory Simul 6. 2023

Discovering models



Jug, et al. Comput. Methods Appl. Mech. Engrg. 2024

Engineering



Potisk et al. ACS Appl. Nano Mater. 2023

High-level overview of LAMMPS

- **Script**-based system for specifying simulation set-ups
- Key components of input scripts:
  - **Force fields**: several 100
  - **Fixes**:
    - Modify the behavior of the simulation
    - Define integrators (NVE, NVT, NPT, NPH, **μVT**)
    - Define walls
    - Average quantities
    - ...
  - **Computes**: Calculates properties (on-the-fly)
  - **Dumps**: Output of simulation data / computed data

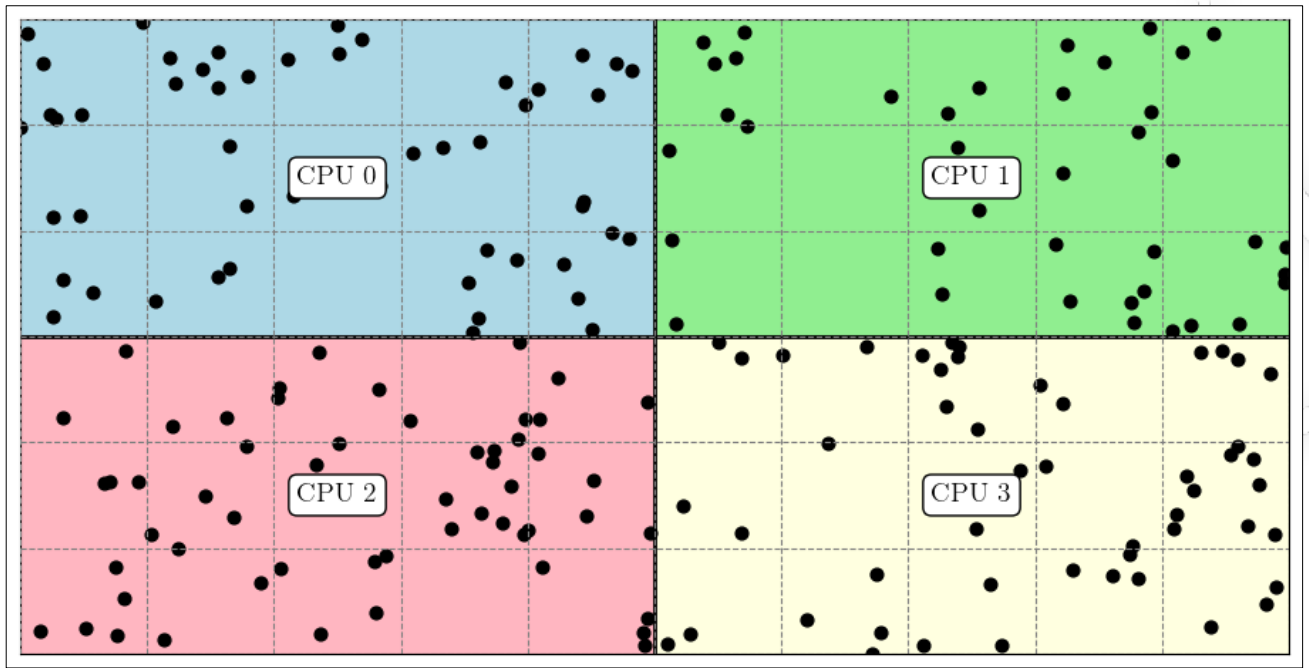
```
1   units               lj
2
3   atom_style          atomic
4
5   boundary            p p p
6
7   read_data           in.data
8
9   comm_modify         vel yes
10  pair_style          dpd 1.0 1.0 12345
11  pair_coeff          * * 100.0 10.0
12
13  neighbor            0.1 bin
14
15  timestep            0.01
16
17  fix                 1 all nve
18
19  thermo              100
20  thermo_style        custom step temp press spcpu
21
22  dump                1 all custom 100 positions.lammpstrj id type x y z vx vy vz
23
24  run                 1000
```

Neighborlists 1/2

- Solving a dynamic system for N particles:

$$m_i \ddot{\mathbf{r}}_i = \sum_{j \neq i} \mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D + \mathbf{F}_{ij}^R$$

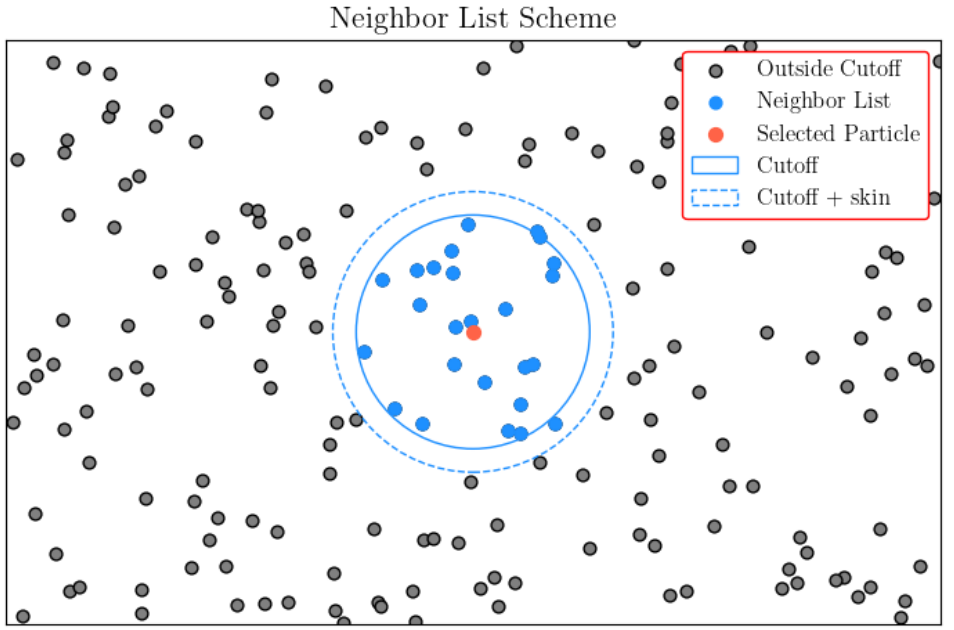
- Naive way: calculate forces between all pairs of particles
- **Spatial decomposition** method crucial for efficient MD engine
- Each processor takes care of a region of the simulation box
- During the simulation timestep the processors communicate with each other using e.g. MPI or OpenMP framework



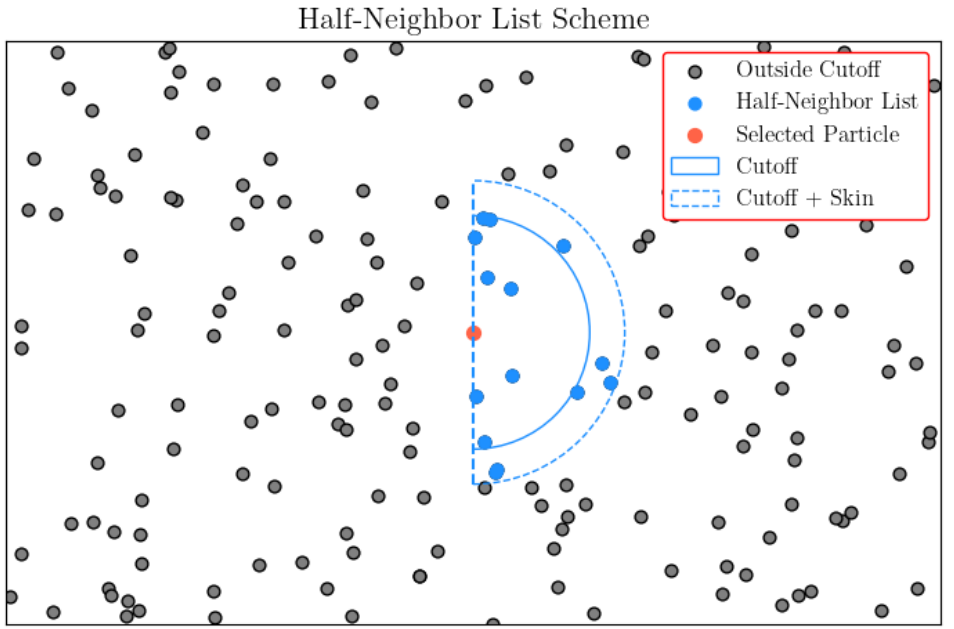
Multiscale

Neighborlists 2/2

- All particles within a cutoff + skin
- Full neighborlist



- Default in LAMMPS:
- Half-neighborlists



Multiscale

Getting started with LAMMPS 1/3

- Define units and atom_style

```
1   units               lj
2
3   atom_style          atomic
```

- Define boundaries (p – periodic, f – open, s – shrink wrapped)

```
5   boundary            p p p
```

- Set or read initial configuration

```
7   read_data           in.data
```

- Set force-fields

```
9   comm_modify         vel yes
10  pair_style          dpd 1.0 1.0 12345
11  pair_coeff          * * 100.0 10.0
```

- Define neighborlist – details (skin and build type)

```
13  neighbor            0.1 bin
```

- Define timestep

```
15  timestep            0.01
```

Getting started with LAMMPS 2/3

- Define fix-es

```
17  fix                 1 all nve
```

- Output to screen or logfile

```
19  thermo              100
20  thermo_style        custom step temp press spcpu
```

- Dump trajectories

```
22  dump                1 all custom 100 positions.lammpstrj id type x y z vx vy vz
```

- Run the simulation

```
24  run                 1000
```

```
Per MPI rank memory allocation (min/avg/max) = 4.565 | 4.565 | 4.565 Mbytes
   Step          Temp          v_press        Atoms         v_center          S/CPU
         0   0             57.678624           9000   5424              0
       100   1.484784      60.158753           7903   5656              830.79269
       200   1.1065102     55.542654           7633   5460              823.77649
       300   1.0200879     51.255367           7615   5253              942.2622
       400   1.0202678     52.617374           7441   5321              925.58211
       500   1.0233872     55.384671           7430   5456              919.45496
       600   1.0112488     59.582081           7430   5653              885.99751
       700   1.0053704     61.256254           7430   5725              837.87076
       800   1.0072056     57.958903           7430   5591              864.5623
       900   0.9995747     54.634099           7430   5408              884.05734
      1000   1.017831      52.705618           7430   5326              939.06104
      1100   1.0121781     53.503393           7425   5377              950.25134
      1200   1.0010637     57.149316           7425   5546              930.7289
      1300   0.99962791    60.014635           7425   5674              874.11364
      1400   1.0110957     59.861568           7425   5667              854.49099
      1500   1.0105783     56.575944           7425   5520              876.20316
```

Multiscale

Getting started with LAMMPS 3/3

- Taking into account the default settings:

```
1   read_data           in.data
2   comm_modify         vel yes
3   pair_style          dpd 1.0 1.0 12345
4   pair_coeff          * * 100.0 10.0
5   fix                 1 all nve
6   dump                1 all custom 100 positions.lammpstrj id type x y z vx vy vz
7   run                 1000
```

- Command-line options:
  - Basic

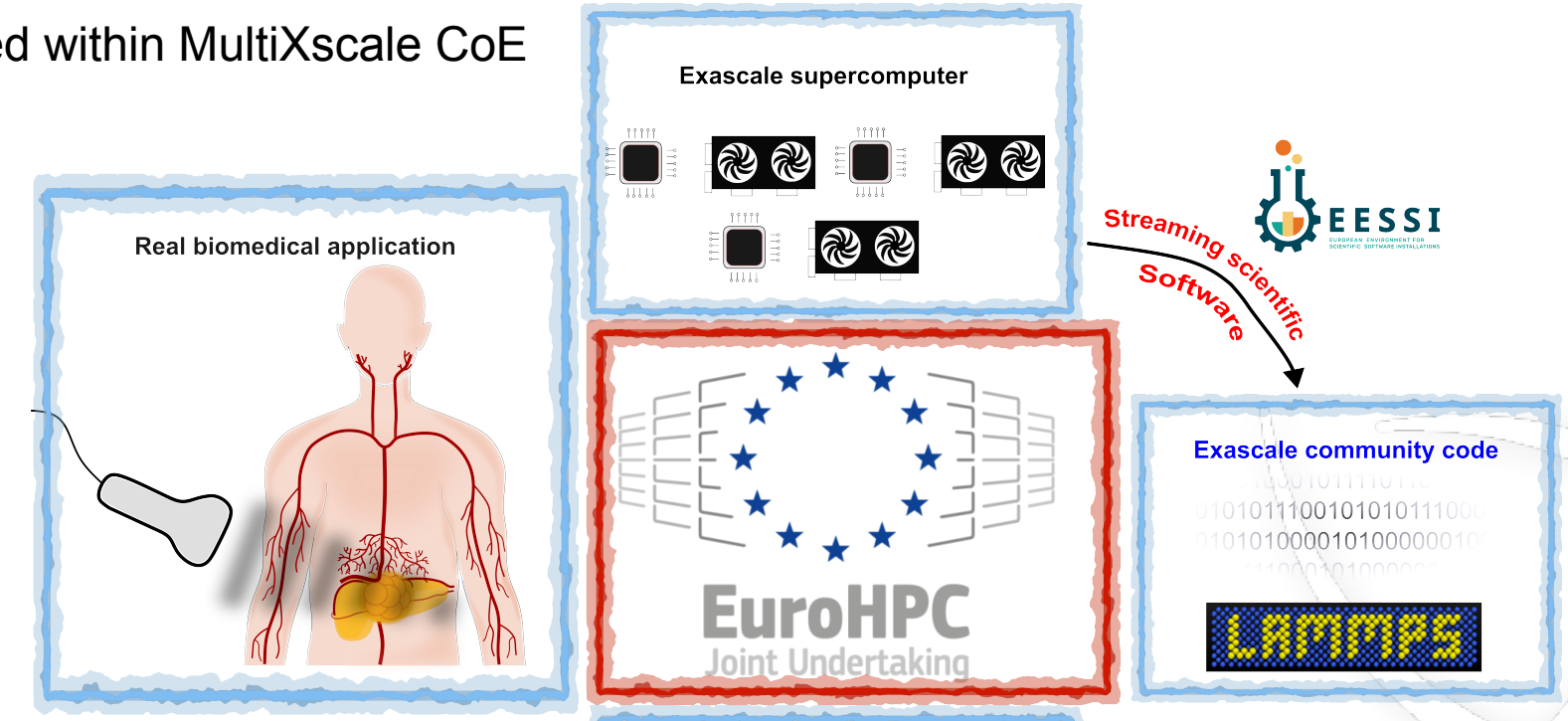
```
tilen@tilen:~$ lmp -in in.lammps
```

  - Utilizing accelerated features (e.g. KOKKOS)

```
tilen@tilen:~$ lmp -in in.lammps -k on -sf kk -pk kokkos newton on
```

Open Boundary Molecular Dynamics in LAMMPS

- Developed within MultiXscale CoE



MultiXscale

Open Boundary Molecular Dynamics in LAMMPS

- OBMD implemented as a fix style,
- combining existing fix-es:
  - **fix deposit** (for inserting particles)
  - **fix evaporate** (for particle deletion)
  - **fix addforce** (imposing fluxes)

- Boundary: use f – type in the open direction

```
boundary            f p p
```

- Define buffer regions

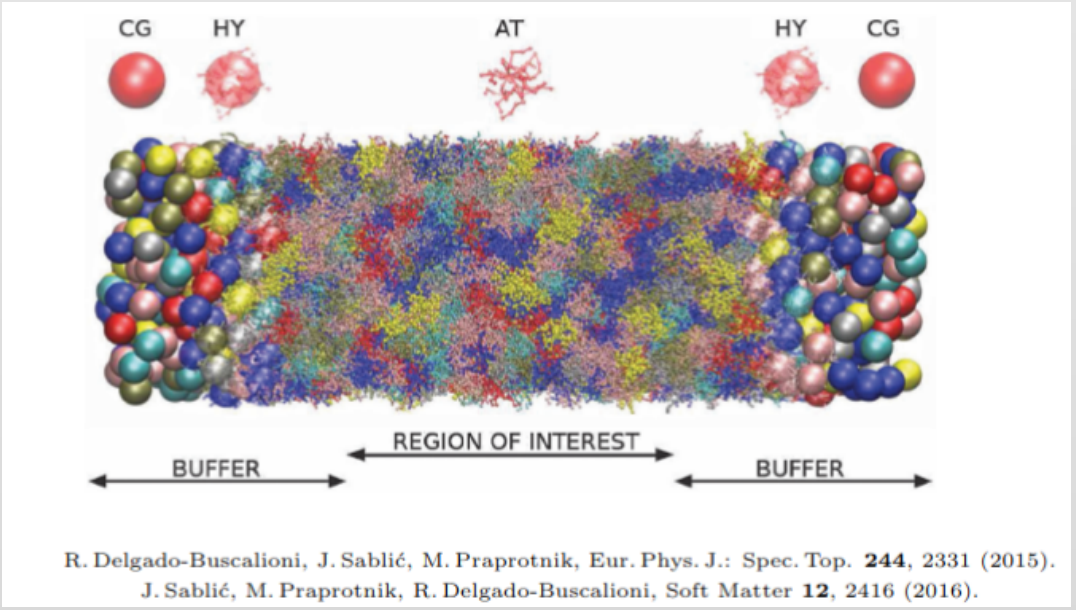
```
region              leftB block 0 buffer_size 0 Ly 0 Lz
```

- Particle exchange is controlled by deletions and insertions at each step

$$\Delta N = \frac{\Delta t}{\tau}(N_0 - N)$$

- Fluxes are conserved by adding forces to the buffer particles:

$$J A \mathbf{n} = \sum_i \mathbf{f}_i + \sum_{i'} \frac{\Delta(m_{i'} \mathbf{v}_{i'})}{\Delta t}$$



R. Delgado-Buscalioni, J. Sablić, M. Praprotnik, Eur. Phys. J.: Spec. Top. **244**, 2331 (2015).
J. Sablić, M. Praprotnik, R. Delgado-Buscalioni, Soft Matter **12**, 2416 (2016).

Open Boundary Molecular Dynamics in LAMMPS

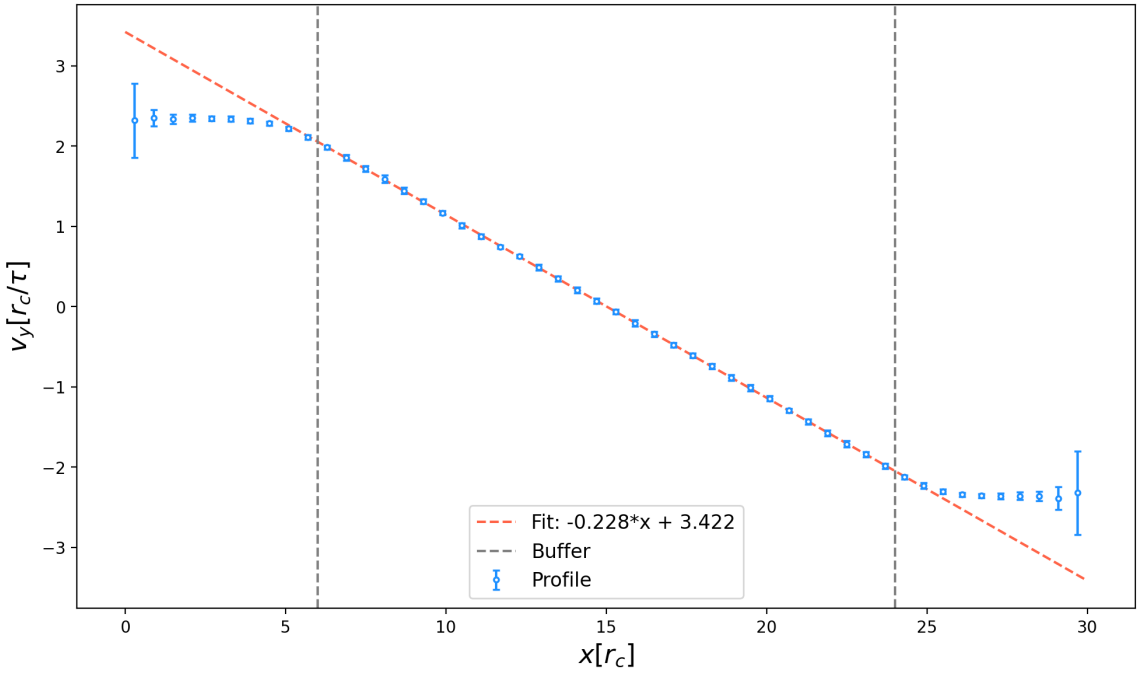
- Define fix obmd to perform insertion, deletions and impose fluxes

```
fix                 ID group-ID obmd insert-type nevery seed Pxx Pxy Pxz dPxx Tleft Tright lambda &
                    region1 ID1 region2 ID2 region3 ID3 region4 ID4 region5 ID5 region6 ID6 &
                    buffersize buffer_size shearsize shear_size alpha alpha tau tau_usher &
                    nbuf nbuf step stepflag attempt nattempts &
                    usher usherflag Et ds0 dtheta0 uovlp dsovlp eps0 nattempt
```

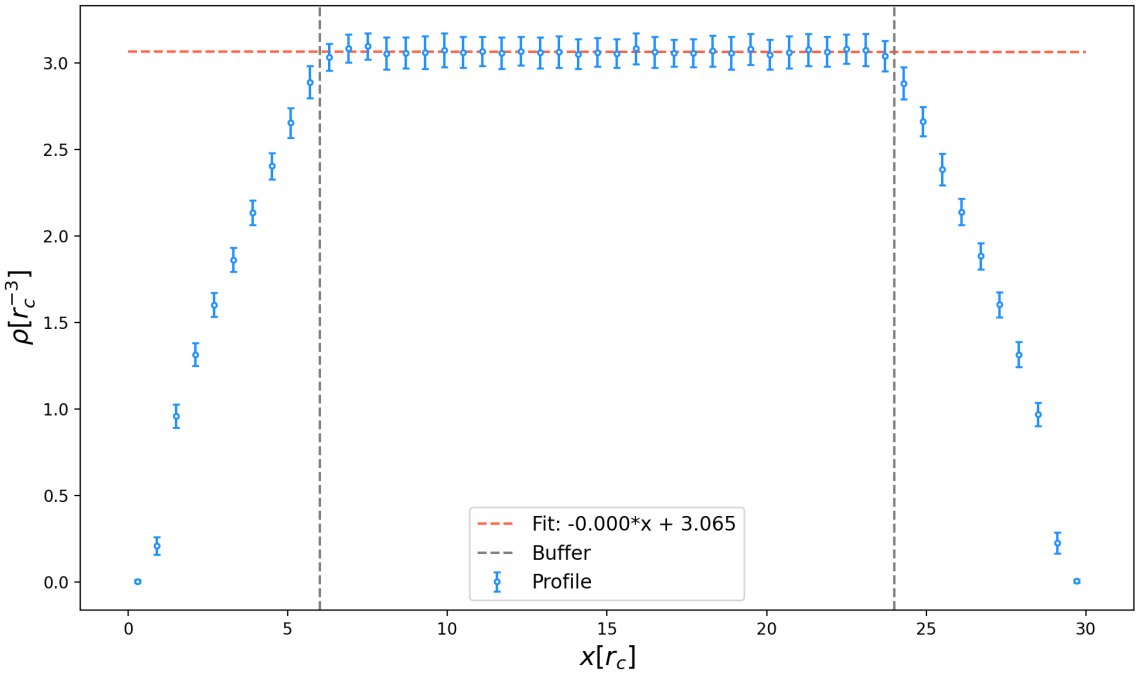
Open Boundary Molecular Dynamics in LAMMPS

- OBMD – also a framework for nonequilibrium MD simulations
- E.g.: imposing a shear stress Pxy on a fluid

Velocity profile

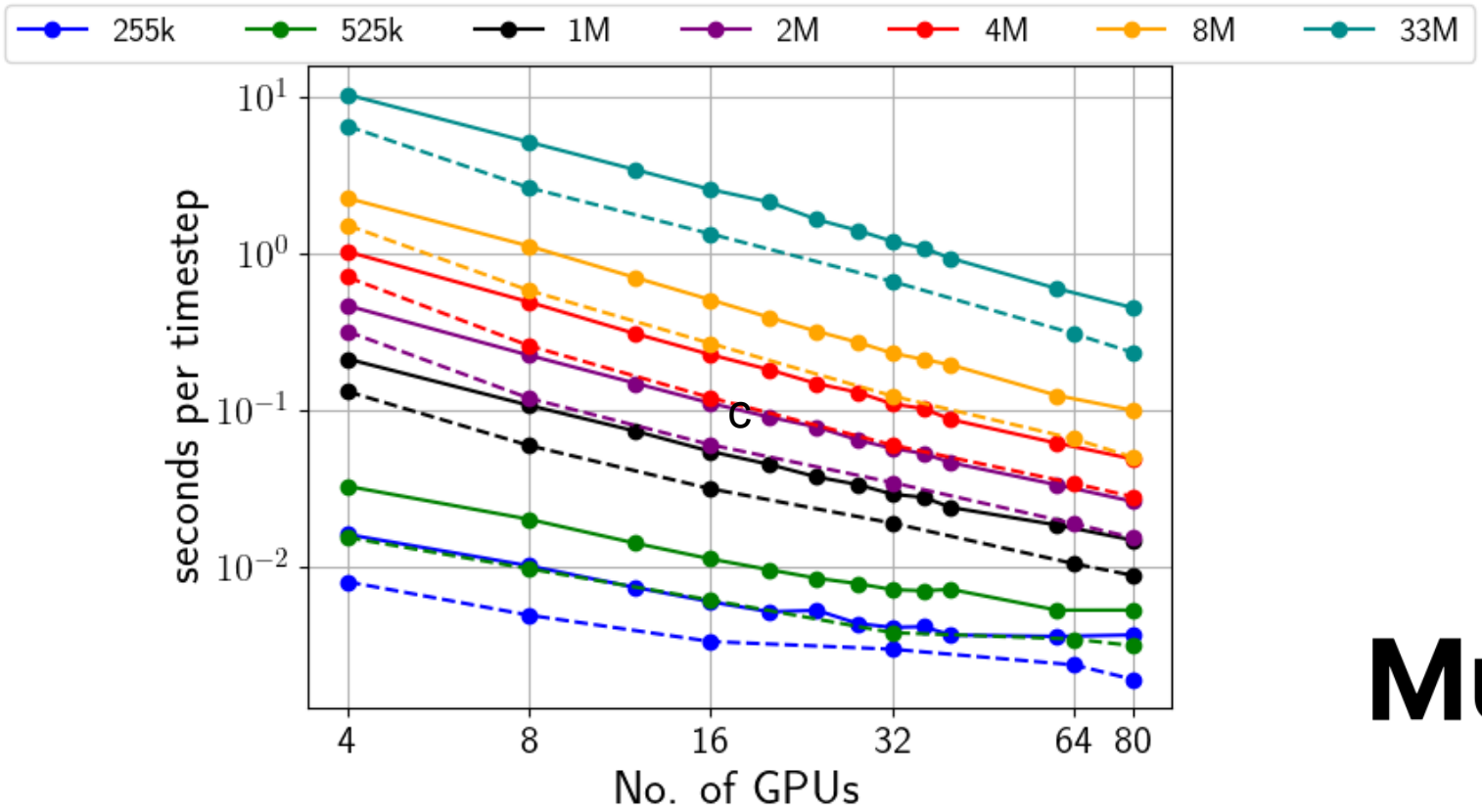


Density profile



Open Boundary Molecular Dynamics in LAMMPS

- Strong scaling analysis of LAMMPS with OBMD on EuroHPC Vega
- From 0.25 to 33 M particles.
- Up to 80 GPUs (A100) = 1/3 of EuroHPC Vega GPU partition



MultiXscale

Co-funded by the European Union

EuroHPC Joint Undertaking

Web page: multixscale.eu
Facebook: MultiXscale
X: @MultiXscale
LinkedIn: MultiXscale
Youtube channel: MultiXscale

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and countries participating in the project under grant agreement No 101093169.

Financiranje

Projekt EuroCC 2 financira Evropska unija. Financiran je s sredstvi Skupnega podjetja za evropsko visokozmogljivo računalništvo (EuroHPC JU) ter Nemčije, Bolgarije, Avstrije, Hrvaške, Cipra, Češke republike, Danske, Estonije, Finske, Grčije, Madžarske, Irske, Italije, Litve, Latvije, Poljske, Portugalske, Romunije, Slovenije, Španije, Švedske, Francije, Nizozemske, Belgije, Luksemburga, Slovaške, Norveške, Turčije, Republike Severne Makedonije, Islandije, Črne gore in Srbije v okviru sporazuma o dodelitvi sredstev št. 101101903.

Delovanje Nacionalnega kompetenčnega centra SLING sofinancira Ministrstvo za visoko šolstvo, znanost in inovacije.

Medijski sponzor

Računalniške novice
www.racunalniske-novice.com

Co-funded by the European Union

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA VISOKO ŠOLSTVO, ZNANOST IN INOVACIJE