GTC 20130319

A microsecond a day keeps the doctor away: Efficient GPU Molecular Dynamics with GROMACS

Erik Lindahl

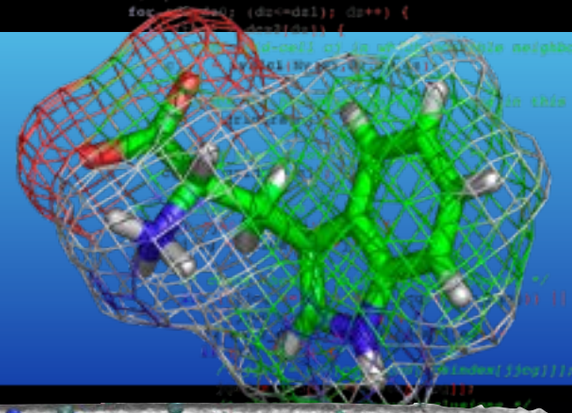
erik.lindahl@scilifelab.se

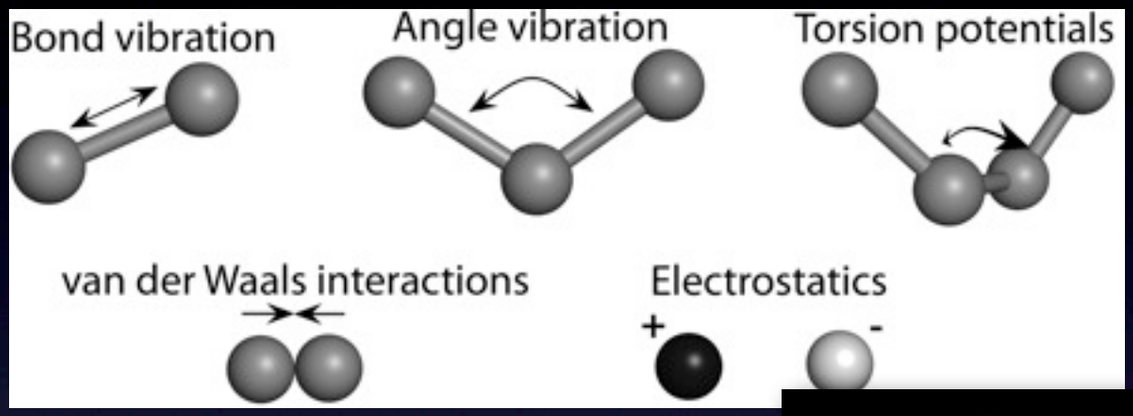
SciLifeLab STOCKHOLM

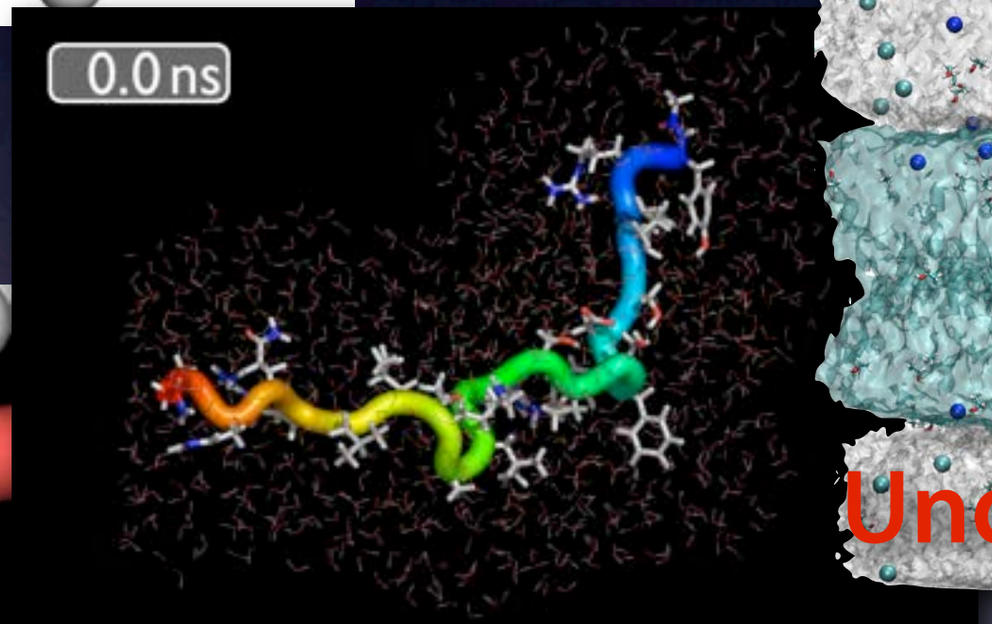
UNIVERSITET STOCKHOLMS

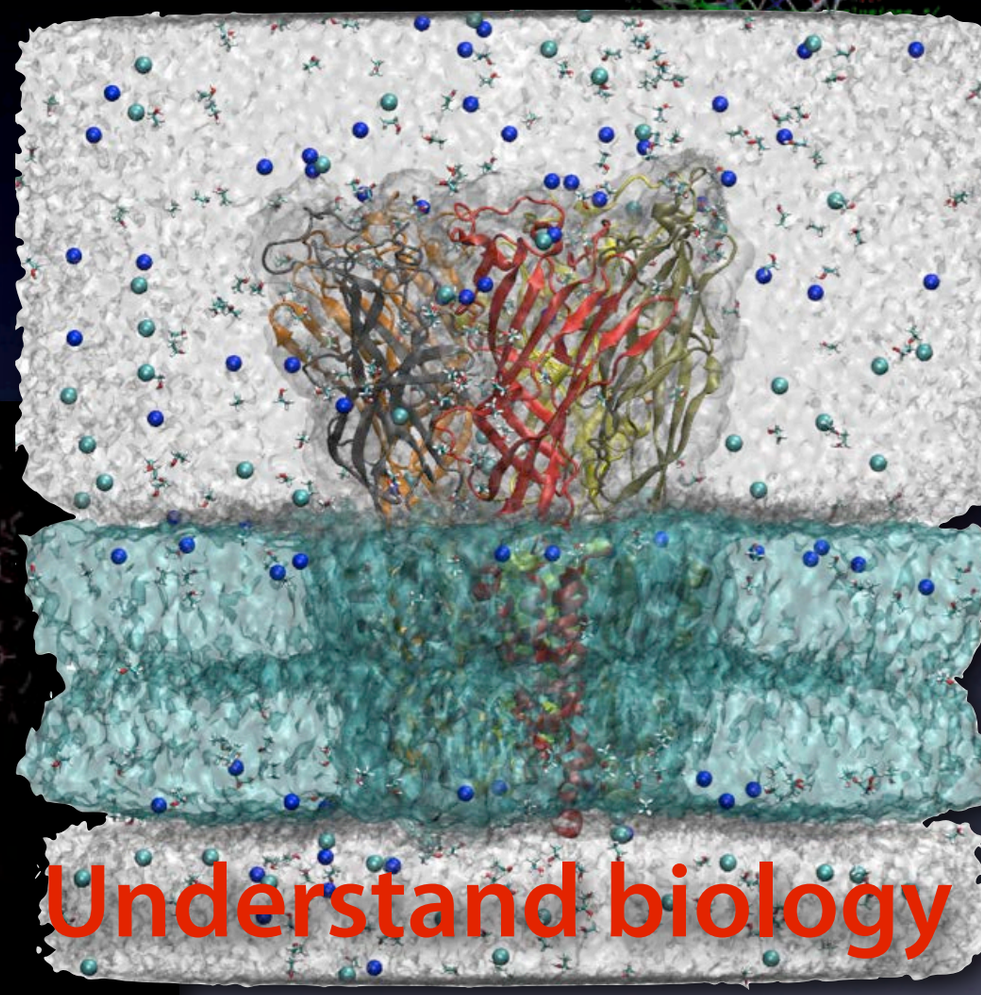
KTH VETENSKAP OCH KONST

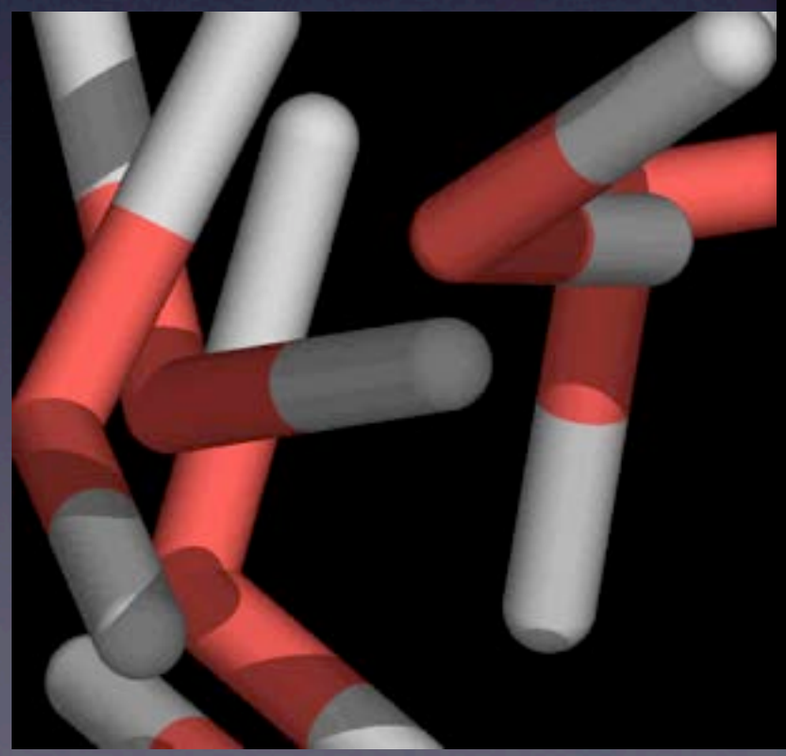
Molecular Dynamics

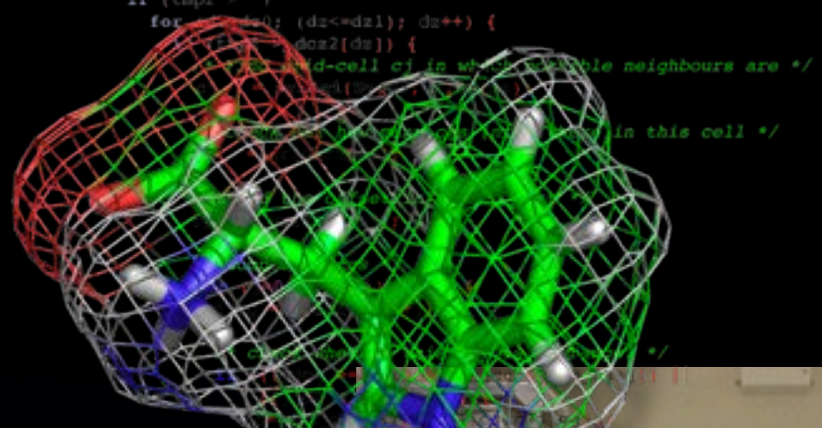








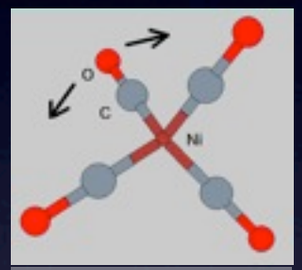


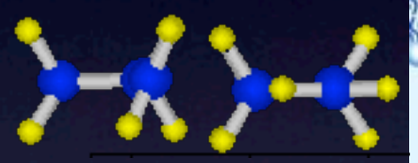


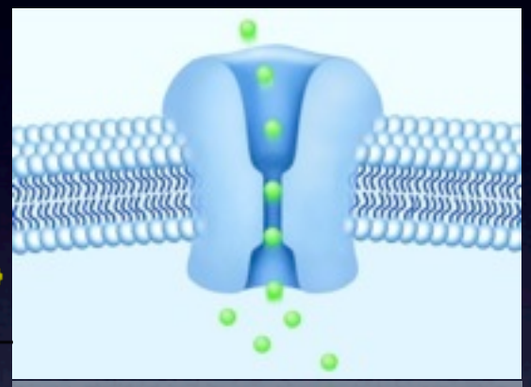
We're comfortably on the single-µs scale today

Larger machines often mean larger systems, not necessarily longer simulations

MD Simulation Challenges







Experiments

Efficient averaging

Less detail

Physics | Chemistry | Biology

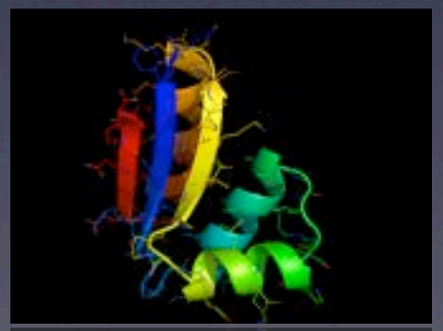
10^{-15}s 10^{-12}s 10^{-9}s 10^{-6}s 10^{-3}s 10^{0}s 10^{3}s

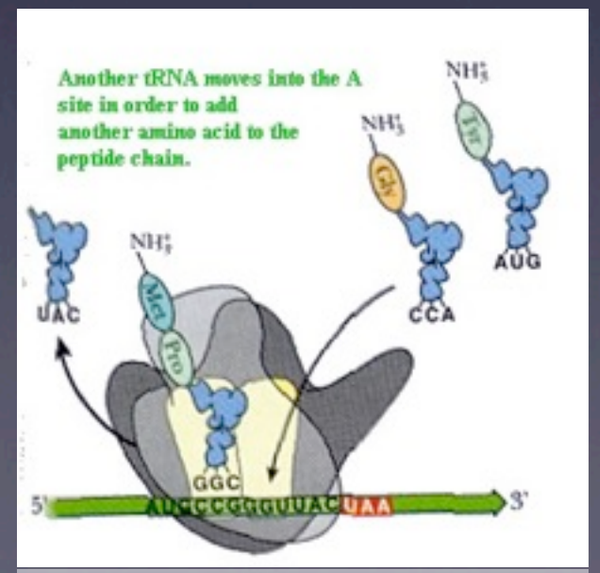
Simulations

Extreme detail

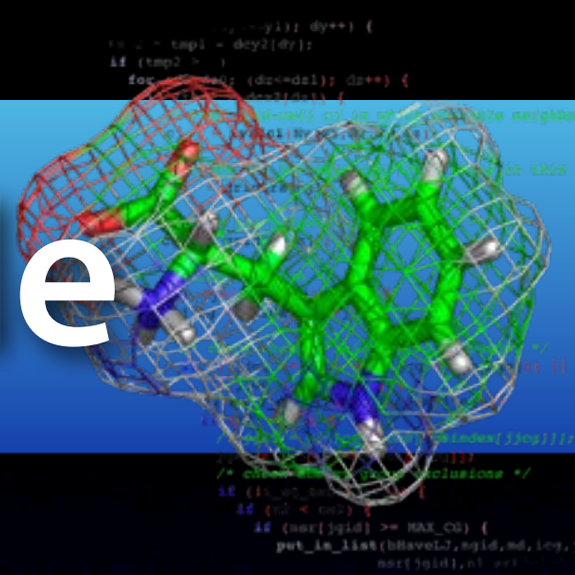
Sampling issues?

Parameter quality?





The iteration cycle



With a time step of 5fs...

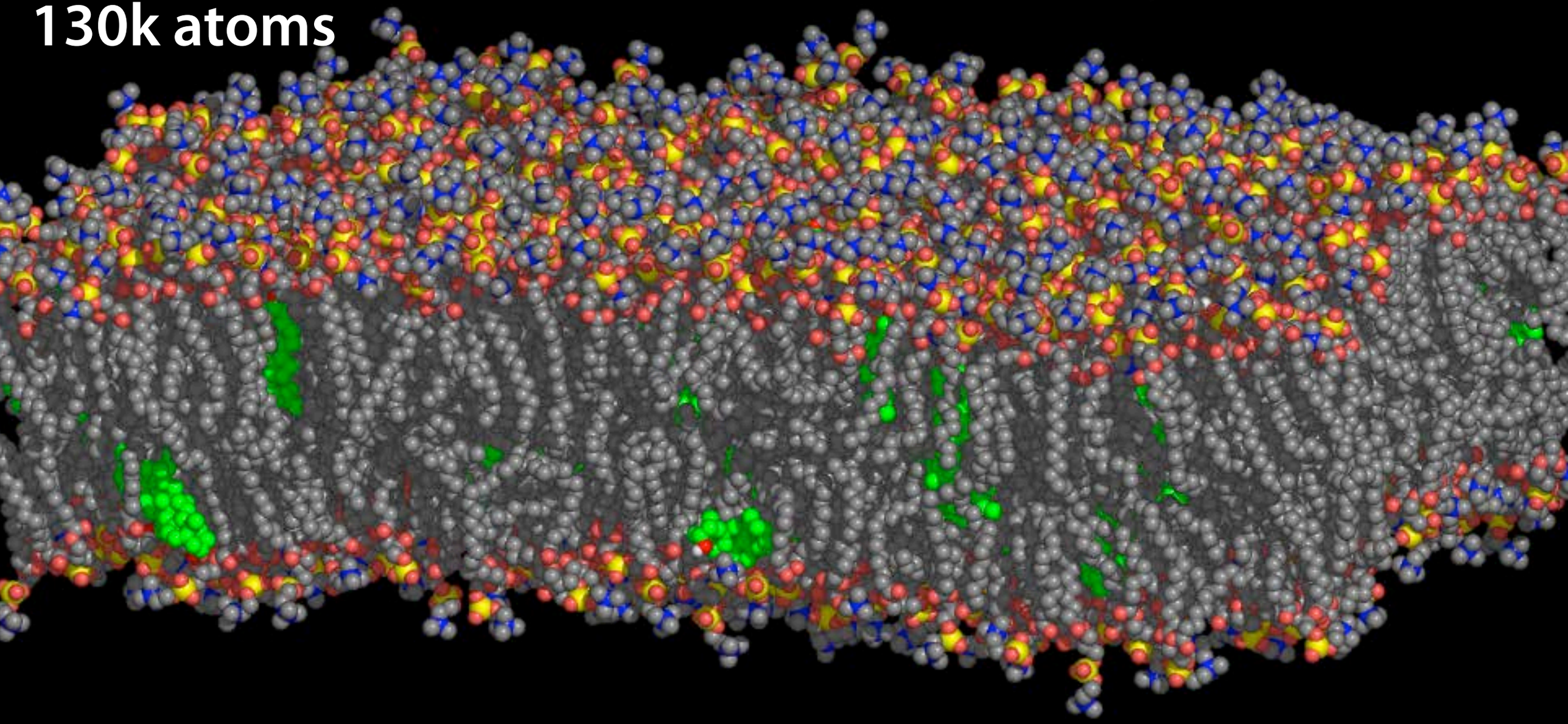
... you need 200 million iterations to reach
1 μs of simulated time

To achieve that in a day (86,400 seconds)...

...each iteration must complete in 432 (wallclock) μs

How not to do it:
(Old scaling data from 2008)
DPPC & Cholesterol
130k atoms

Blue Gene/L & Blue Matter:
scaled to 3 atoms/CPU
~10ns/day on 8192 CPUs



GROMACS 3: 2ns/day

...on a single dual
dual-core Opteron!

It is easier to get a reference problem/algorithm to scale

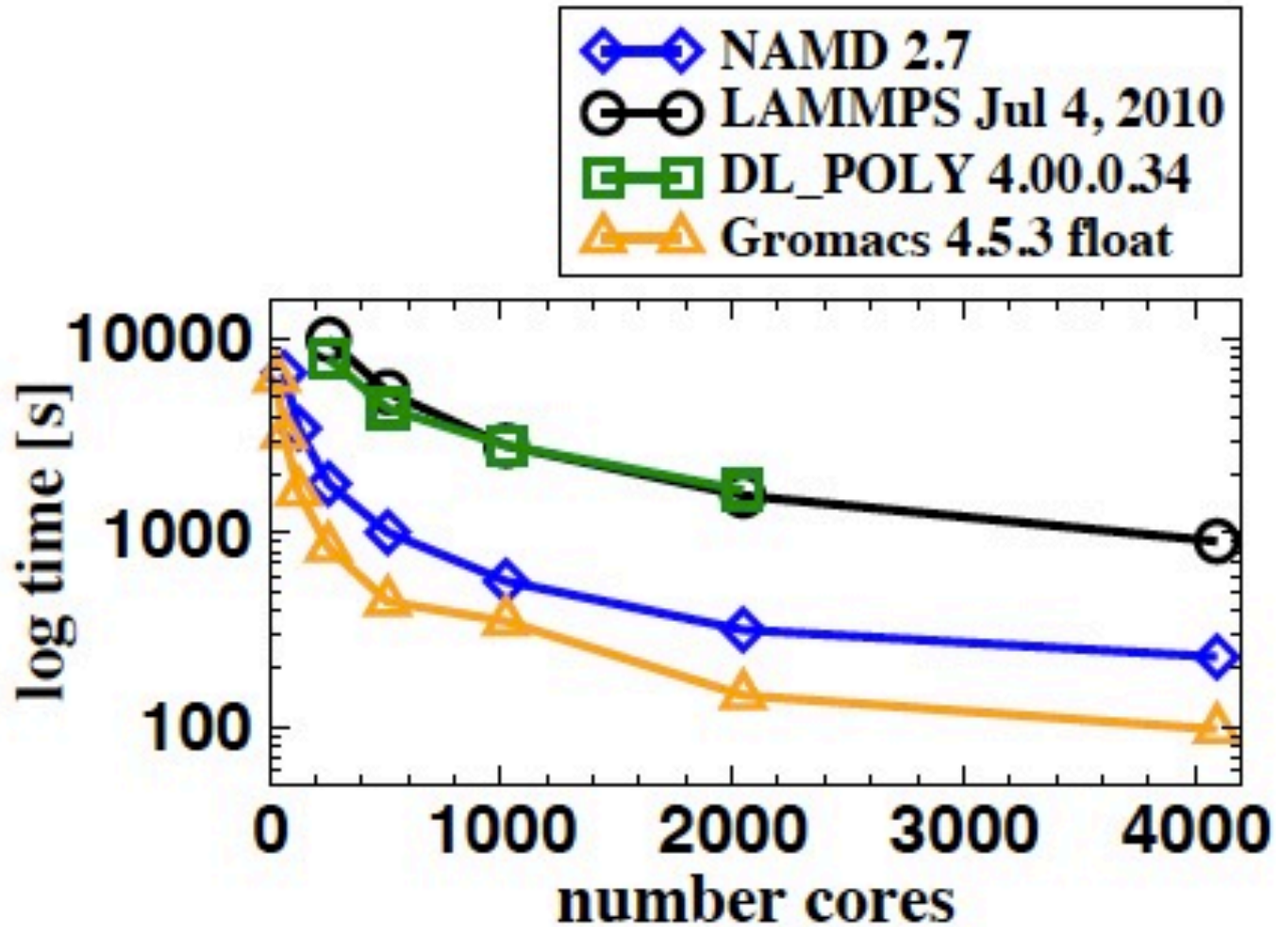
i.e., you see much better *relative* scaling before introducing any optimization

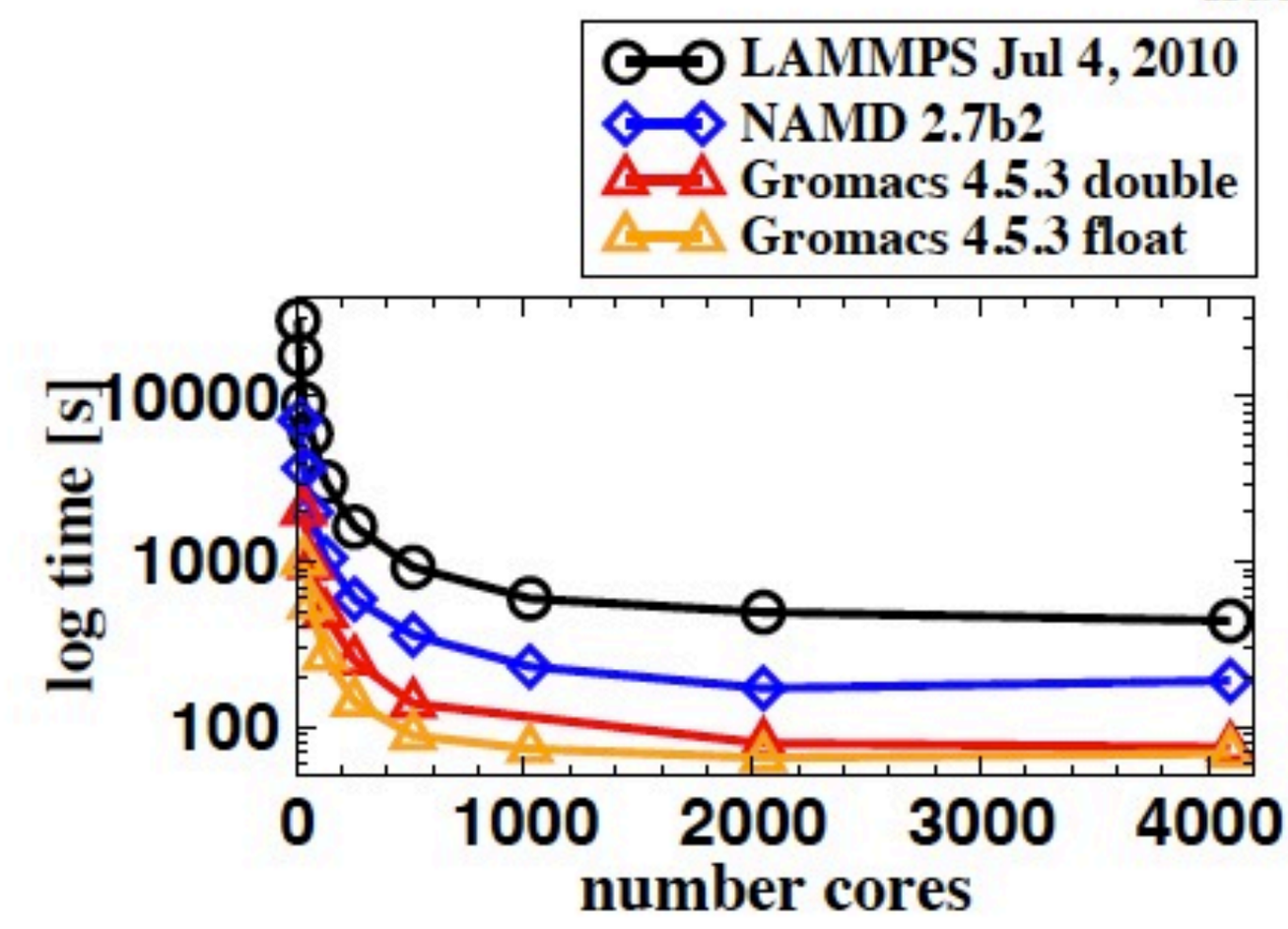
But conversely: The relevant comparison for GPUs is not a slow reference implementation running on a single core!

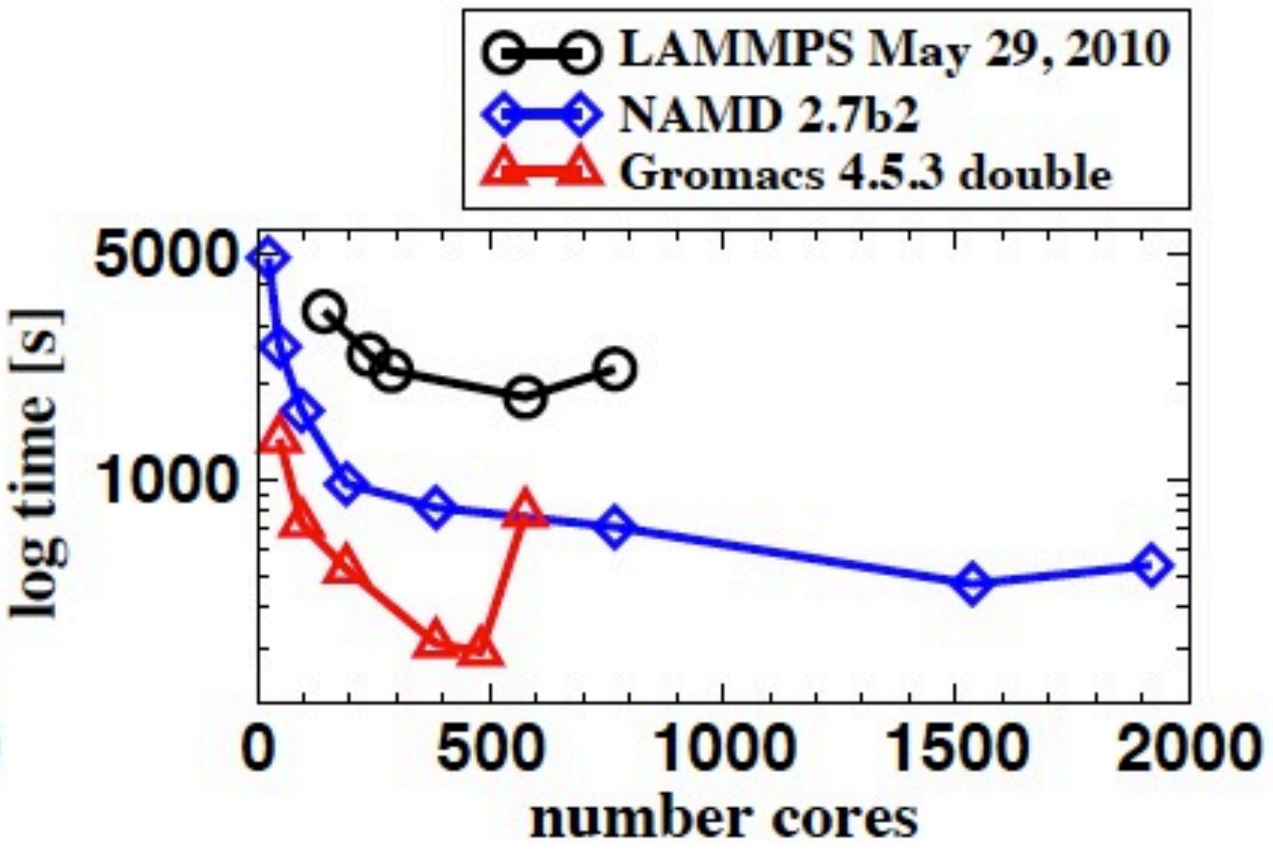
Even with amazing network we used to hit a limit at ~250 atoms/core

100 atoms/core is certainly within reach, maybe 10, not 1

We need *faster* nodes, not just more nodes at lower clock


NAMD 2.7
LAMMPS Jul 4, 2010
DL_POLY 4.00.0.34
Gromacs 4.5.3 float
log time [s]
10000
1000
100
0
1000
2000
3000
4000
number cores

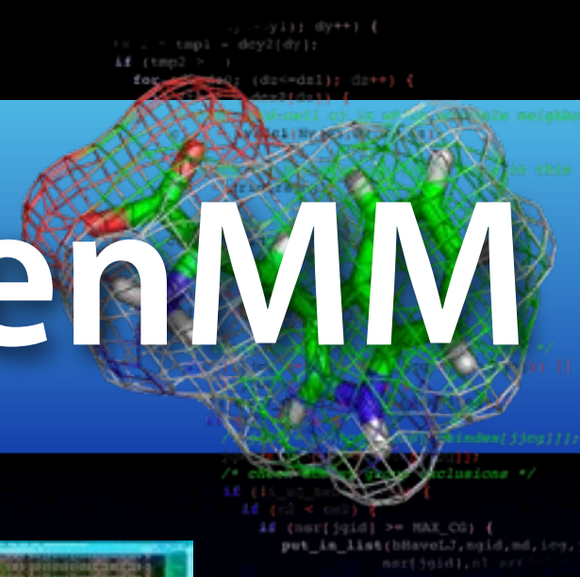

LAMMPS Jul 4, 2010
NAMD 2.7b2
Gromacs 4.5.3 double
Gromacs 4.5.3 float
log time [s]
10000
1000
100
0
1000
2000
3000
4000
number cores


LAMMPS May 29, 2010
NAMD 2.7b2
Gromacs 4.5.3 double
log time [s]
5000
1000
0
500
1000
1500
2000
number cores

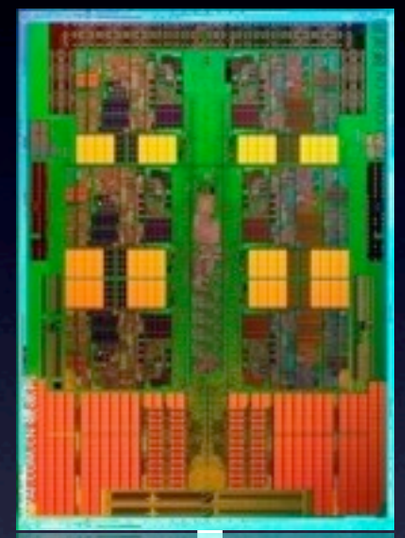
Early GPU acceleration


GT21
TERRATEC
258

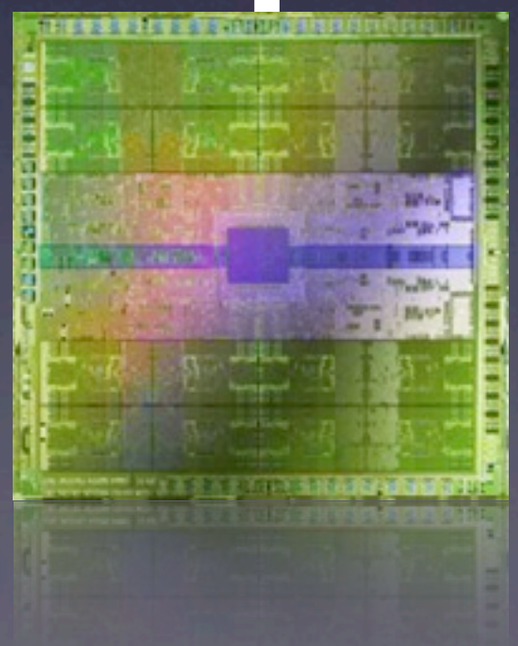
Gromacs-4.5 with OpenMM



Gromacs running
entirely on CPU as
an interface

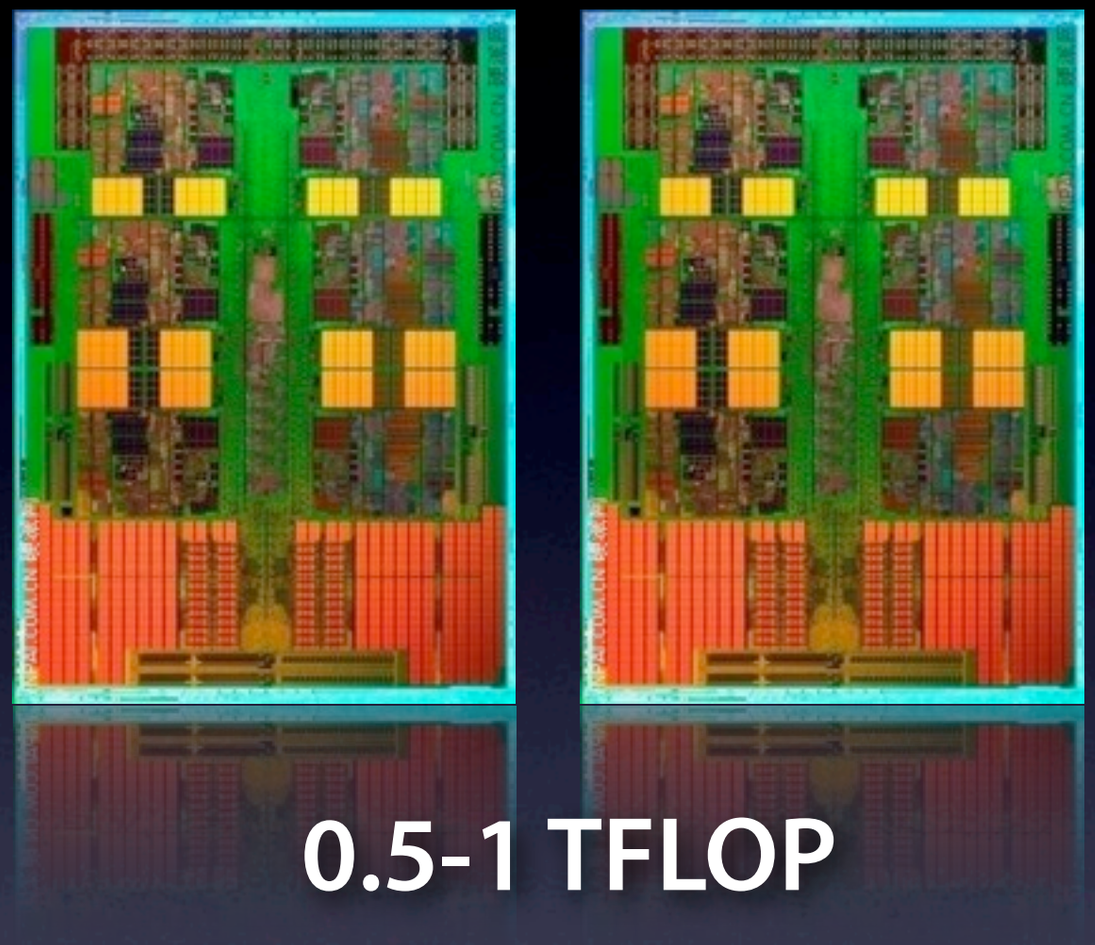


Actual simulation running
entirely on GPU
using OpenMM kernels



Only a few algorithms worked

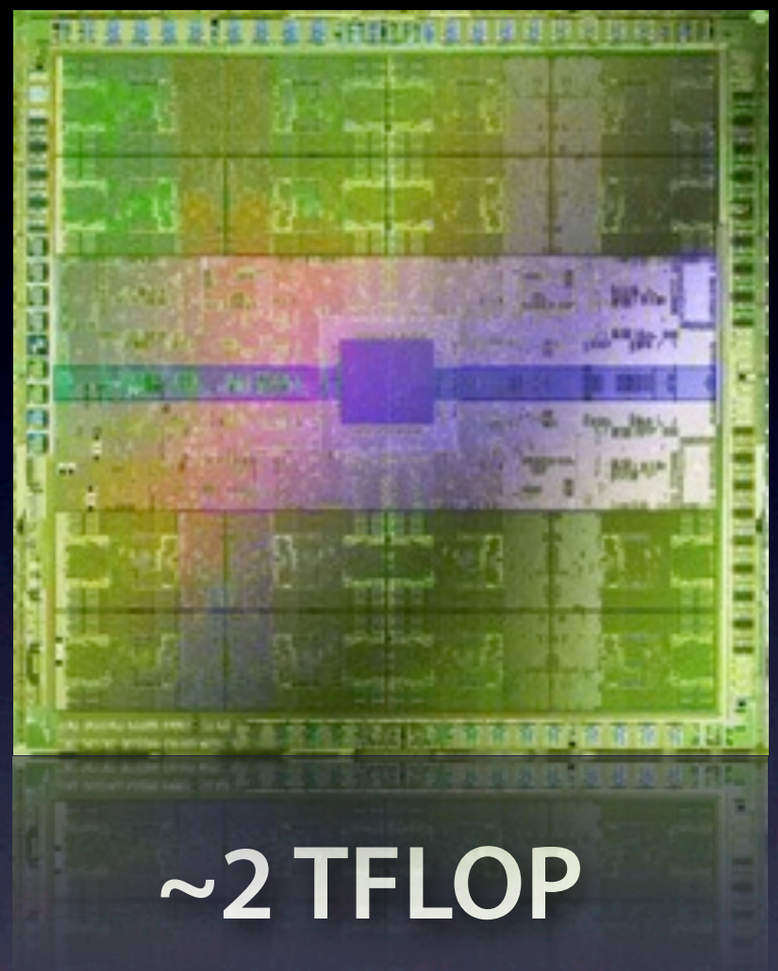
Why do we use the CPU too?



0.5-1 TFLOP

Random memory access OK (not great)

Great for complex latency-sensitive stuff (domain decomposition, etc.)

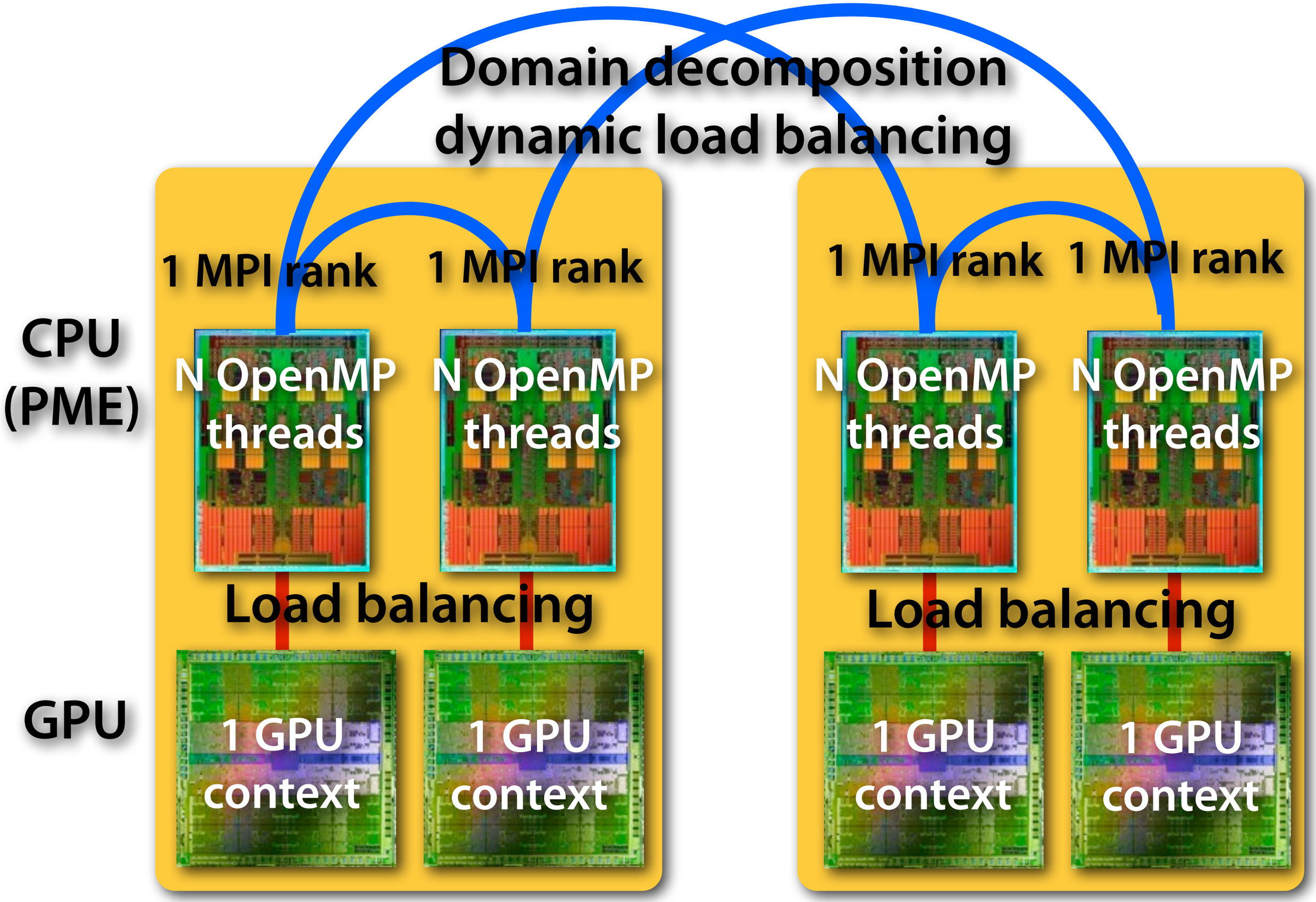


~2 TFLOP

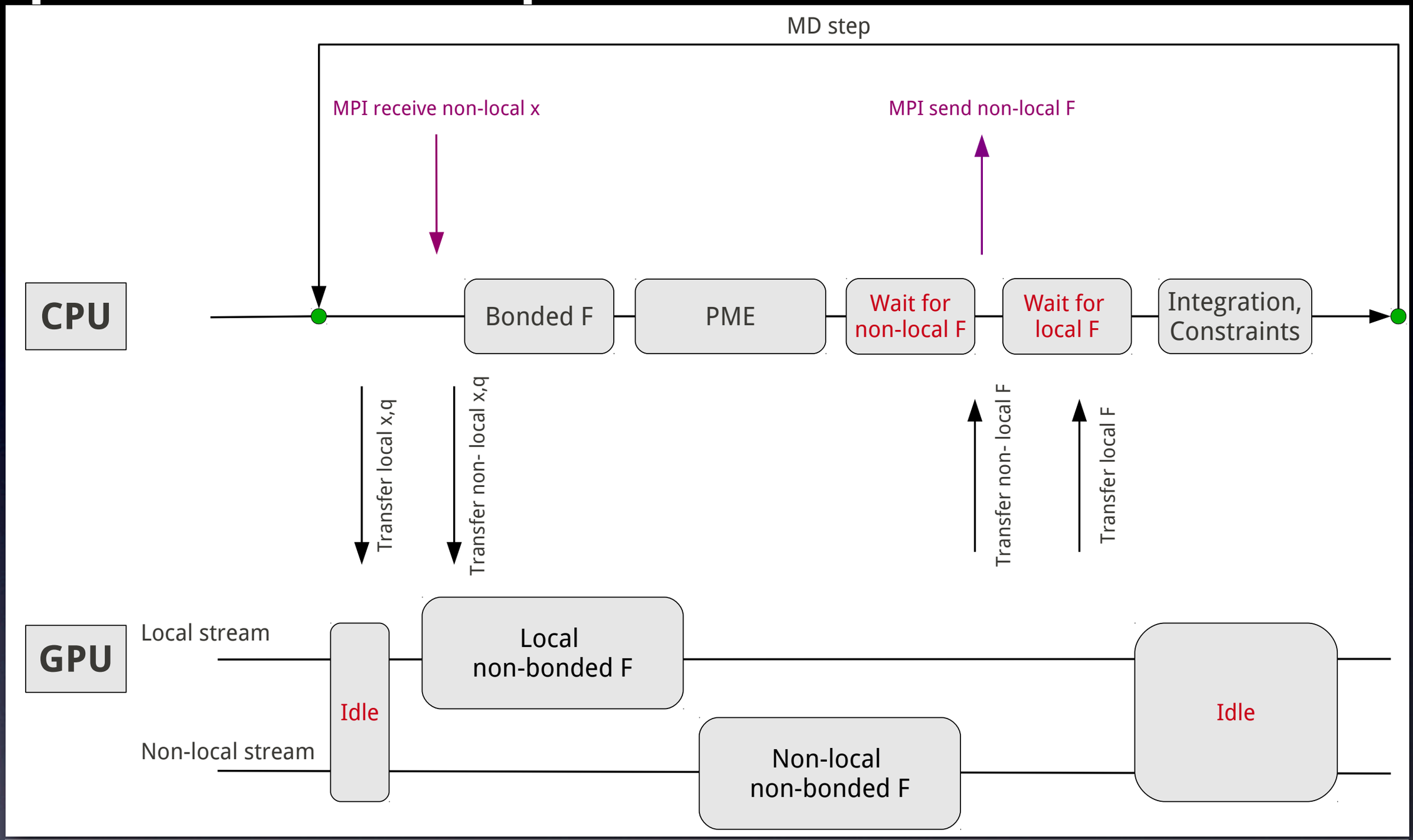
Random memory access won't work

Great for throughput

Gromacs-4.6 next-generation GPU implementation:

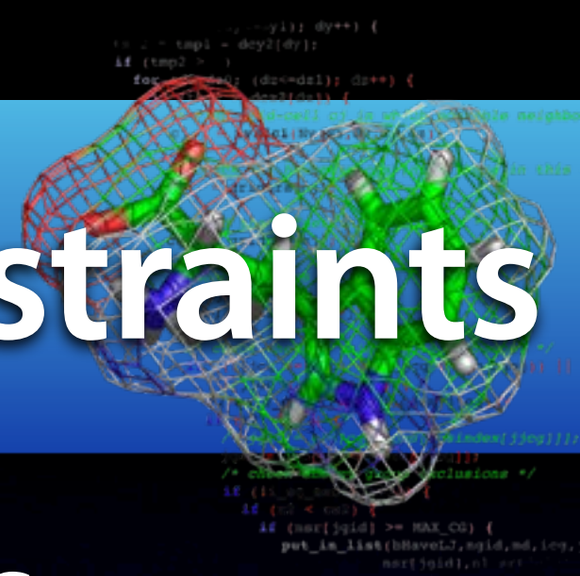


Simplified execution path



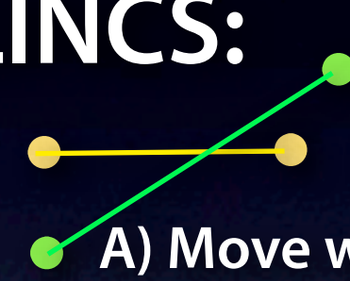
Wallclock time for an MD step:
~0.5 ms if we want to simulate 1µs/day

CPU trick 1: all-bond constraints

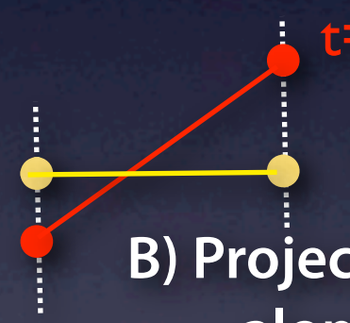


- **Δt limited by fast motions - 1fs**
  - **Remove bond vibrations**
- **SHAKE (iterative, slow) - 2fs**
  - **Problematic in parallel (won't work)**
  - **Compromise: constrain h-bonds only - 1.4fs**
- **GROMACS (LINCS):**
  - **LINear Constraint Solver**
  - ***Approximate* matrix inversion expansion**
  - **Fast & stable - much better than SHAKE**
  - **Non-iterative**
  - **Enables 2-3 fs timesteps**
  - **Parallel: P-LINCS (from Gromacs 4.0)**

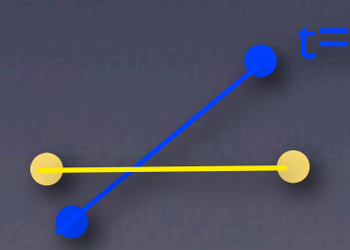
LINCS:



A) Move w/o constraint

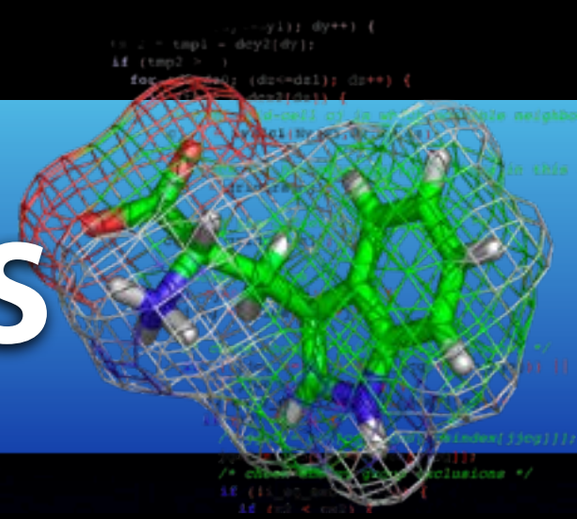


B) Project out motion along bonds

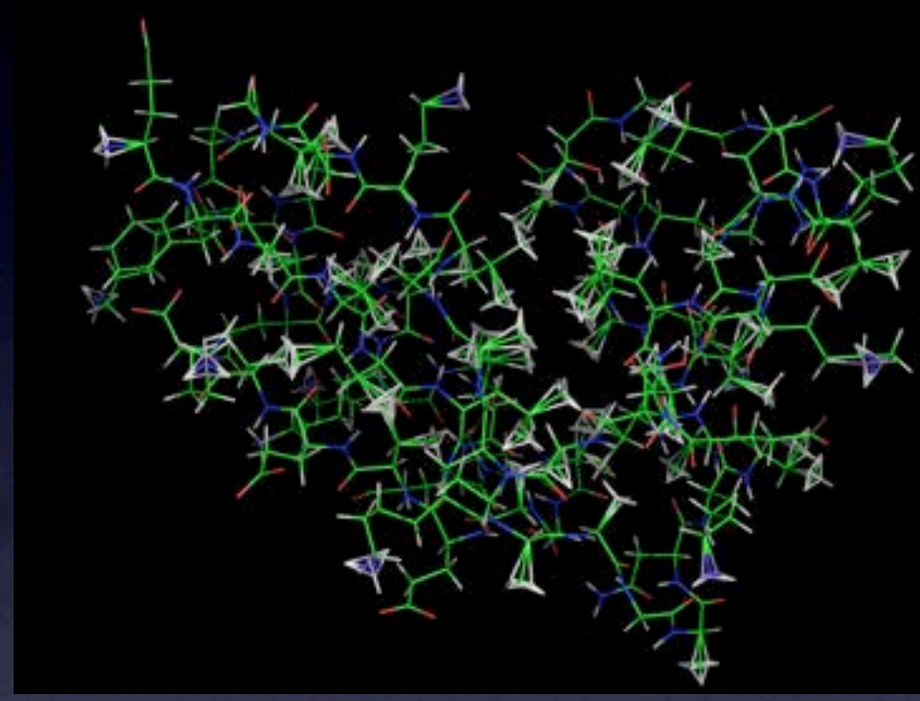


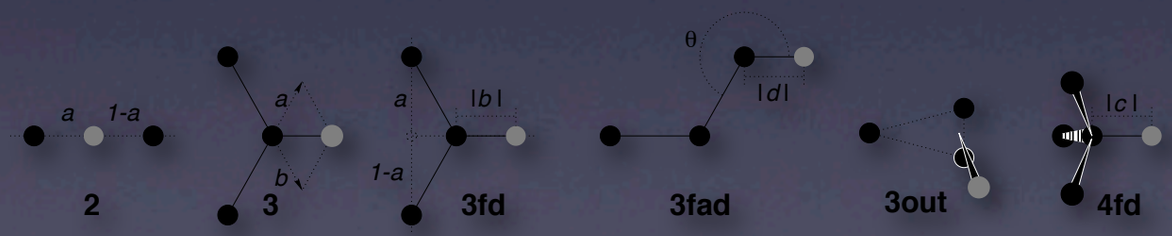
C) Correct for rotational extension of bond

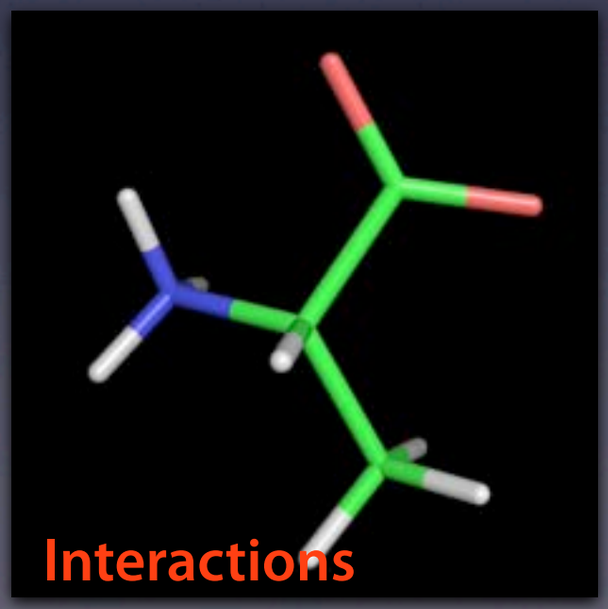
CPU trick 2: Virtual sites

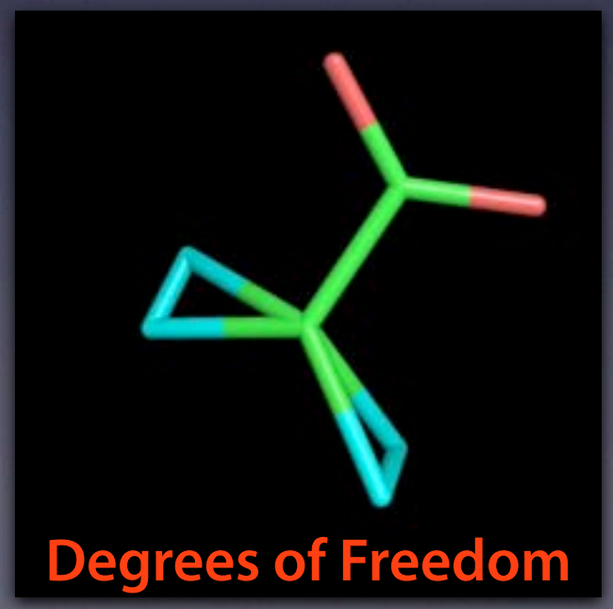


- Next fastest motions is H-angle and rotations of CH_3/NH_2 groups
- Try to remove them:
  - Ideal H position from heavy atoms.
  - CH_3/NH_2 groups are made rigid
  - Calculate forces, then project back onto heavy atoms
  - Integrate only heavy atom positions, reconstruct H's
- Enables 5fs timesteps!

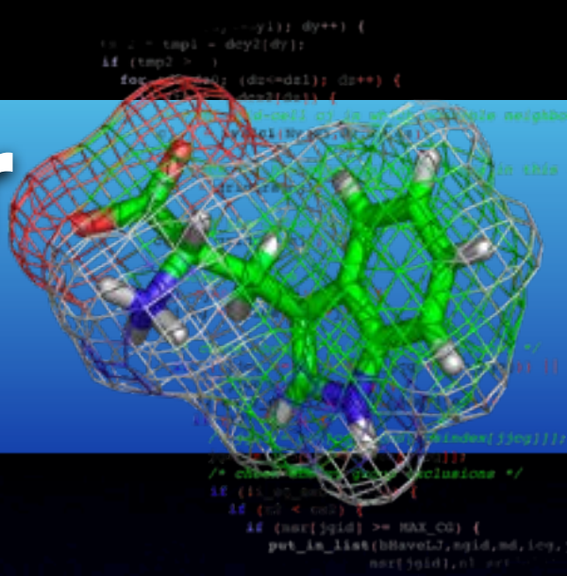


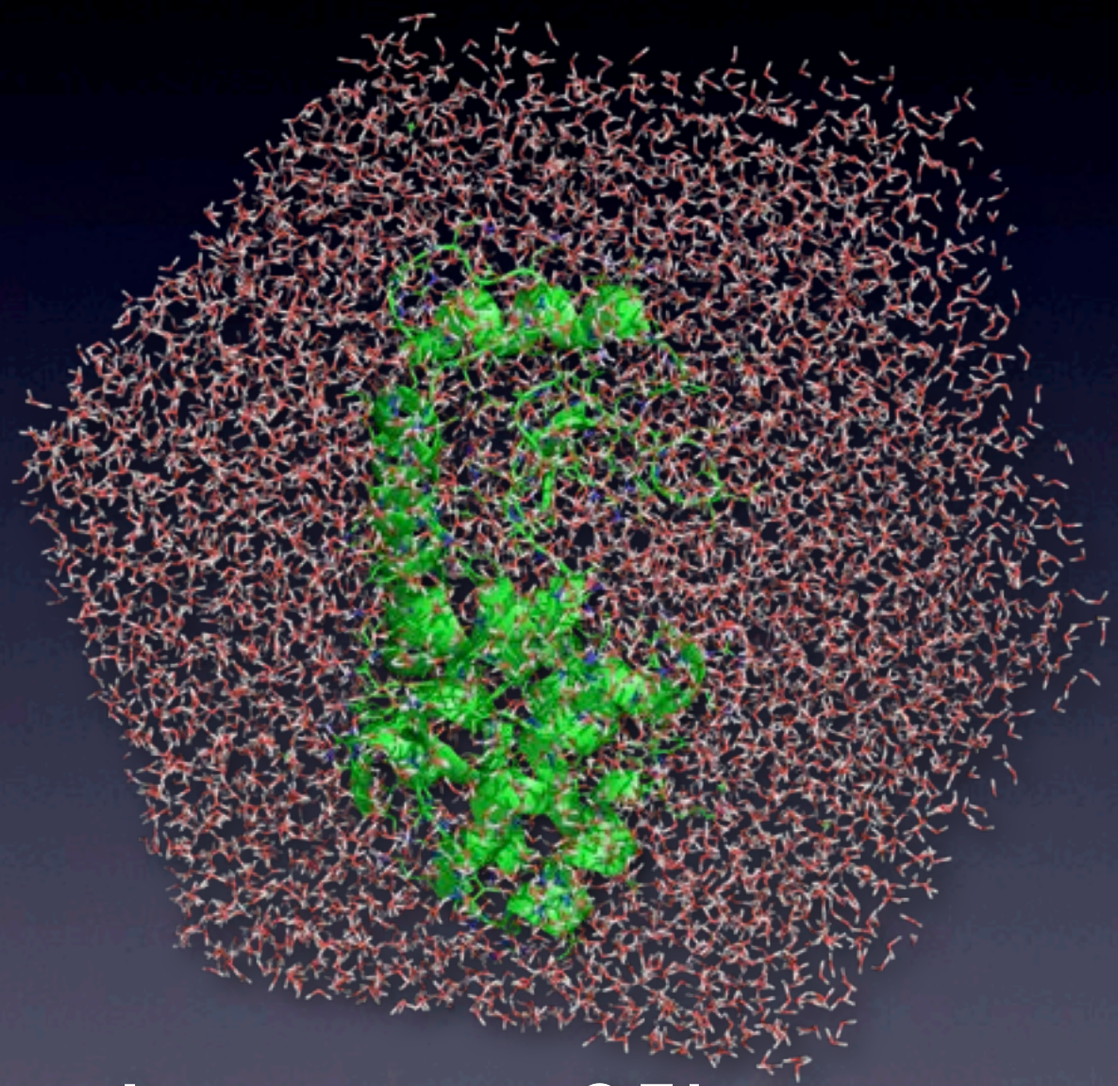




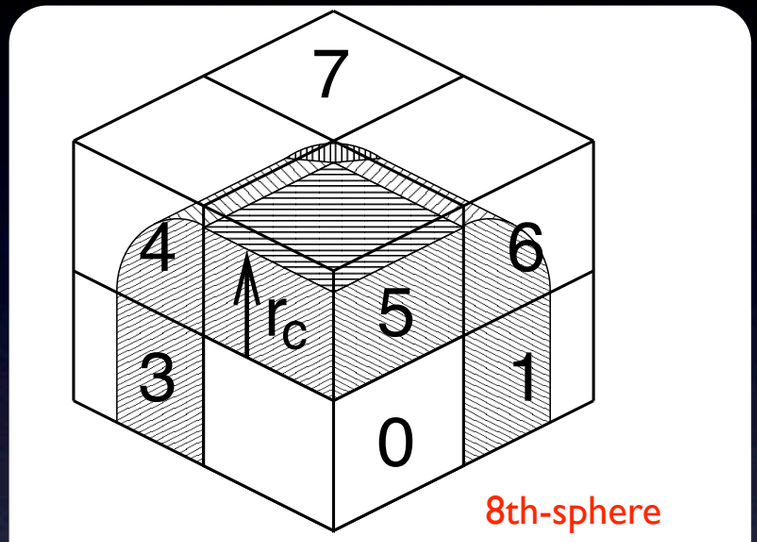


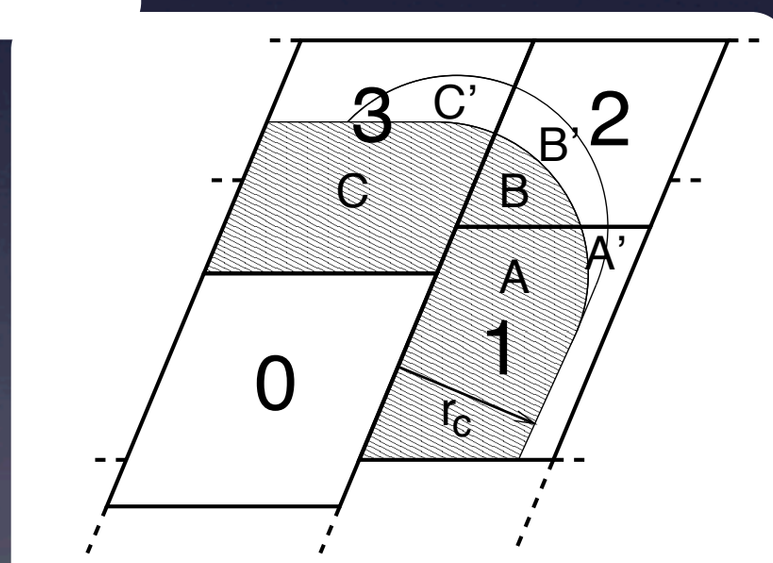
CPU trick 3: Non-rectangular cells & decomposition





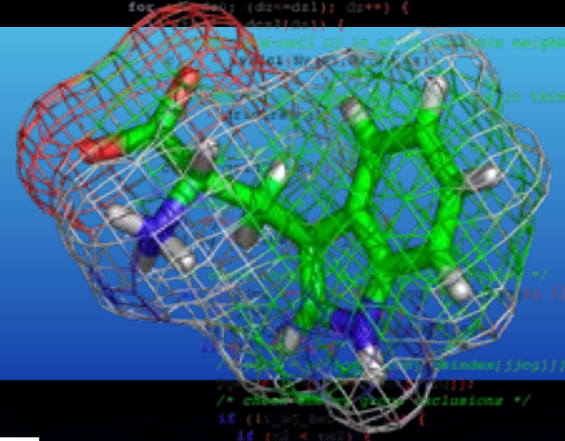
Lysozyme, 25k atoms
Rhombic dodecahedron
(36k atoms in cubic cell)

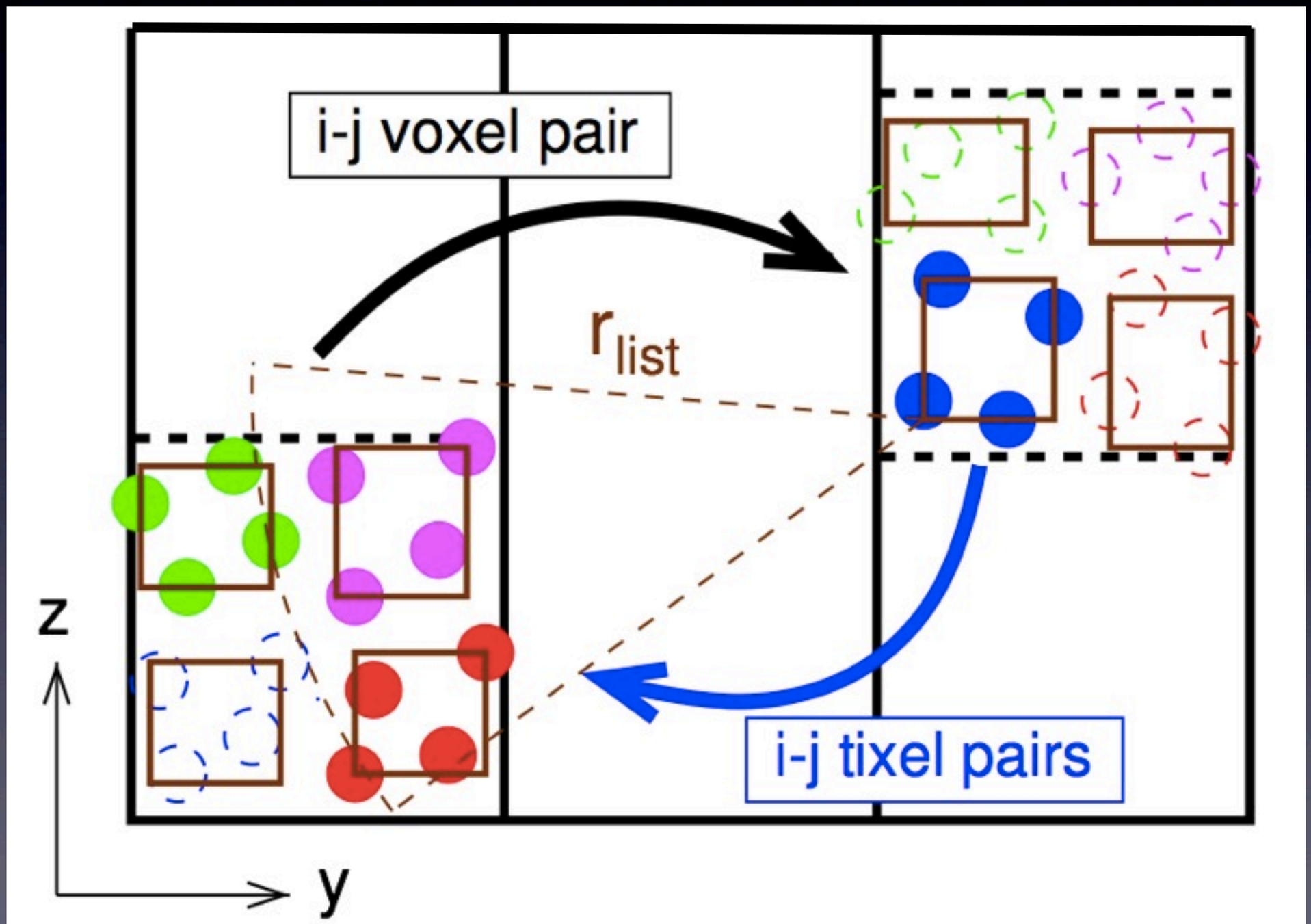




Load balancing works for arbitrary triclinic cells

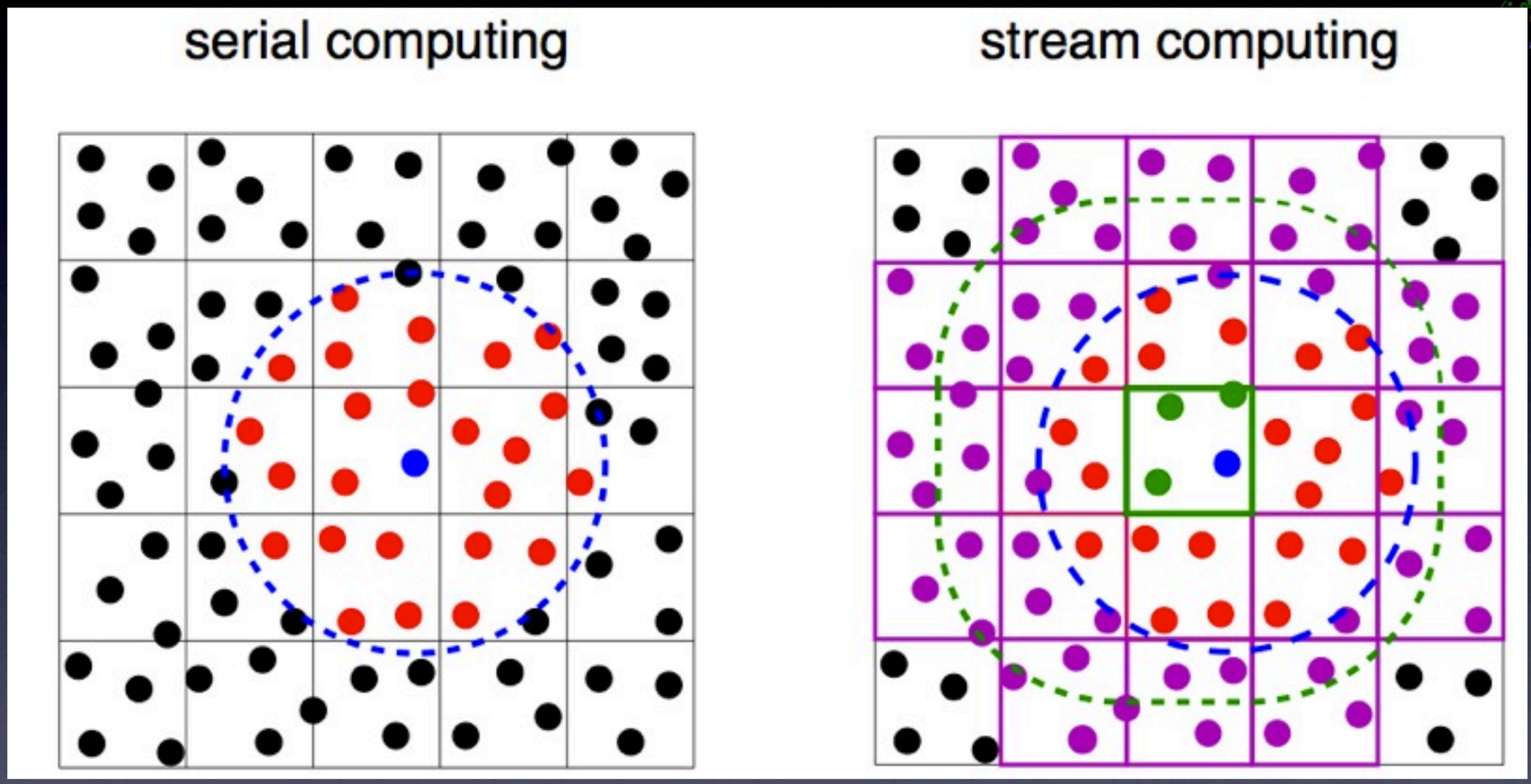
From neighborlists to pairs of proximity cells





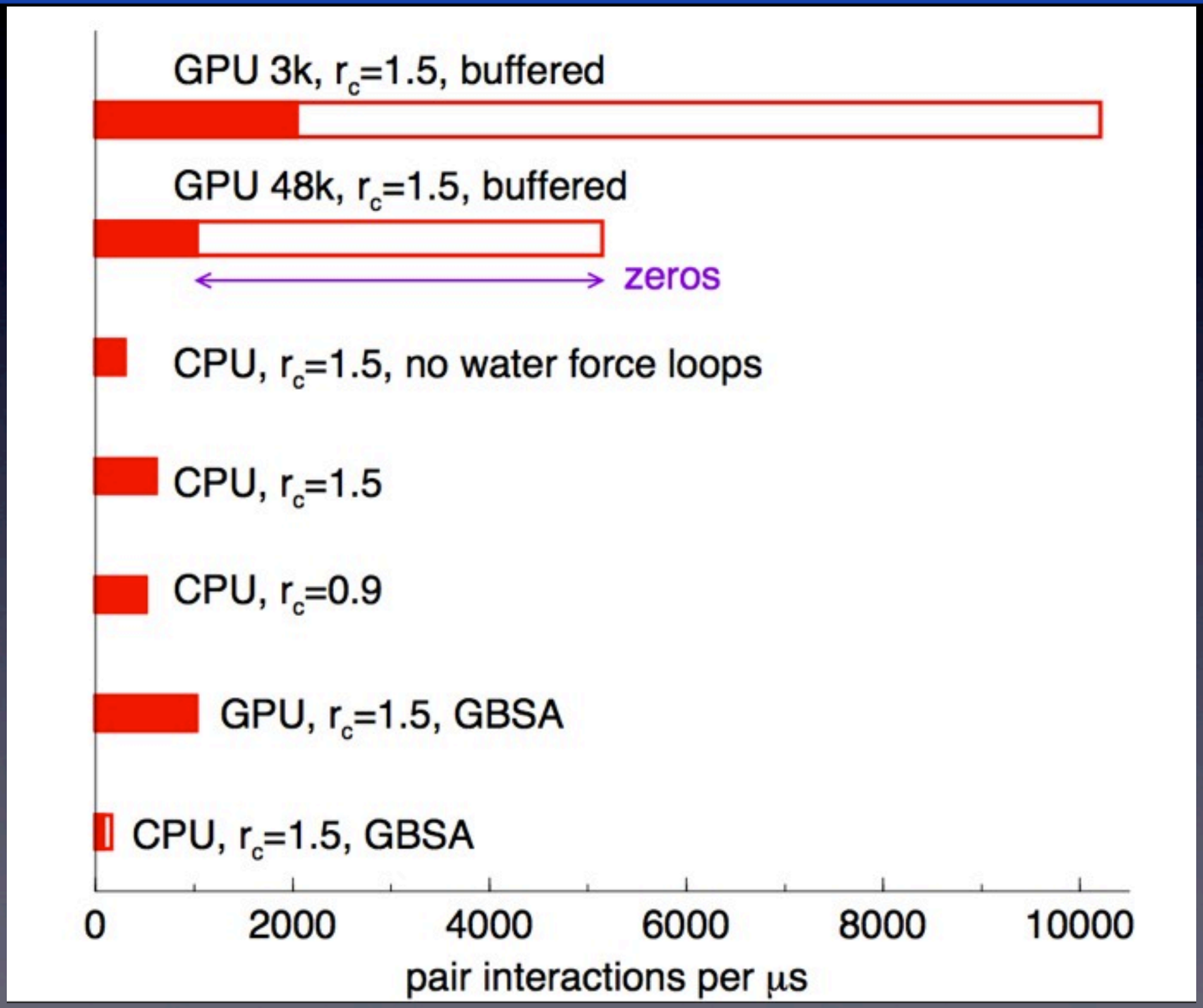
Organize as tiles with all-vs-all interactions:

Tiling circles is difficult!



- You need a lot of cubes to cover a sphere
- All interactions beyond cutoff need to be zero

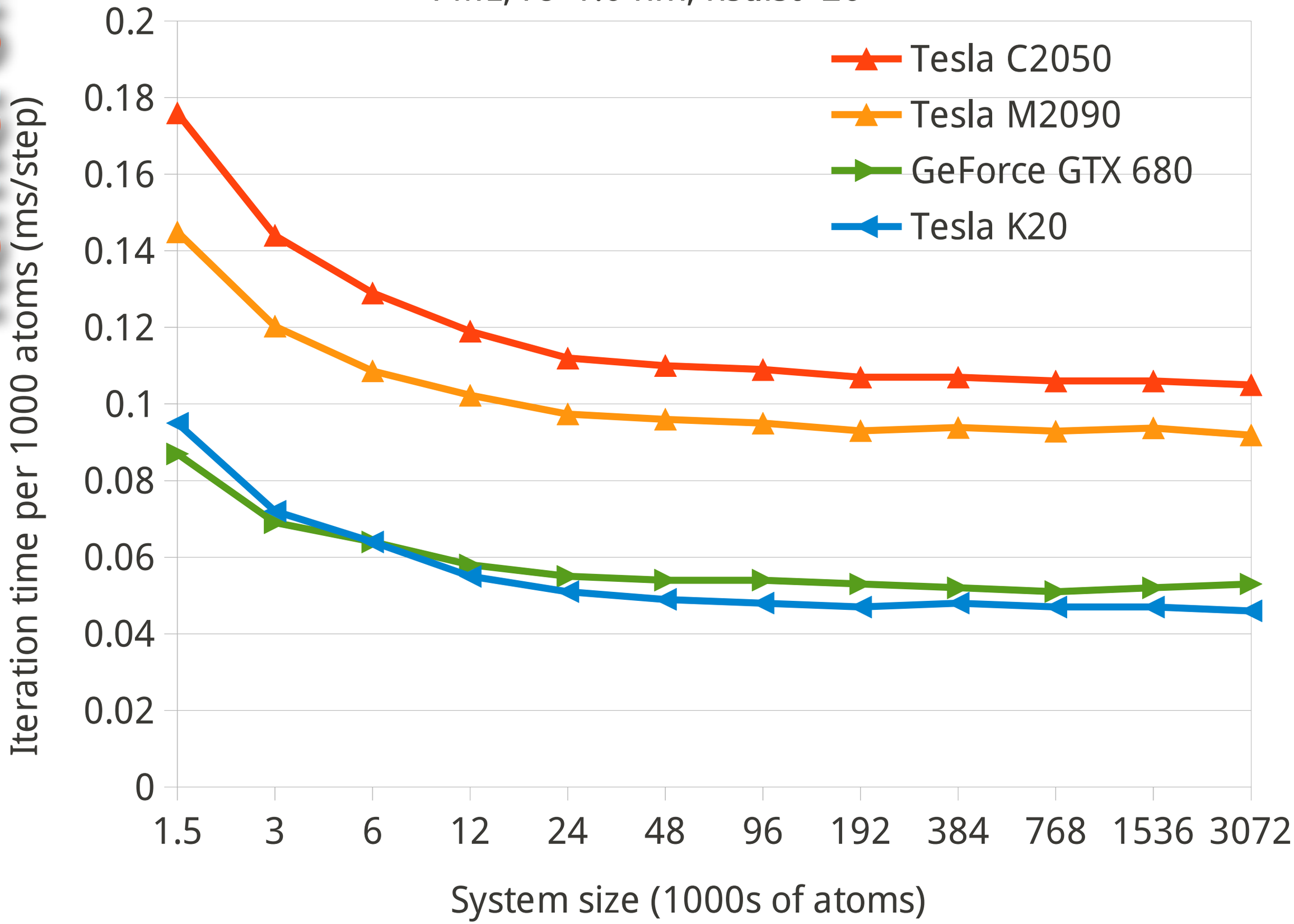
The art of calculating zeros



Kernel only

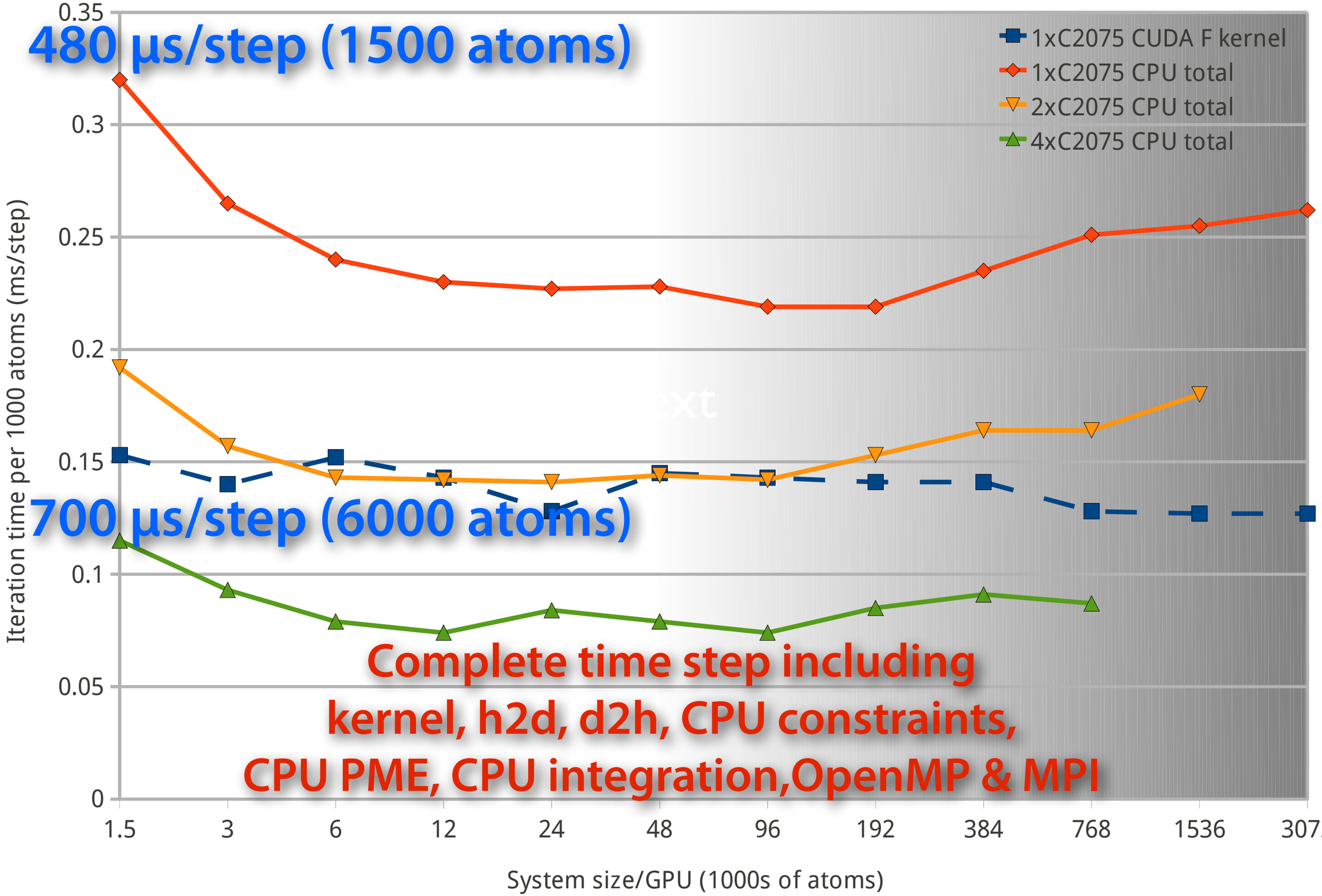
CUDA non-bonded force kernel scaling

PME, rc=1.0 nm, nstlist=20

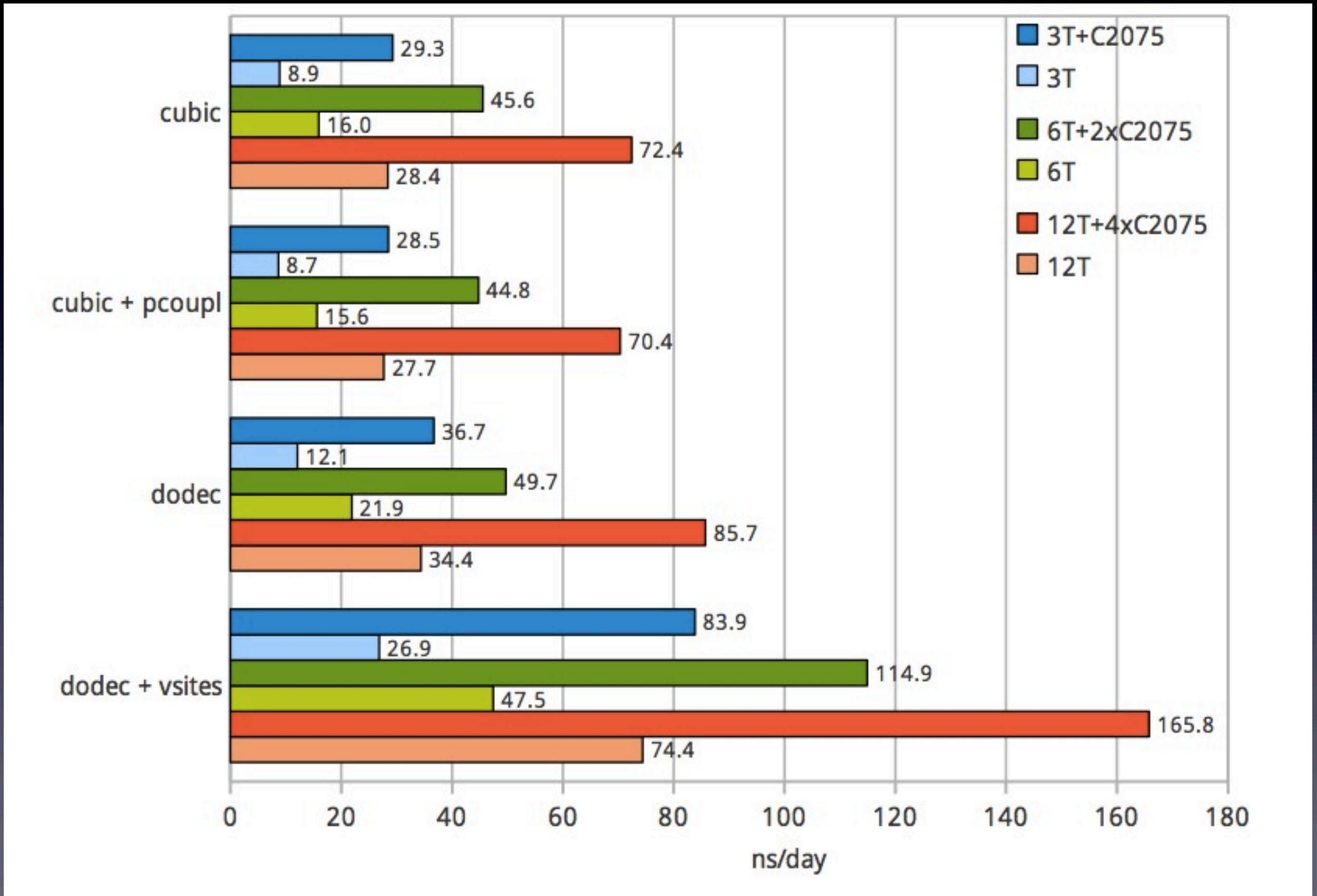


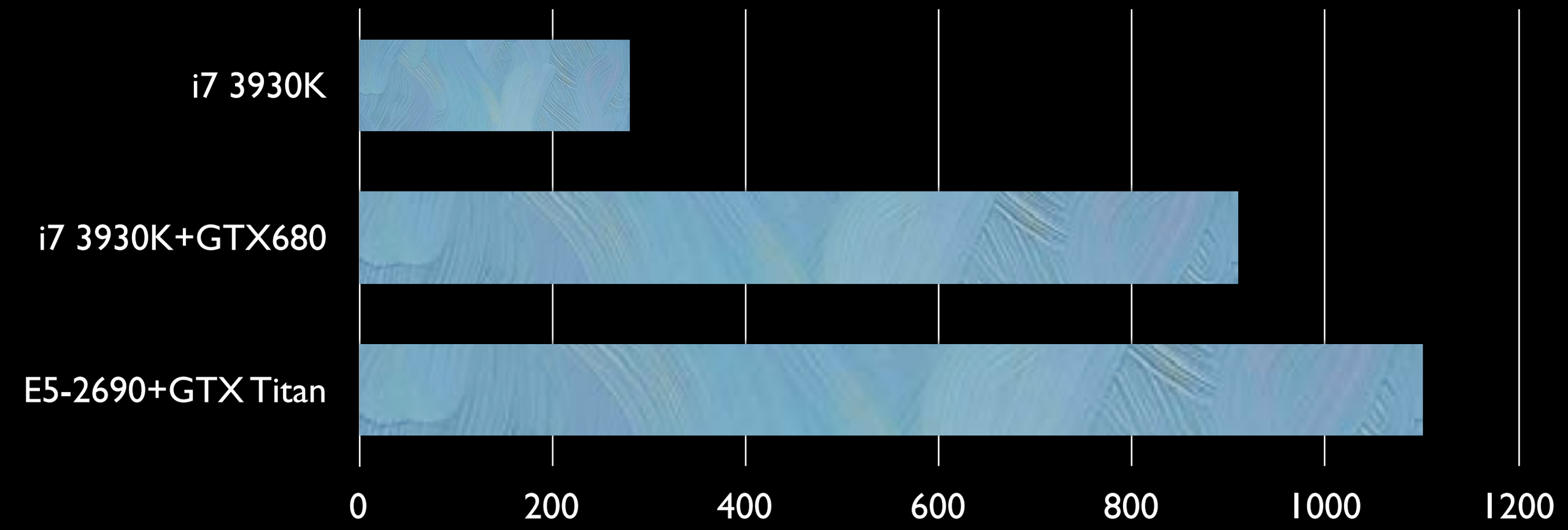
PME weak scaling

Xeon X5650 3T + C2075 / process

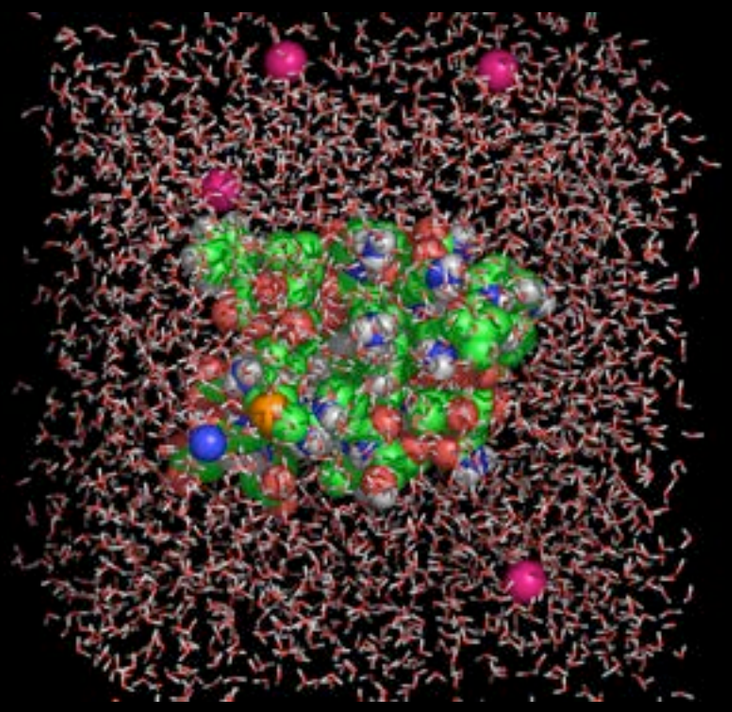


Example performance: 24,000 atom protein (ns/day)



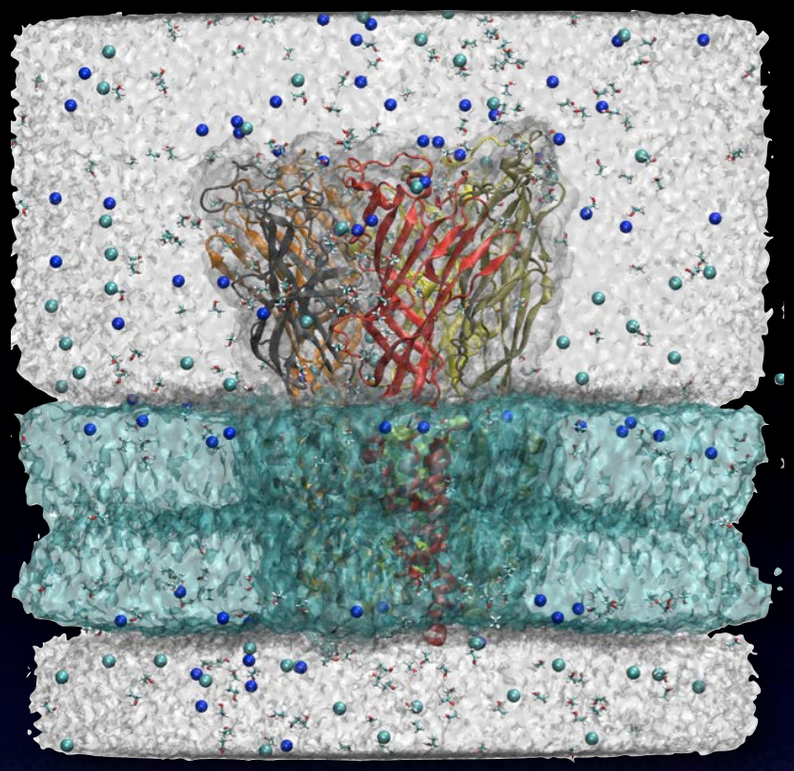


ns/day



The Villin headpiece
8,000 atoms
water
triclinic box
PME electrostatics

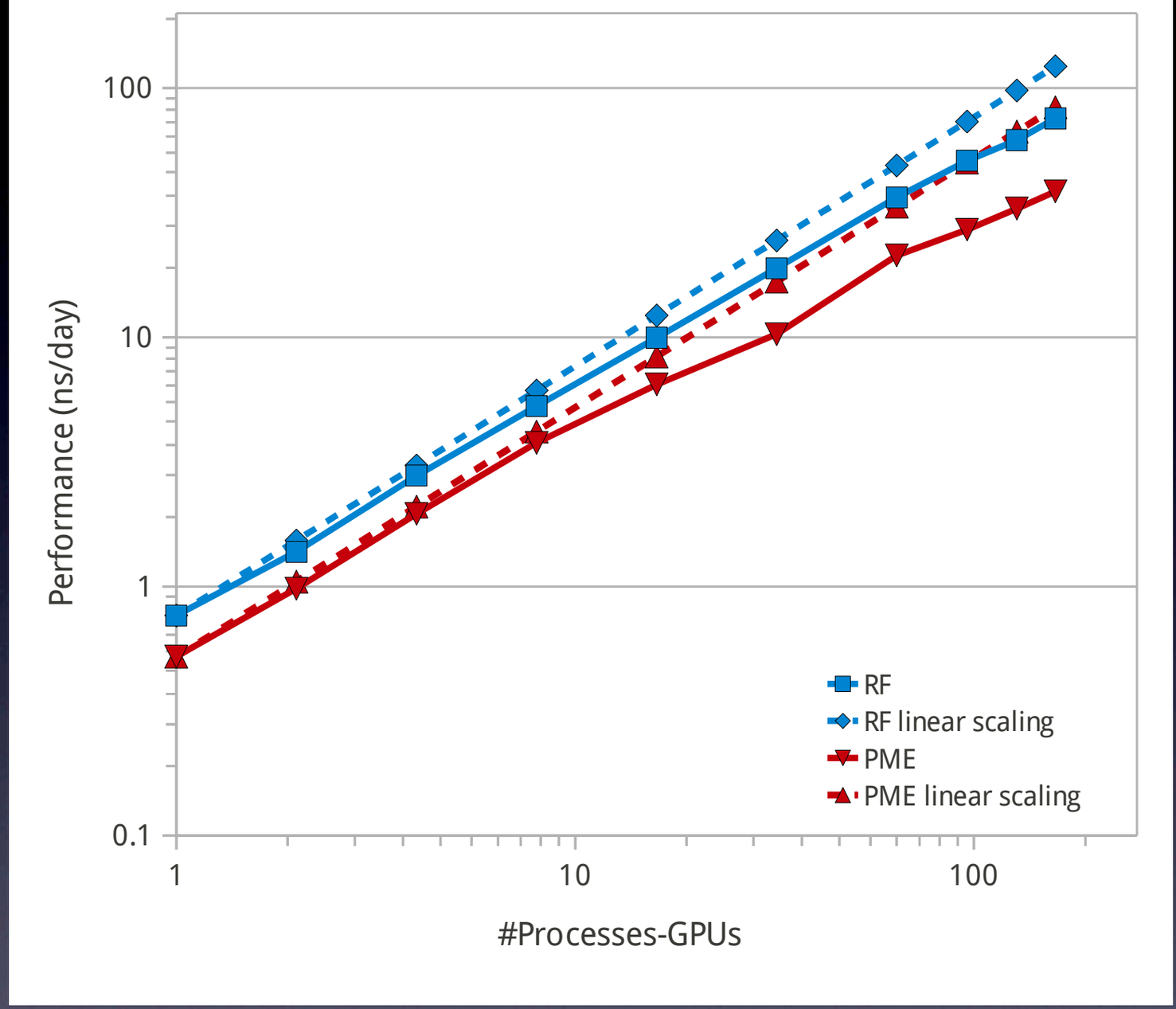
2546 FPS (beat that, Battlefield)



GLIC: Ion channel membrane protein

150,000 atoms

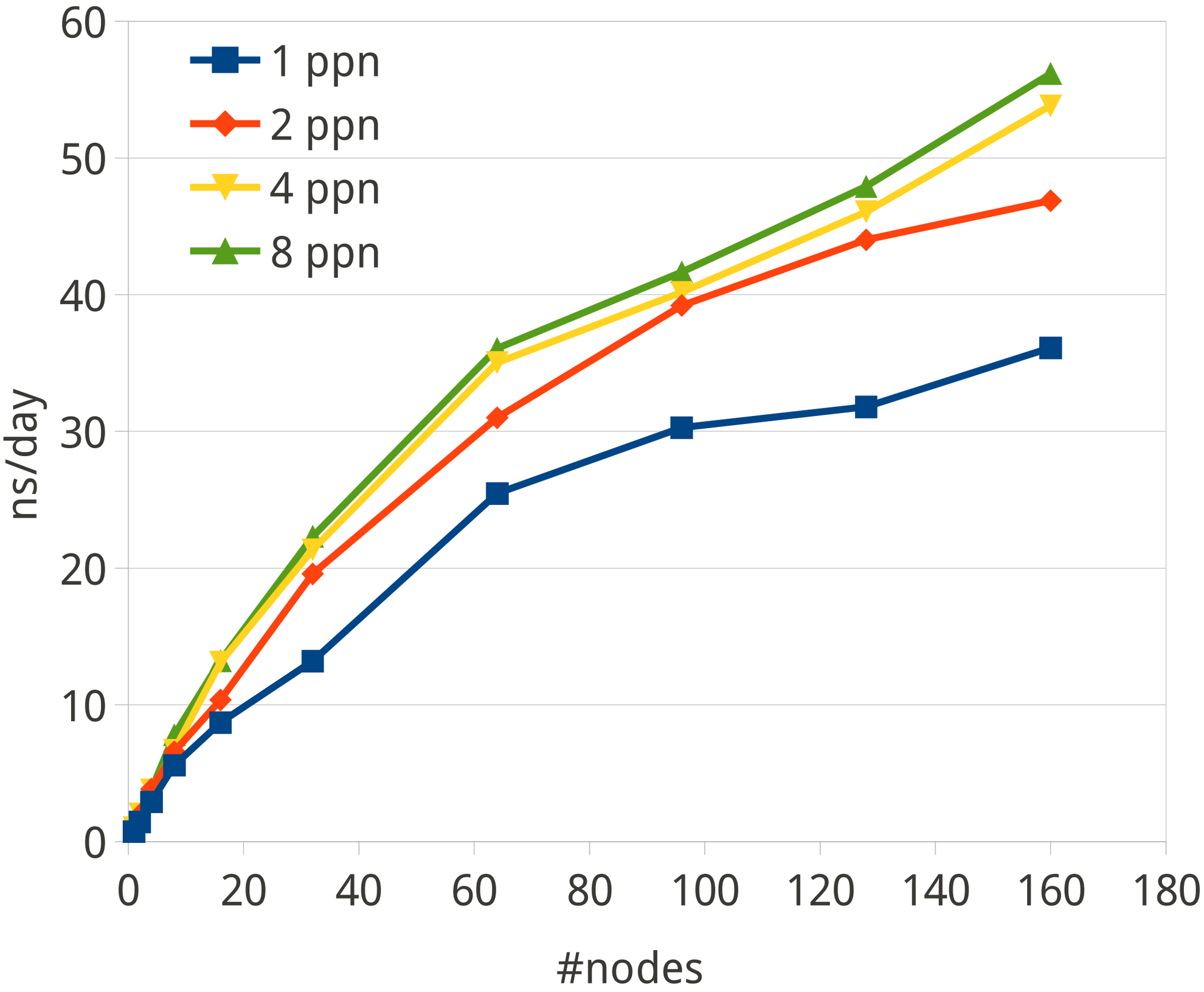
Scaling of Reaction-field & PME



Challenge: GROMACS has very short iteration times - hard requirements on latency/bandwidth

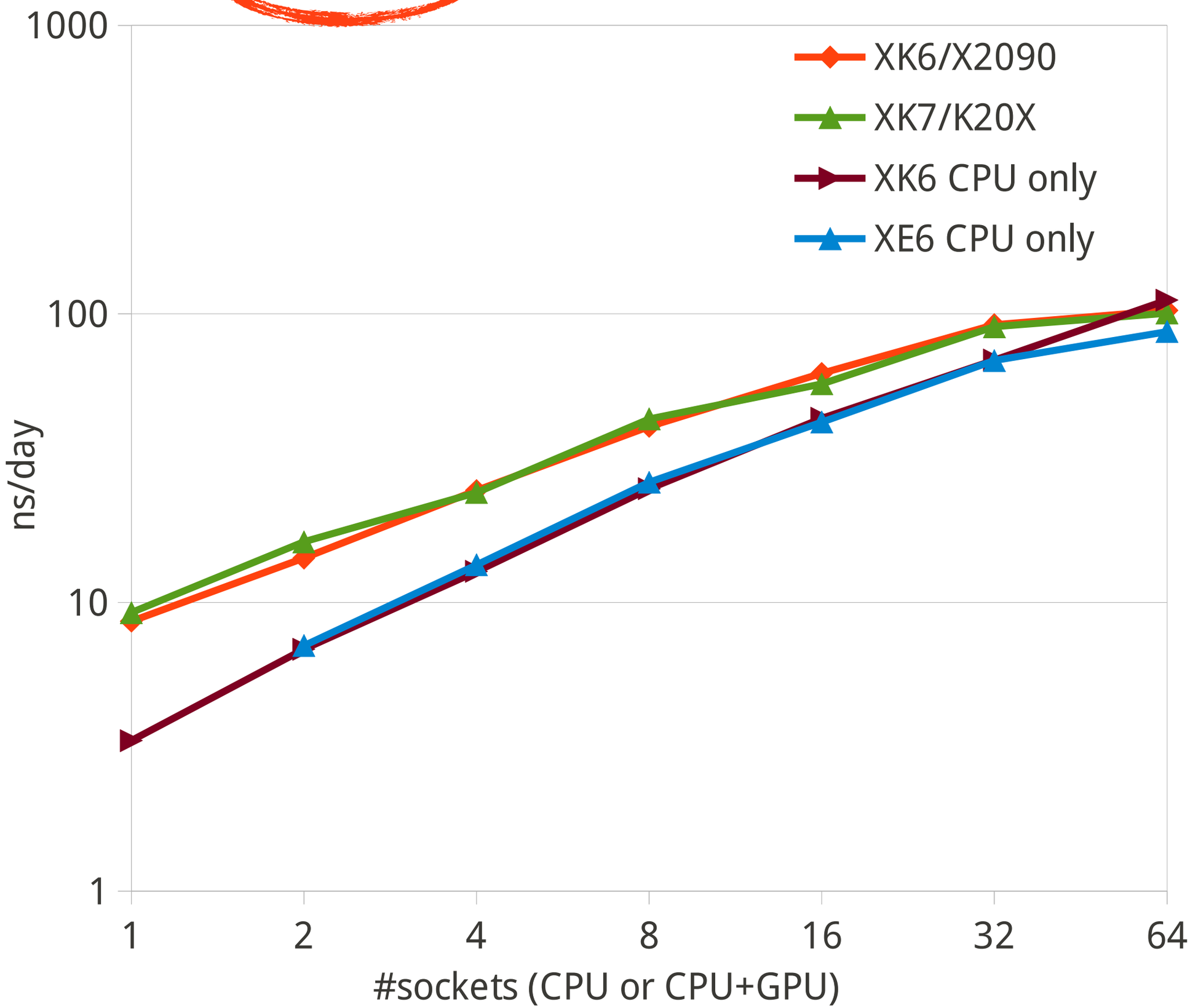
Strong scaling on Cray XK6

Input: water-box 1.5M atoms, PME, rc >= 0.9



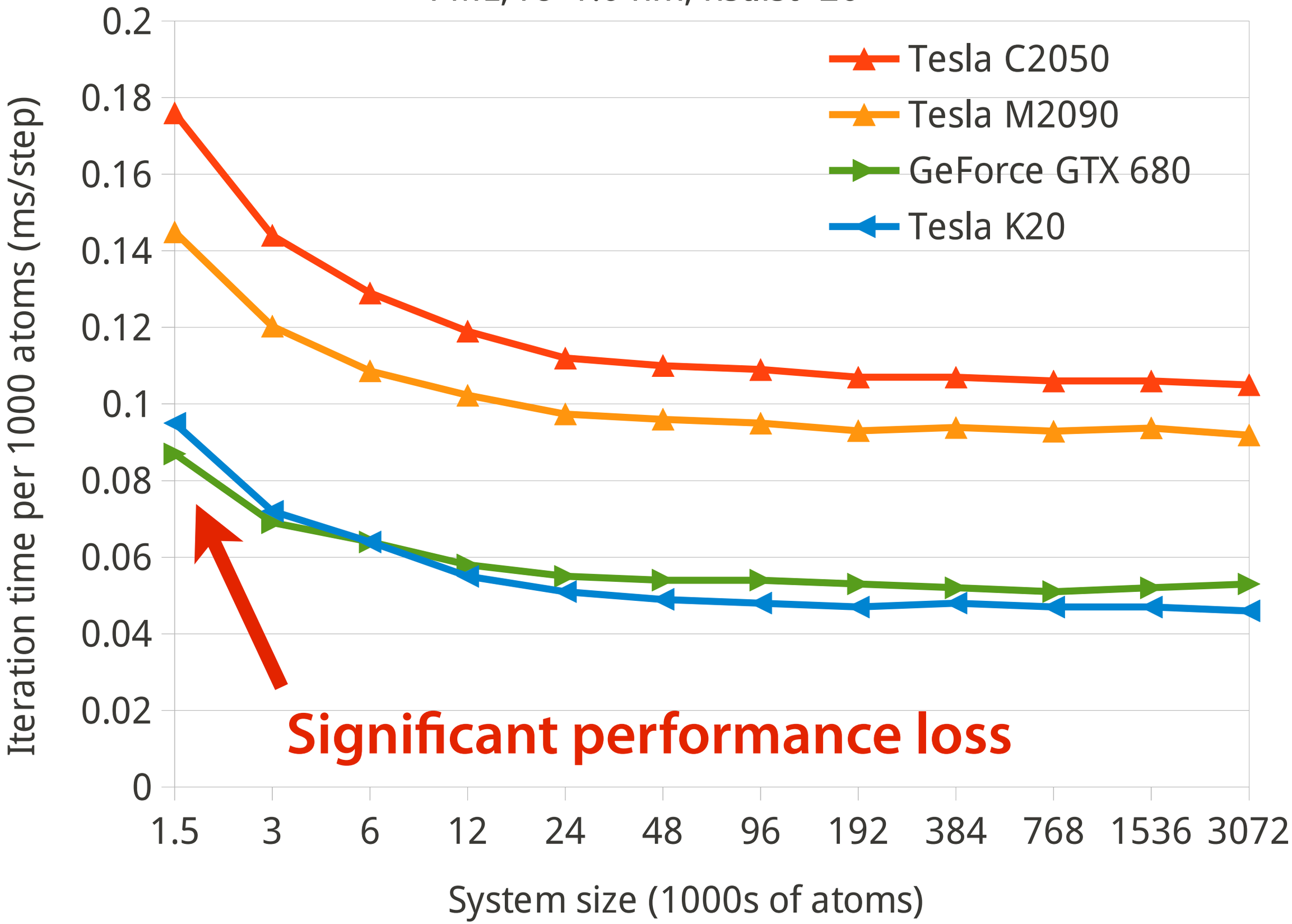
GROMACS 4.6 extreme scaling

Scaling to 130 atoms/core: ADH protein 134k atoms, PME, rc >= 0.9

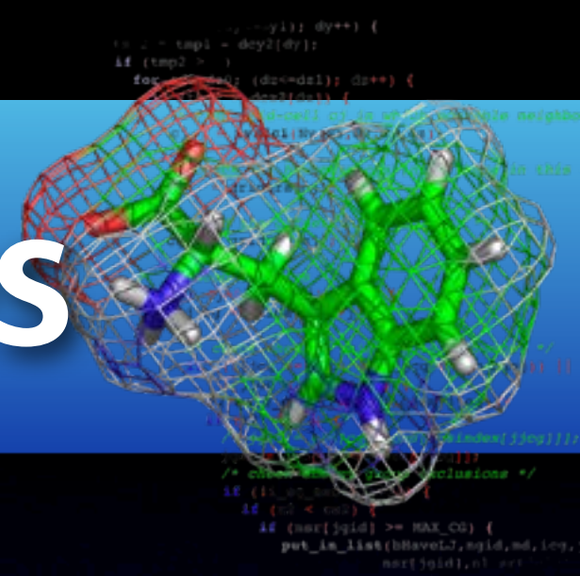


CUDA non-bonded force kernel scaling

PME, rc=1.0 nm, nstlist=20

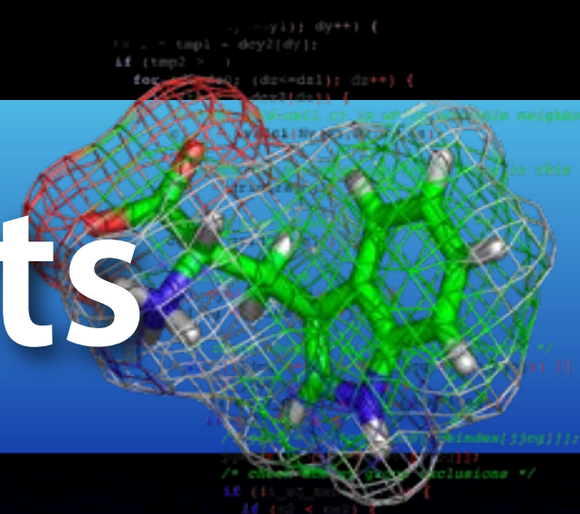


Remaining challenges



- Increase performance for small systems - necessary for improved parallelization
- Overlap communication/calculation (GPU-CPU-MPI)
- Keep all SMs busy 100% of time
- Reduce latencies
- Interested in the gory details?
  Szilárd Páll, Wednesday 10am
  Room212B

Acknowledgments



- **GROMACS:** Berk Hess, David van der Spoel, Per Larsson
- **Gromacs-GPU:** Szilard Pall, Berk Hess, Rossen Apostolov
- **Multi-Threaded PME:** Roland Shultz, Berk Hess
- **Copernicus:** Sander Pronk, Iman Pouya, Peter Kasson
- **Nvidia:** Mark Berger, Scott LeGrand, Duncan Poole,Andrew Walsh

erc European Research Council

SJUNDE RAMPROGRAMMET

Vetenskapsrådet

Stiftelsen för Strategisk Forskning

NVIDIA