Running GROMACS on CPU and GPU

Alessandra Villa
PDC-Center for High Performance Computing,
KTH-Royal Institute of Technology,
Stockholm, Sweden

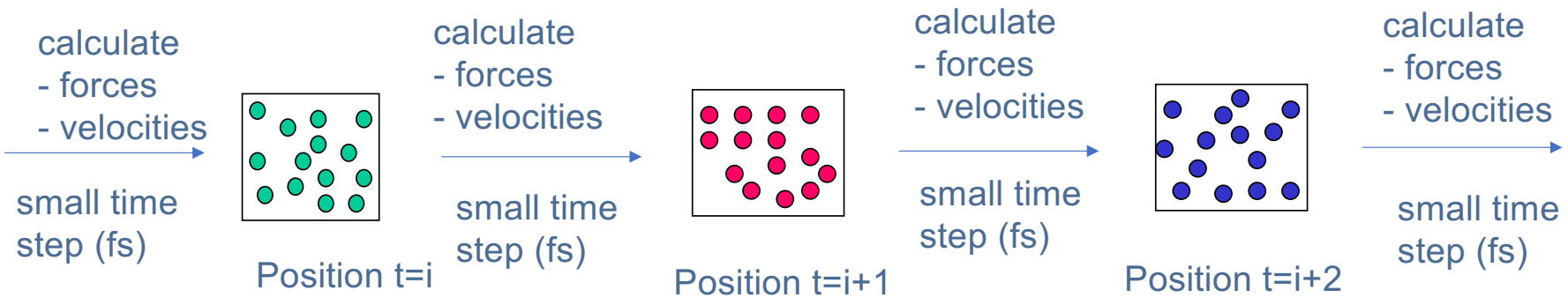
avilla@kth.se

Why good performance

GROMACS

One goal of a molecular dynamics simulation is to generate enough representative conformations of the molecular system in such a way that accurate values of a property can be obtained.

How? by iteratively solving equations of motion



High computational cost

GROMACS

Some interactions are more costly than others

- Non-bonded interactions
  - Calculated over every pair of atoms in the system
  - ~ to N^2 where N is the number of atoms in the system
  - More than **90% of the computing time**

=> cut-off, PME (mdp parameters)

GROMACS

How to get good performance

- Optimal mdp parametes
    - Currently most mdp parameters do not affect performance much (except PME order and grid).
    - Automated PME tuning optimises the Coulomb cut-off and PME grid size (in GROMACS)
- Choose good options for mapping tasks in mdrun to available hardware

=> effect on performance but not easy

Some terms

core: A hardware compute unit that executes instructions. More than one core in a processor.

node: A node is the name usually used for one unit in a computer cluster (shared access to the same memory without requiring any network hardware)

thread: A stream of instructions for a core to execute (a sequence of instructions made available over time)

GROMACS

CPU vs GPU

central processing unit vs Graphics processing unit

- CPU
  - Optimized for serial tasks
  - Optimized for latency
  - Typically requires lower amount of parallelism:
    - fewer faster cores, fewer threads
  - Large complex cache hierarchy

- GPU
  - Optimized for highly parallel tasks
  - Optimized for throughput
  - Requires a lot of parallelism:
    - Large number of threads active
  - High memory bandwidth

Latency -> time to finish a fixed task

Throughput -> number of tasks in fixed time

GROMACS

Hardware role: Devana Cluster

- 2 x Intel Xeon Gold 6338 processors each with 32 CPU cores and 100 Gbit/s HDR Infiniband interconnect. => we can use max 64 cores per node for any CPU runs

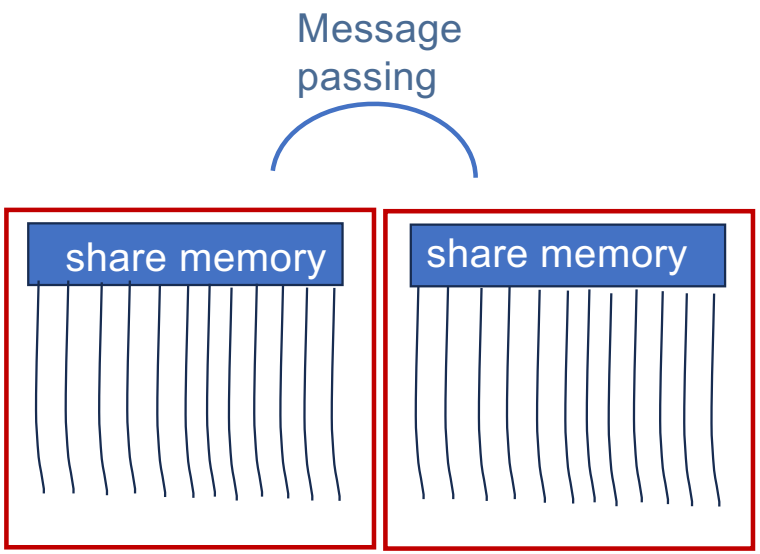
- four NVIDIA Volta™ A100 accelerators for each GPU nodes

Parallelization:

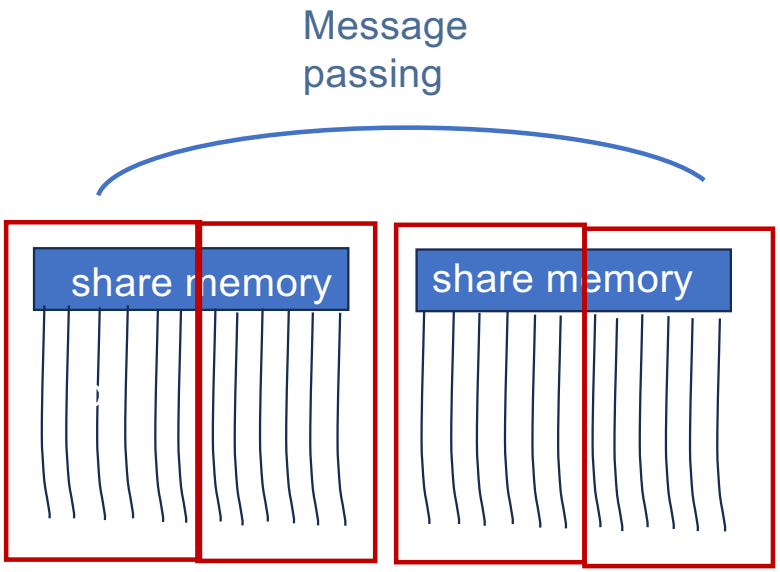
Thread parallelization:
Use multiple threads to execute code in parallel on the same and/or different cores. All threads in a process have access to all data in the process, but it still takes time to move that data from one core to the another (OpenMP for thread parallelisation)

Message passing:
Process running on cores in the same or different nodes exchange data by passing messages (MPI or thread-MPI GROMACS)

MPI rank/ OMP threads



2 MPI ranks – 12 **thread** per rank



4 MPI ranks – 6 **thread** per rank

You will see mofd in the hands-on

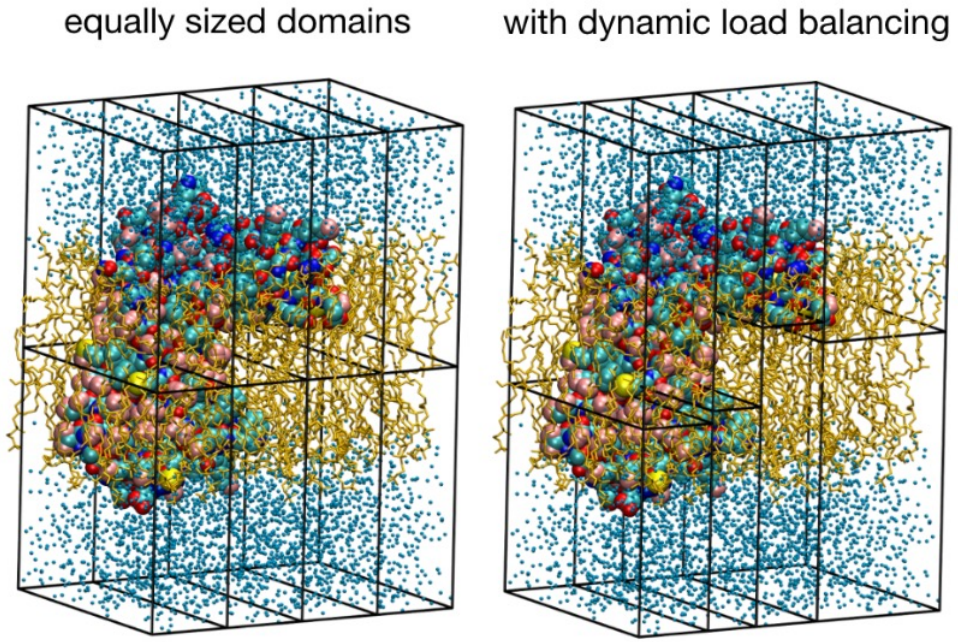
NOTE

In MPI**, a rank** is the smallest grouping of hardware used in the multi-node parallelization scheme.

Algorithms: domain decomposition

GROMACS

The **domain decomposition** (DD) algorithm decomposes the (short-ranged) component of the non-bonded interactions into domains that share spatial locality.
Each domain handles all the particle-particle interactions for its members, and is mapped to a single MPI rank



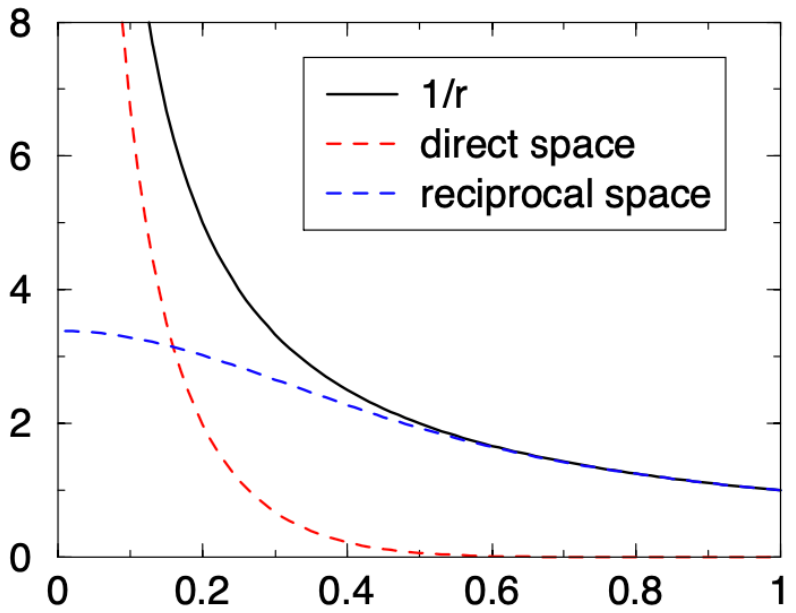
Algorithms – PP/PME domains

Long-range electrostatics

- 1/r is long range: can not use a plain cut-off
- Particle mesh Ewald:
  - Decompose 1/r into short+long-range:

$$V_{\text{Coulomb}} = \underbrace{V_{direct}}_{\text{pair term}} + \underbrace{V_{reciprocal}}_{\text{long-range all vs all}} + \underbrace{V_0}_{\text{constant}}$$

  - The reciprocal part is computed on a grid using a 3D-FFT



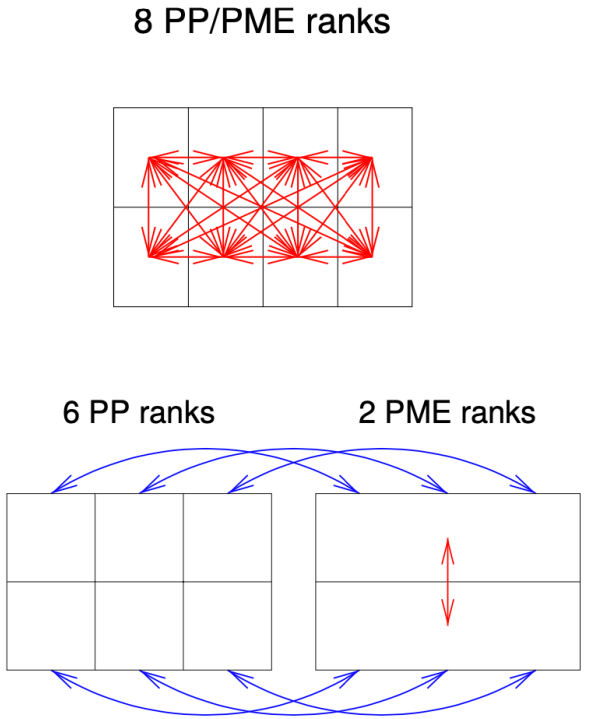
Algorithms – PP/PME domains

GROMACS

3D FFT requires global communication

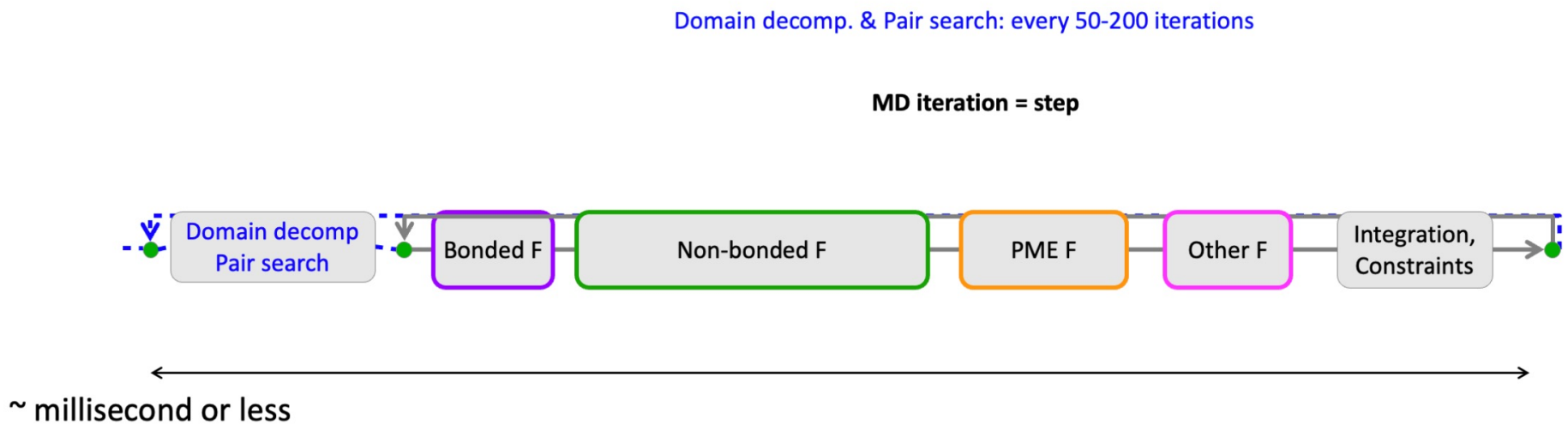
=> parallel efficiency gets worse as more ranks participate.

If there are separate PME ranks, then the remaining ranks handle particle-particle (PP) work.

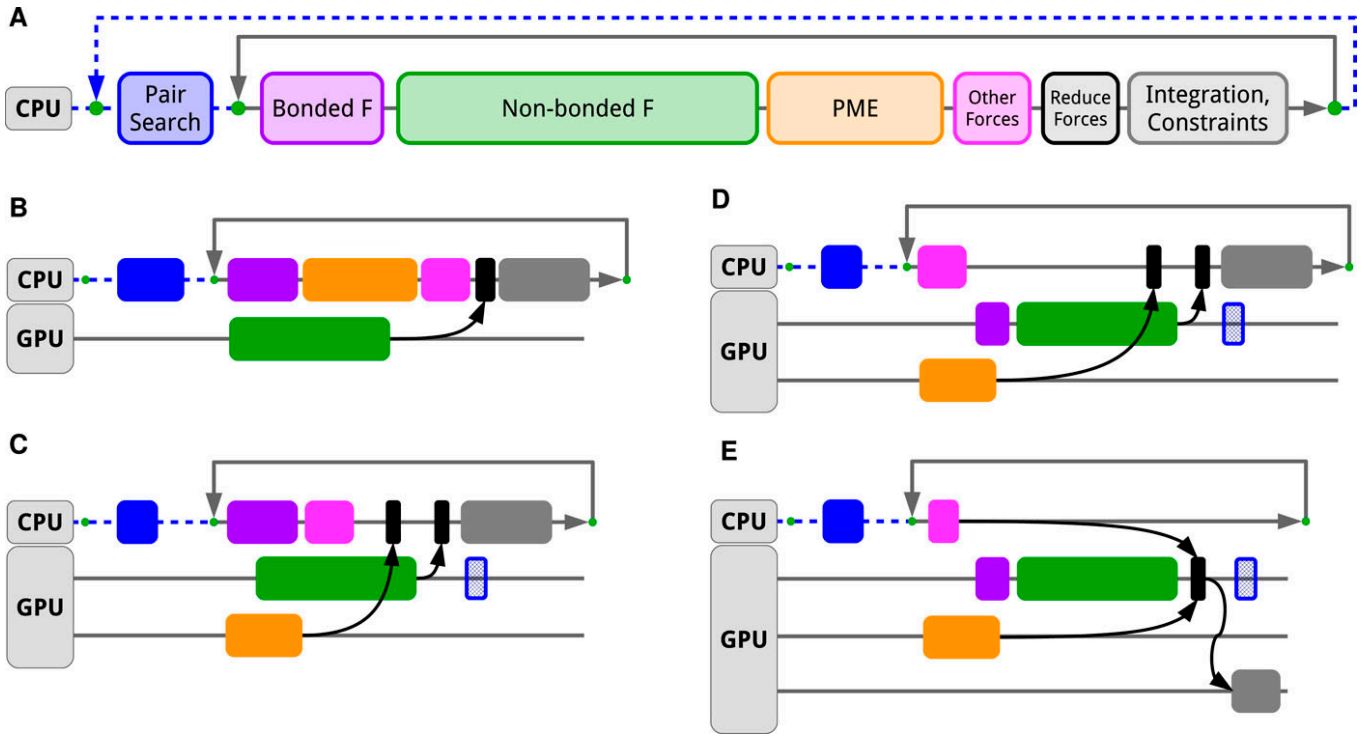


GROMACS

One MD step on single CPU



Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS

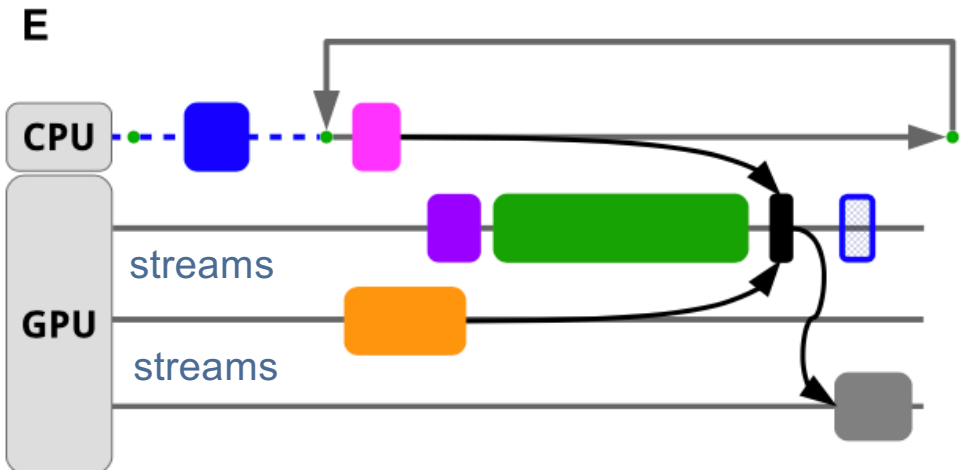


J. Chem. Phys.. 2020;153(13). doi:10.1063/5.0018516

What on CPU and GPU?

GROMACS

CPU is used for scheduling work, transferring data, and launching computation on the accelerator, as well as inter- and intra-node communication.
Accelerator tasks are launched asynchronously using APIs to allow concurrent CPU–GPU execution

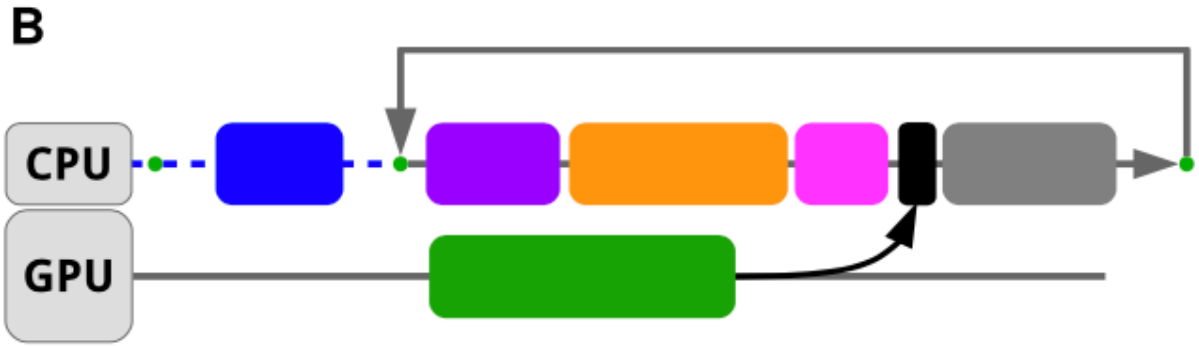


Non bonded forces
PME forces
Bonded forces

dynamic list pruning
integration, constrains

What on CPU and GPU
non-bonded forces off-loaded

GROMACS

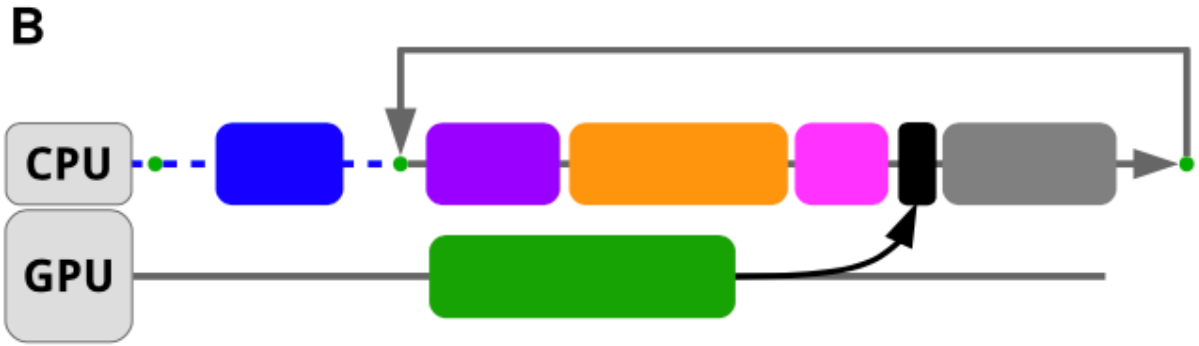


Non bonded forces
PME forces
Bonded forces

dynamic list pruning
integration, constrains

What on CPU and GPU
non-bonded forces off-loaded

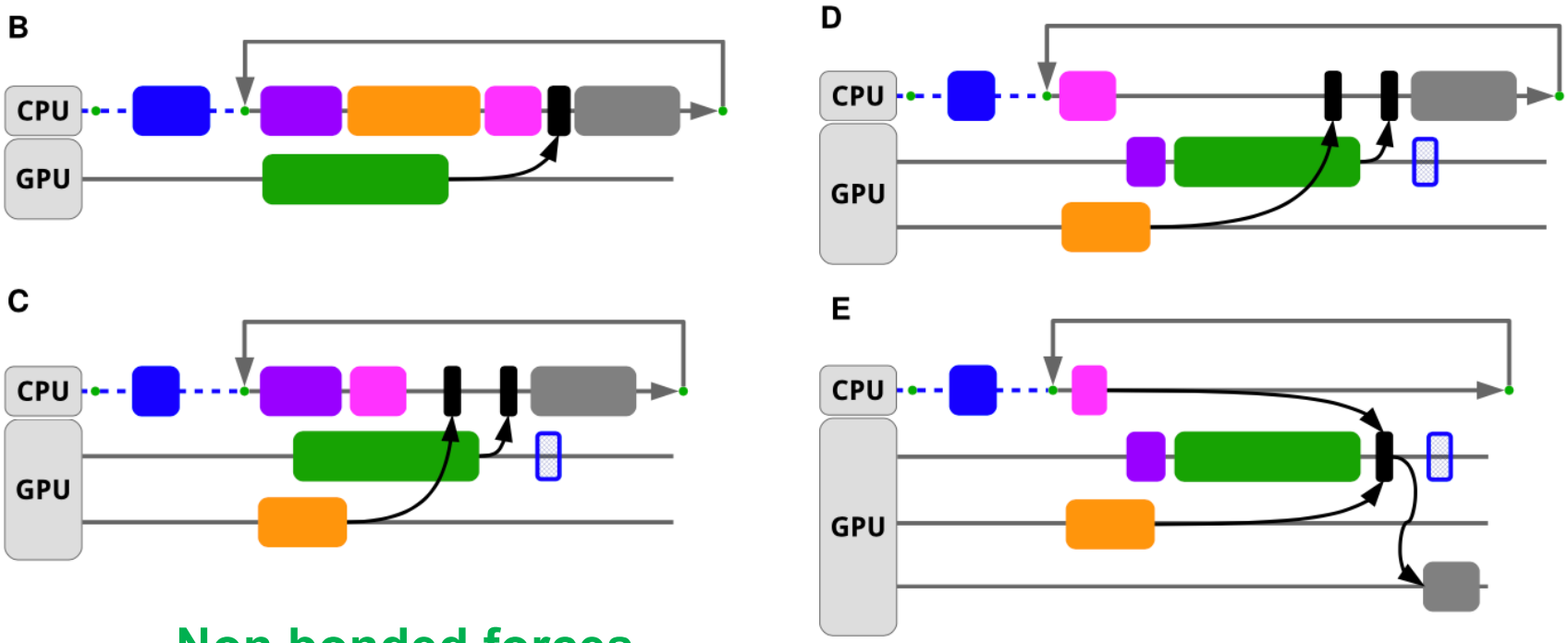
GROMACS



The gradual shift in CPU–GPU performance balance in heterogeneous systems brought the need for offloading further force tasks to avoid the CPU becoming a bottleneck or, from a cost perspective, not needing expensive CPUs.

What on CPU and GPU

GROMACS



Non bonded forces

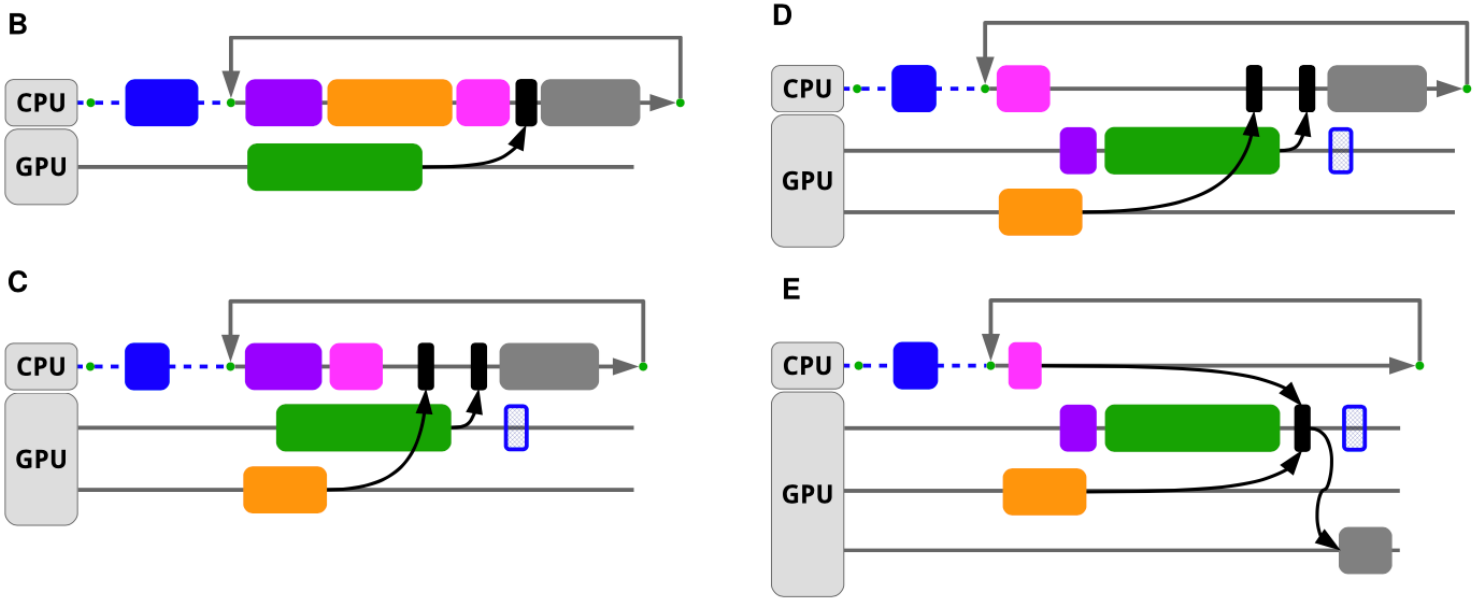
PME forces

Bonded forces

dynamic list pruning

integration, constrains

What on CPU and GPU

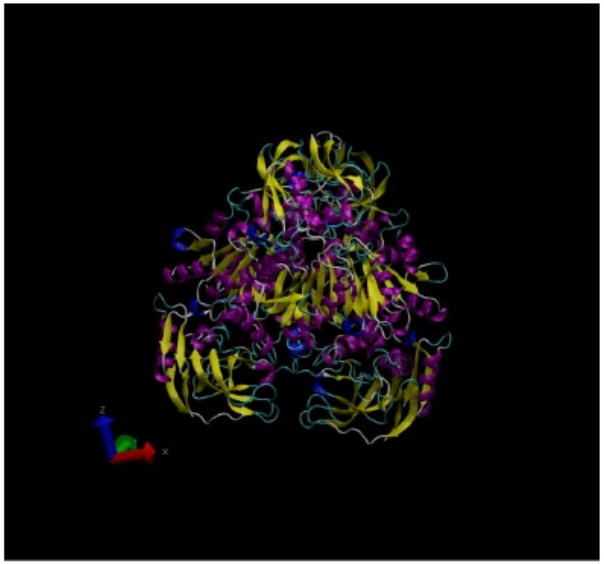


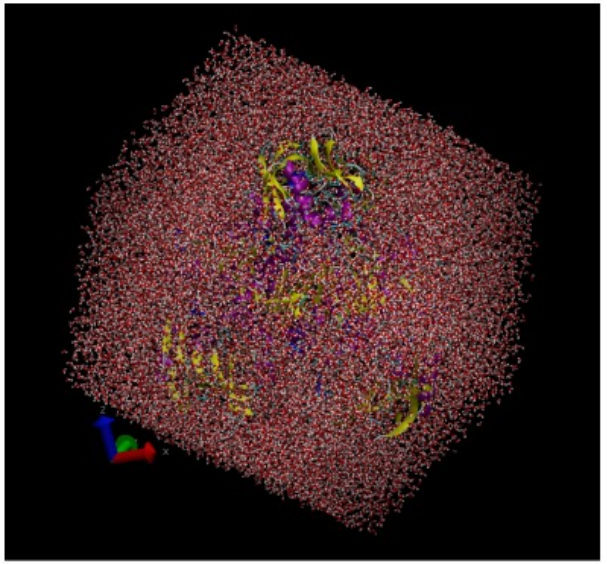
More during hands-on

Tutorial system

GROMACS

- NADP-dependent alcohol dehydrogenase
- protein: 4 chains, 21332 atoms
- Rhombic dodecahedron unit cell
- water: 24379 molecules, 74127 atoms
- 12 Na^+ ions

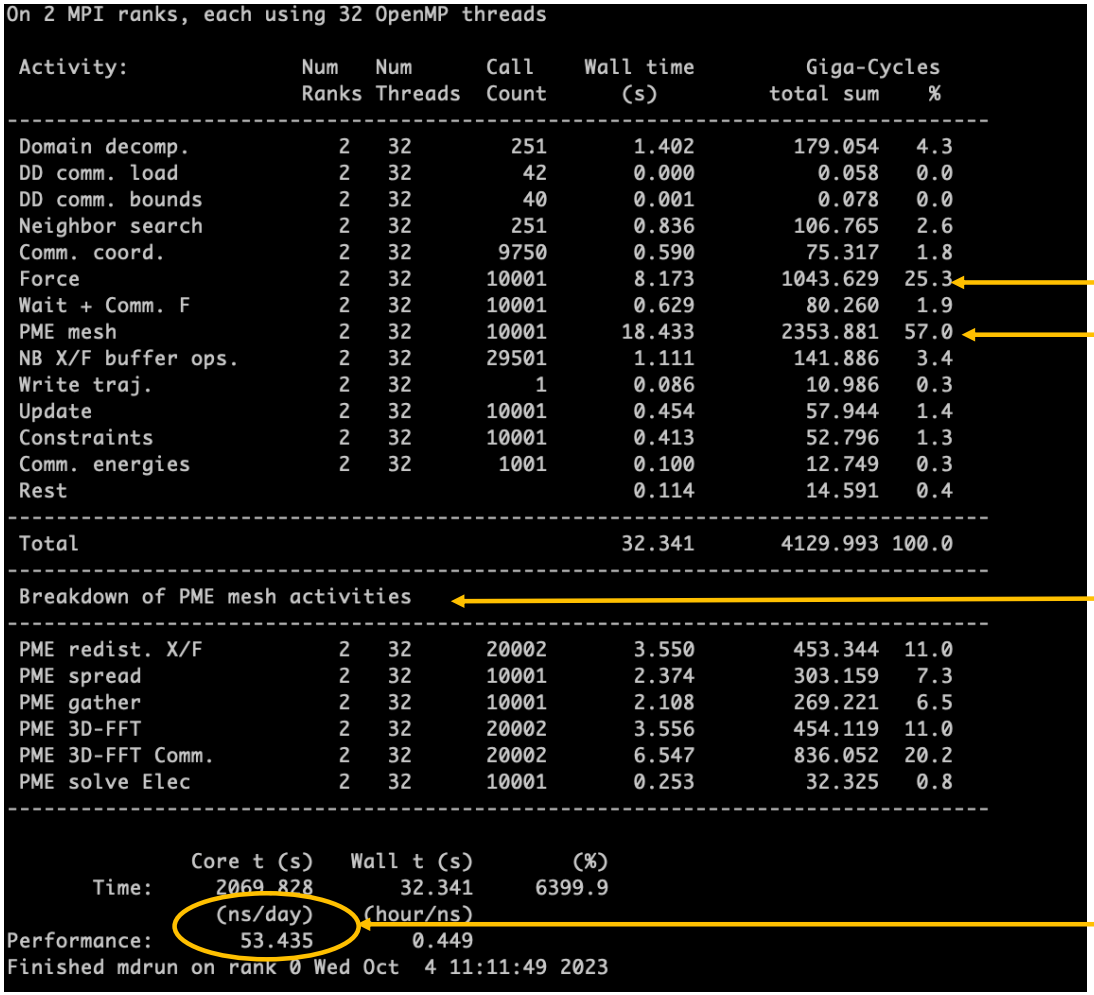




gmx mdrun option

-g LOGFILE set a custom name for the log file (default md.log);
-pin on enable mdrun internal thread affinity setting (might override externally set affinities) (auto default)
-tunepme/-notunepme enable PME task load balancing (yes)
-nsteps N set the number of simulations steps for the current run to N (N=-1 means infinitely long runs, intended to be combined with -maxh);
-maxh H stop the simulation after 0.99*H hours;
-resethway reset performance counters halfway through the run;
-nb/-pme/-bonded/-update task assignment options used to select tasks to run on either CPU, GPU.

Example of md log



```
On 2 MPI ranks, each using 32 OpenMP threads

 Activity:              Num   Num      Call    Wall time        Giga-Cycles
                        Ranks Threads  Count      (s)         total sum    %
-----------------------------------------------------------------------------
 Domain decomp.           2   32         251       1.402        179.054   4.3
 DD comm. load            2   32          42       0.000          0.058   0.0
 DD comm. bounds          2   32          40       0.001          0.078   0.0
 Neighbor search          2   32         251       0.836        106.765   2.6
 Comm. coord.             2   32        9750       0.590         75.317   1.8
 Force                    2   32       10001       8.173       1043.629  25.3
 Wait + Comm. F           2   32       10001       0.629         80.260   1.9
 PME mesh                 2   32       10001      18.433       2353.881  57.0
 NB X/F buffer ops.       2   32       29501       1.111        141.886   3.4
 Write traj.              2   32           1       0.086         10.986   0.3
 Update                   2   32       10001       0.454         57.944   1.4
 Constraints              2   32       10001       0.413         52.796   1.3
 Comm. energies           2   32        1001       0.100         12.749   0.3
 Rest                                           0.114         14.591   0.4
-----------------------------------------------------------------------------
 Total                                         32.341       4129.993 100.0
-----------------------------------------------------------------------------
 Breakdown of PME mesh activities
-----------------------------------------------------------------------------
 PME redist. X/F          2   32       20002       3.550        453.344  11.0
 PME spread               2   32       10001       2.374        303.159   7.3
 PME gather               2   32       10001       2.108        269.221   6.5
 PME 3D-FFT               2   32       20002       3.556        454.119  11.0
 PME 3D-FFT Comm.         2   32       20002       6.547        836.052  20.2
 PME solve Elec           2   32       10001       0.253         32.325   0.8
-----------------------------------------------------------------------------

               Core t (s)   Wall t (s)        (%)
       Time:     2069.828       32.341     6399.9
                 (ns/day)   (hour/ns)
Performance:       53.435        0.449
Finished mdrun on rank 0 Wed Oct  4 11:11:49 2023
```

Part taking most computational time

Subdivision of PME mesh computation

Absolute performance per day

References

- GROMACS Manual https://manual.gromacs.org/documentation/current/user-guide/mdrun-performance.html
- Webinar:Improvements in the GROMACS heterogeneous parallelization by Szilárd Páll
- Short talk by Berk Hess Getting good performance in GROMACS default
- Mapping computation to HPC hardware & GPU accelerators and heterogeneous architectures by Szilárd Páll and Berk Hess https://doi.org/10.6084/m9.figshare.22303477
- Páll, et al. (2020) J. Chem. Phys. 153, 134110 (doi:10.1063/5.0018516)

Thank you and see at hands-on