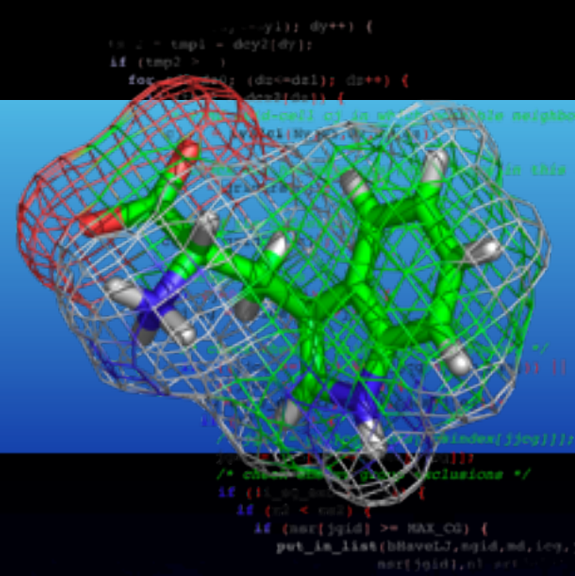
High performance computing with GROMACS

Berk Hess

hess@cbr.su.se
Center for Biomembrane Research
Stockholm University, Sweden

GROMACS history

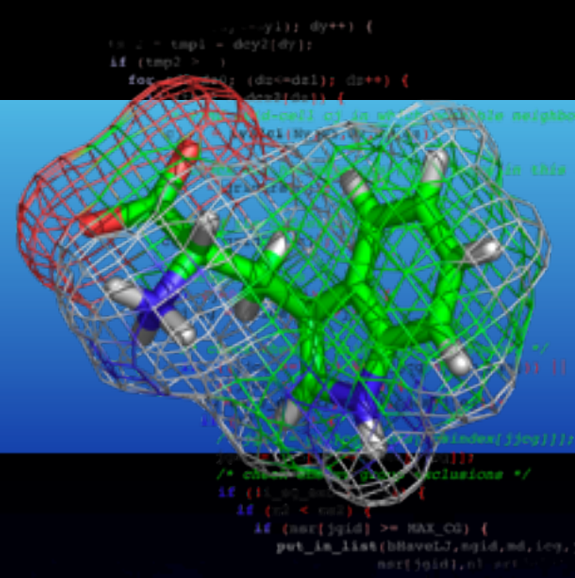


- Started in early 90's in Groningen (Netherlands)
- Originally parallel hardware and software
- Initially, focus has been mainly on high performance on small numbers of processors
- Development of novel, efficient algorithms
- Highly efficient implementation

- The past few years: focus on parallel scaling

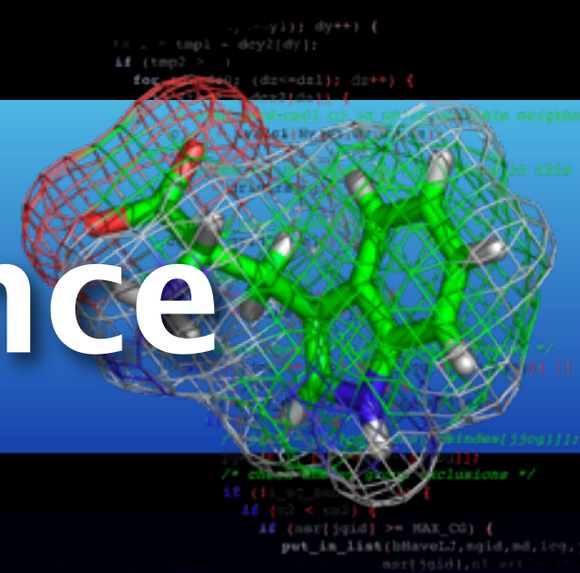
GPL, http://www.gromacs.org

GROMACS



- GROningen MAchine for Chemical Simulation
- Core developers:
  David van der Spoel (Groningen → Uppsala)
  Berk Hess (Groningen → Mainz → Stockholm)
  Erik Lindahl (Stockholm)
- Newer developers:
  Gerrit Groenhof (Groningen → Göttingen)
  Carsten Kutzner (Göttingen)
  Roland Schultz (Oak Ridge)
  Sander Pronk (Stockholm)
  . . .

Improving performance

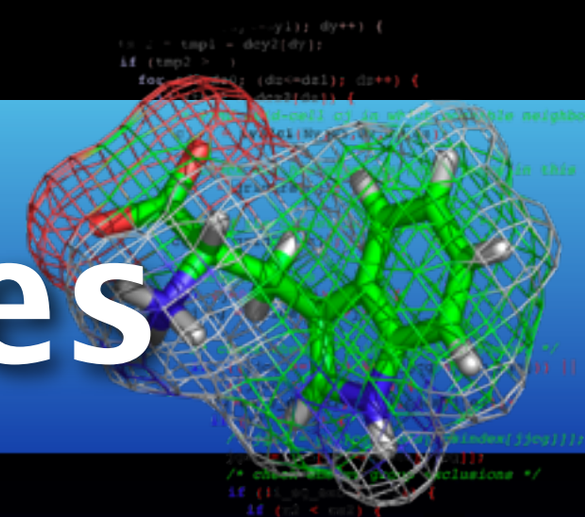


- Increasing the time step:
  - Use bond constraints, LINCS algorithm (2 fs)
  - Remove H-vibrations with virtual sites (5 fs)
  - **Performance increase: factor 2 or more!**
- Reducing the time per step
  - Efficient algorithms and code
  - Run in parallel over many processors

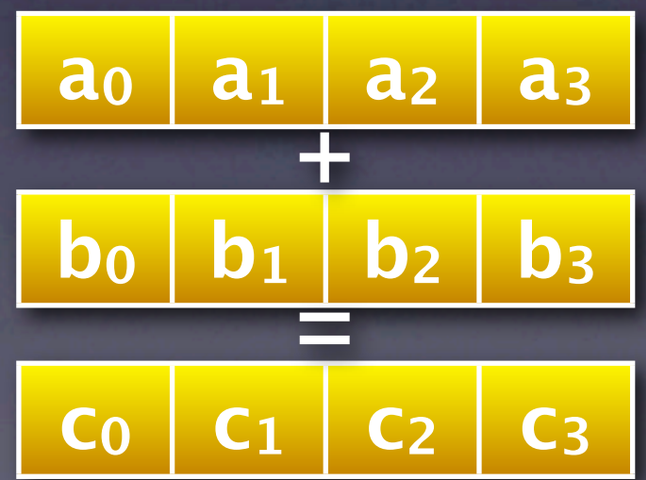
LINCS: Hess, Bekker, Fraaije, Berendsen J. Comp. Chem. 18, 1463 (1997)

virtual sites: Feenstra, Hess, Berendsen, J. Comp. Chem. 20, 786 (1999)

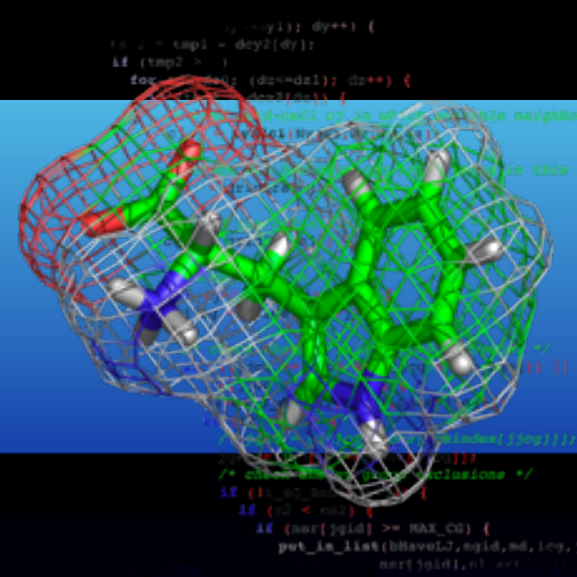
GROMACS Approaches



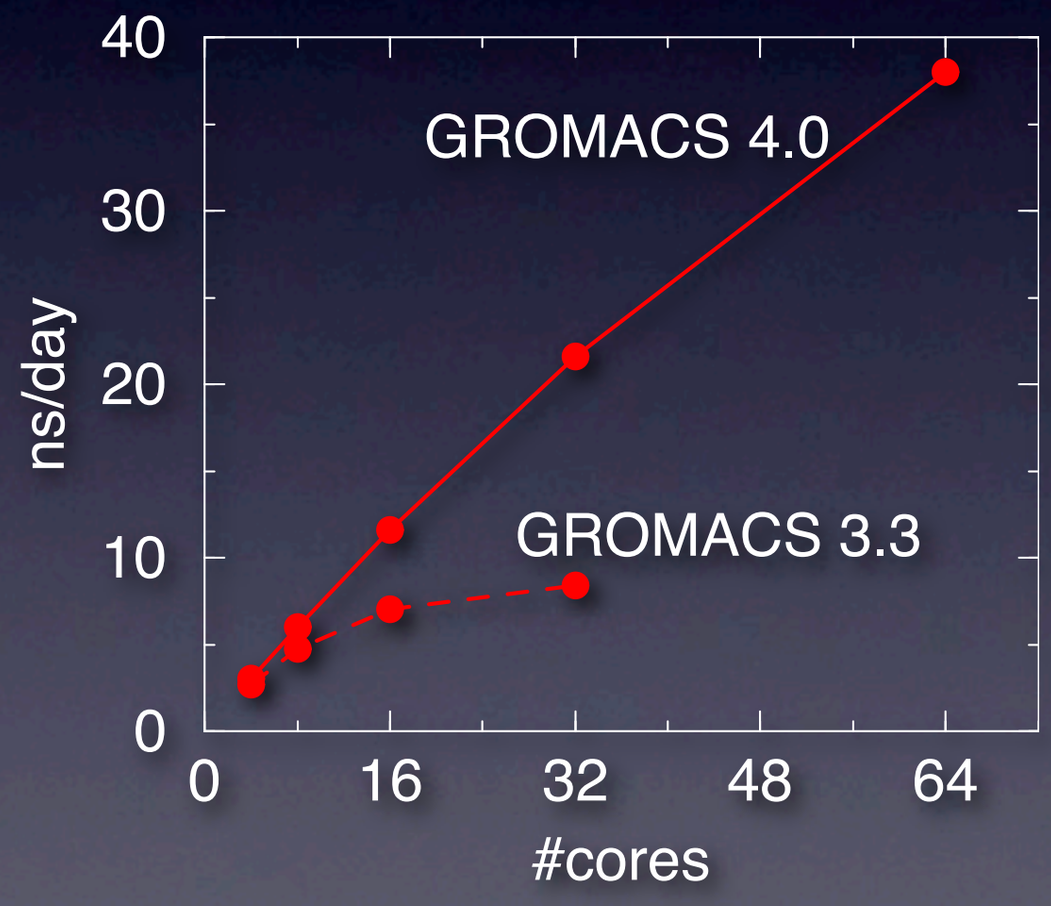
- Algorithmic optimization:
  - No virial in nonbonded kernels
  - Single precision by default (cache, BW usage)
  - Tuning to avoid conditional statements such as PBC checks
  - Triclinic cells everywhere: can save 15-20% on system size
- Optimized 1/sqrt(x)
  - Used ~150,000,000 times/sec
  - Handcoded asm for ia32, x86-64, ia64, Altivec, VMX, BlueGene (SIMD)



GROMACS 4.0

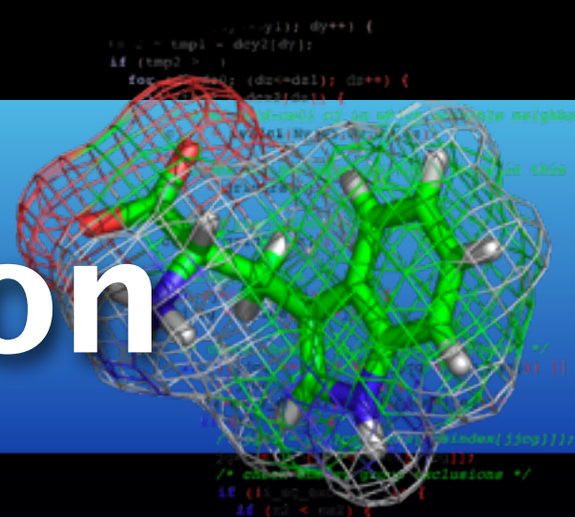


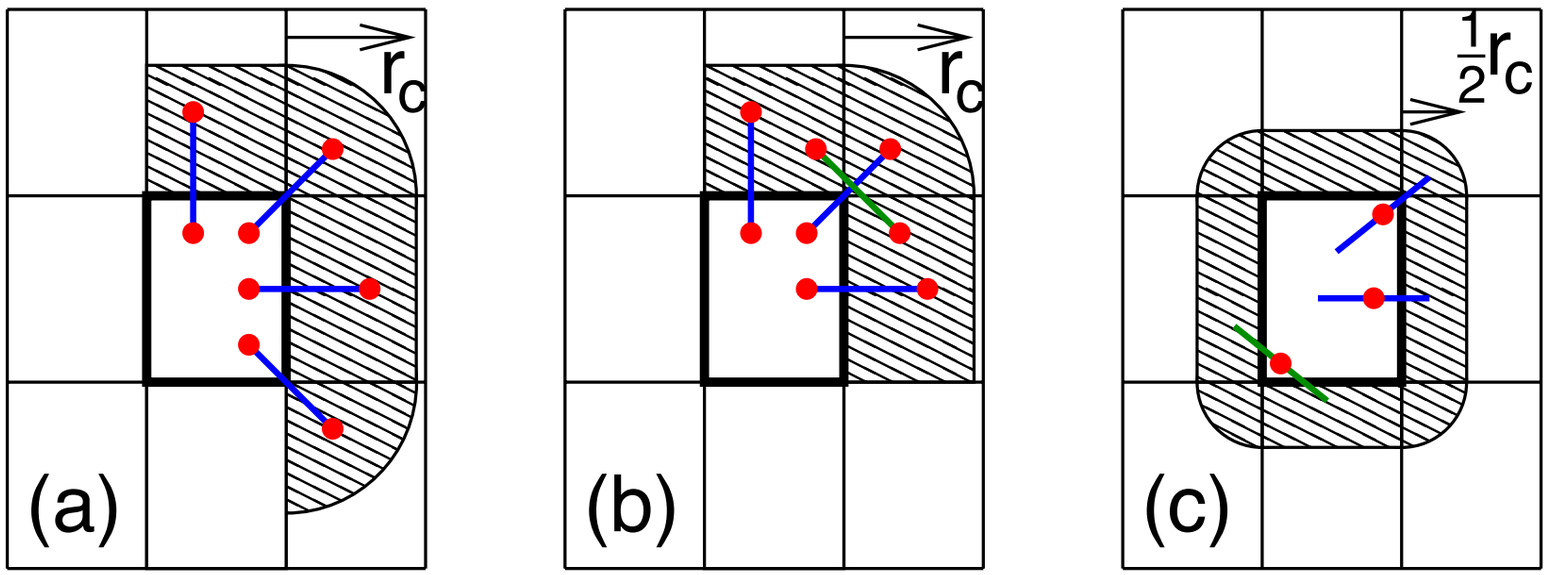
- GROMACS 4.0 released October 2008



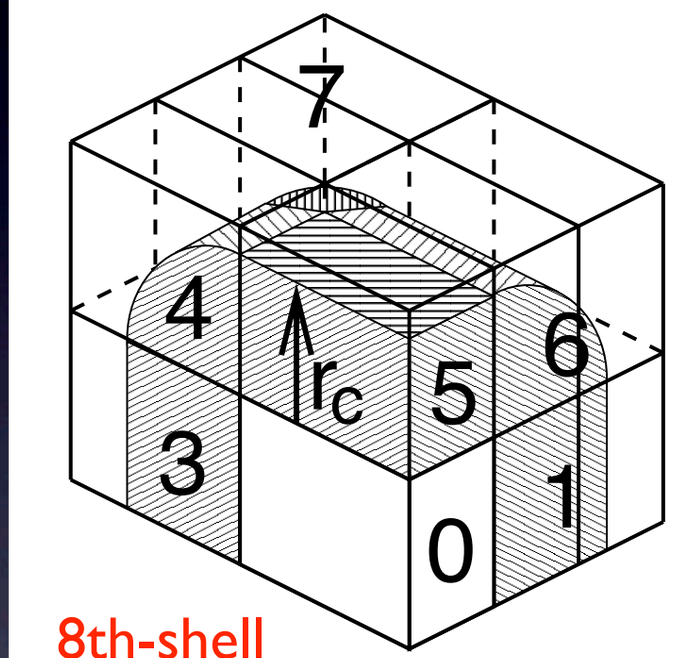
Hess, Kutzner, Van der Spoel, Lindahl; JCTC 4, 435 (2008)

8th-shell decomposition





half-shell 8th-shell midpoint



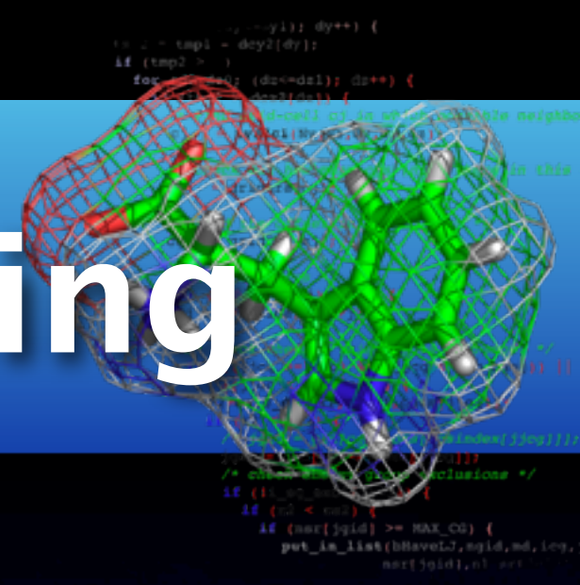
8th-shell

8th-shell 1/4 of the communication of half-shell

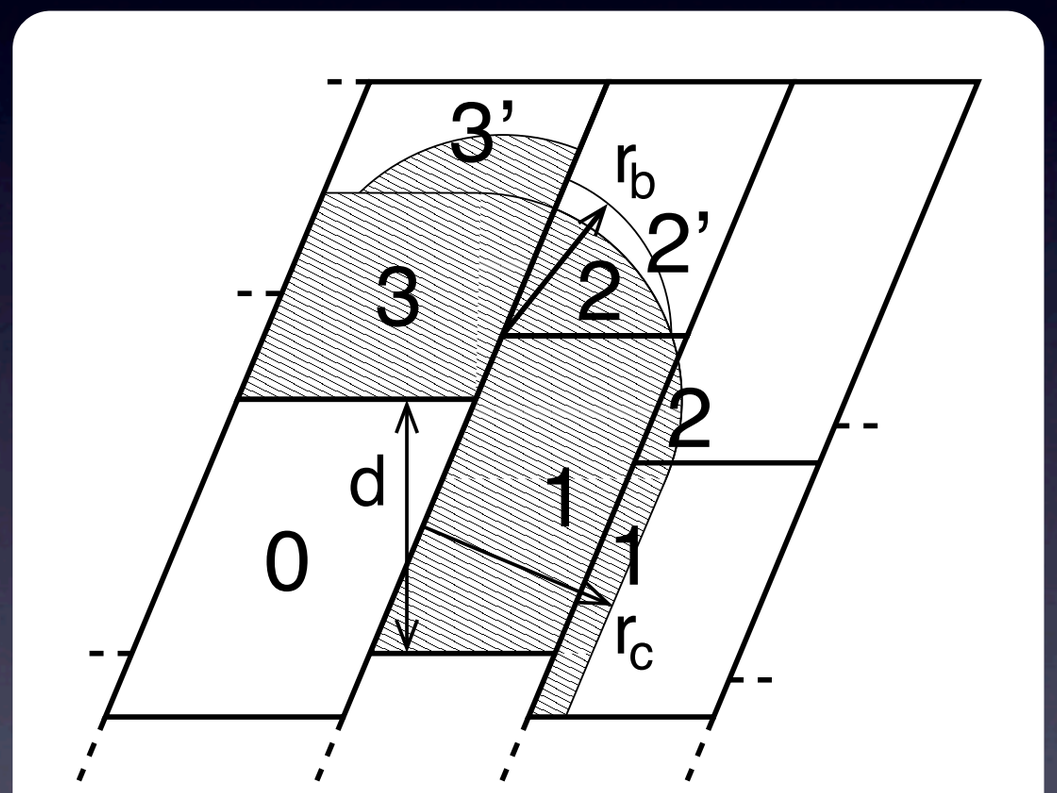
8th shell: Liem, Brown, Clarke; Comput. Phys. Commun. 67(2), 261 (1991)

Midpoint: Bowers, Dror, Shaw, JCP 124, 184109 (2006)

Dynamic load balancing

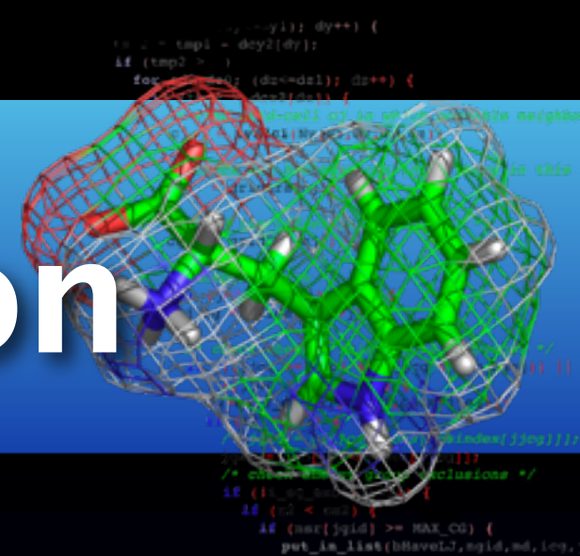


- Causes of load imbalance:
  - Atom imhomogeneity
  - Inhomogeneous interaction cost
  - Statistical fluctuation
- Full, 3D dynamic load balancing required
- Hardware cycle counters

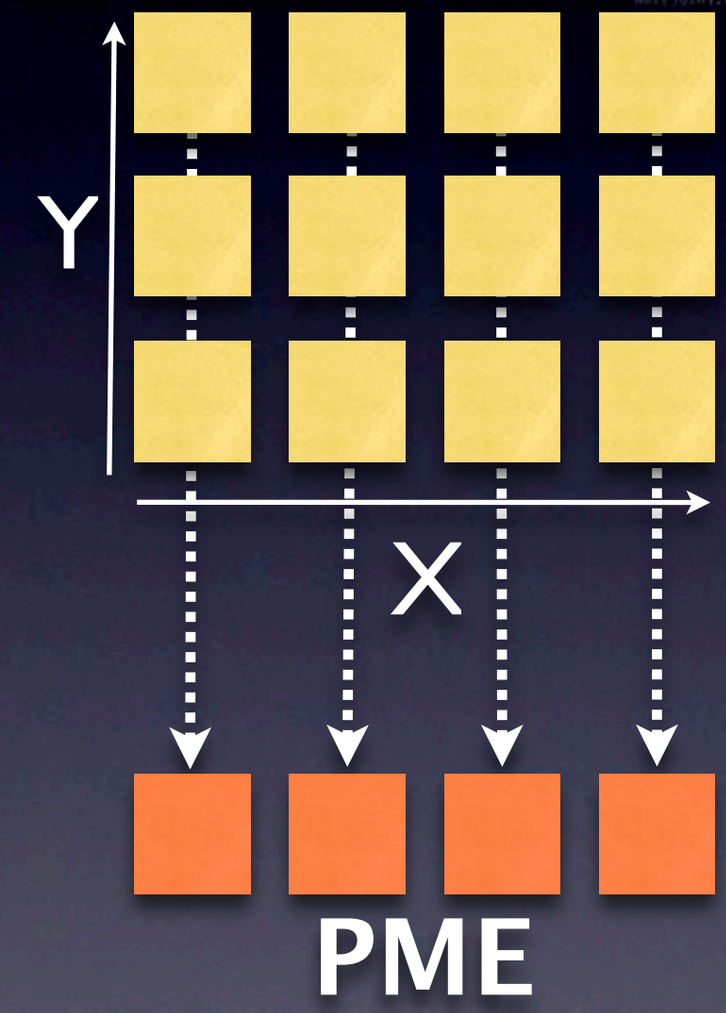


Triclinic, 2D example

MPMD force calculation

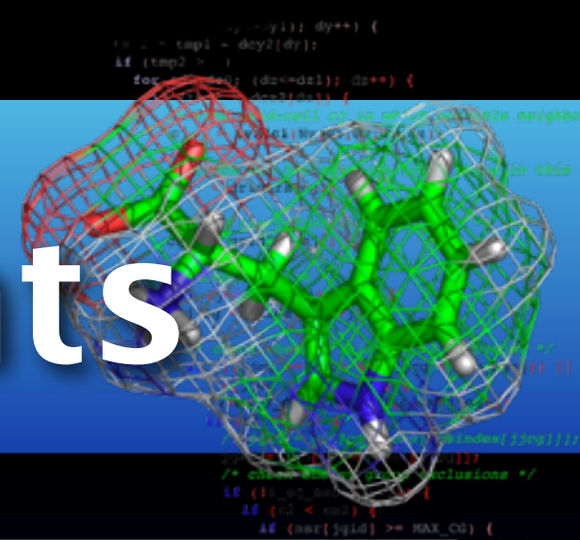


- PME = rapid Ewald summation
- Ubiquitous in simulations today
- Small 3D FFT's scale badly: All-to-All communication
- Real space & PME are independent
- Dedicate a subset of nodes to run a separate PME-only version of the program to improve scaling



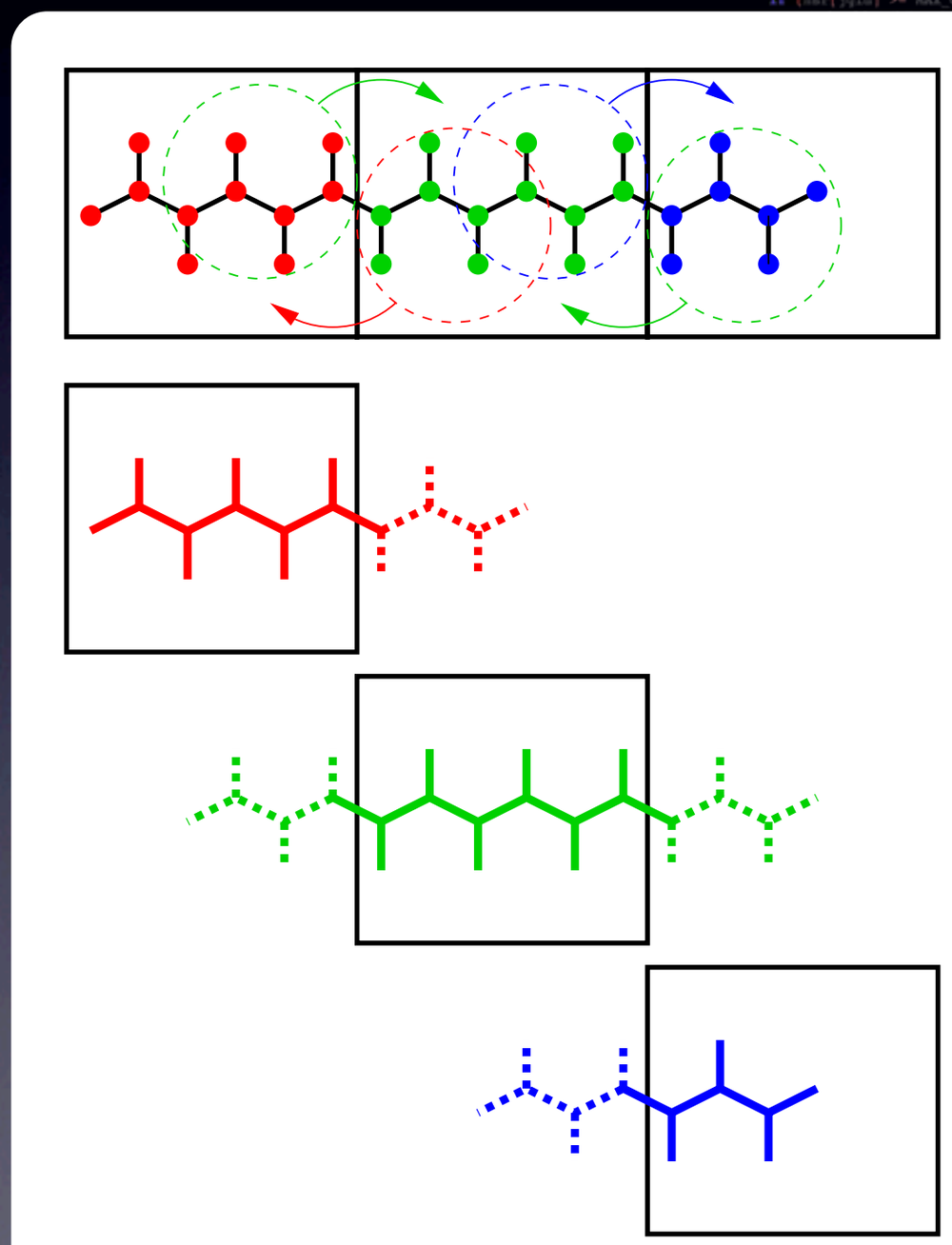
FFT over 4 cores
instead of 16 cores

Parallel constraints

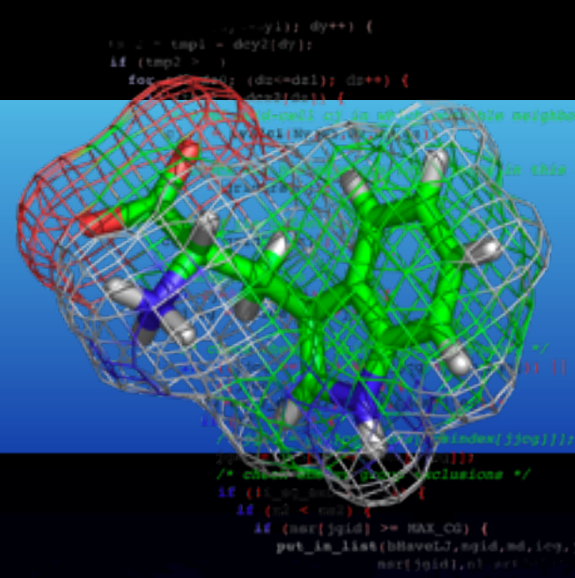


- Constraints required for 5 fs time steps
- Parallel LINCS algorithm: P-LINCS
- LINCS has a (short) finite interaction range
- First efficient parallel constraint algorithm

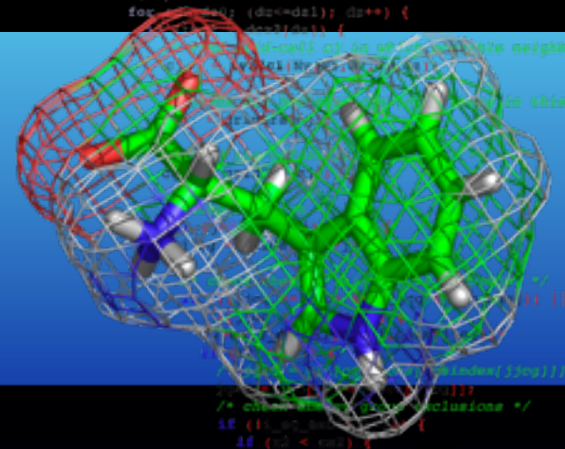
Hess; JCTC 4, 116 (2008)

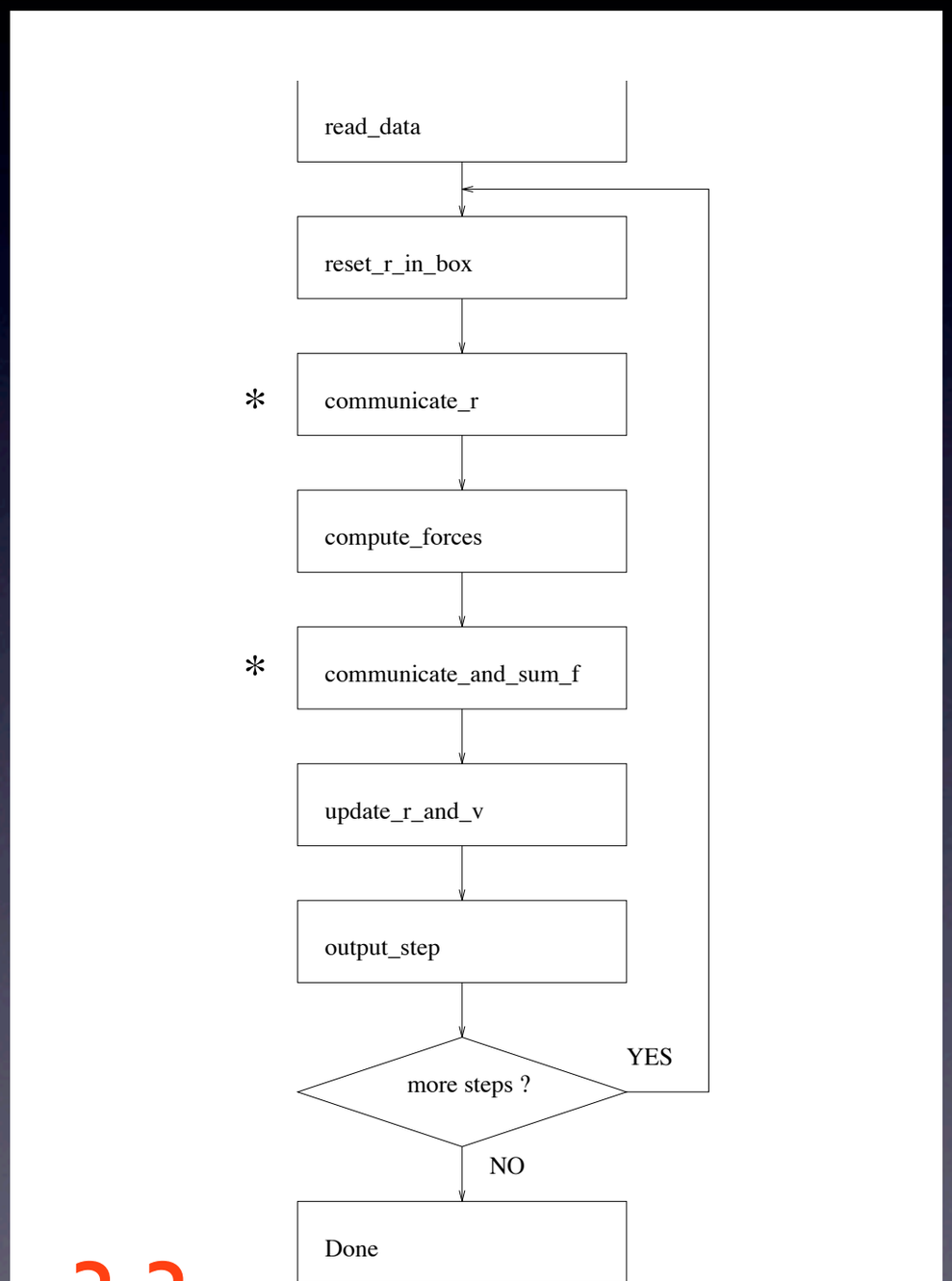


Flowcharts



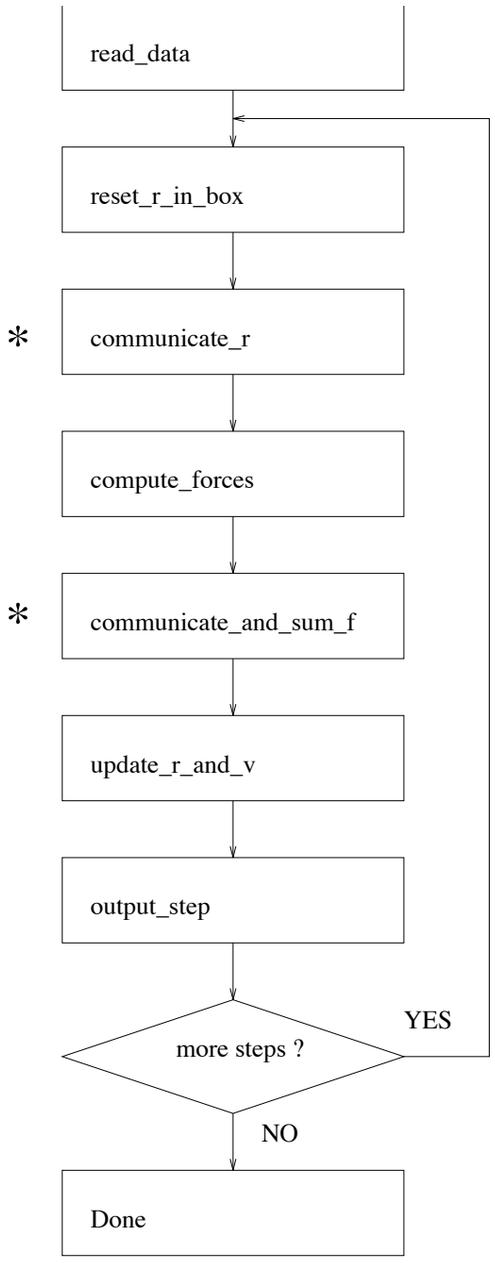
Flowcharts



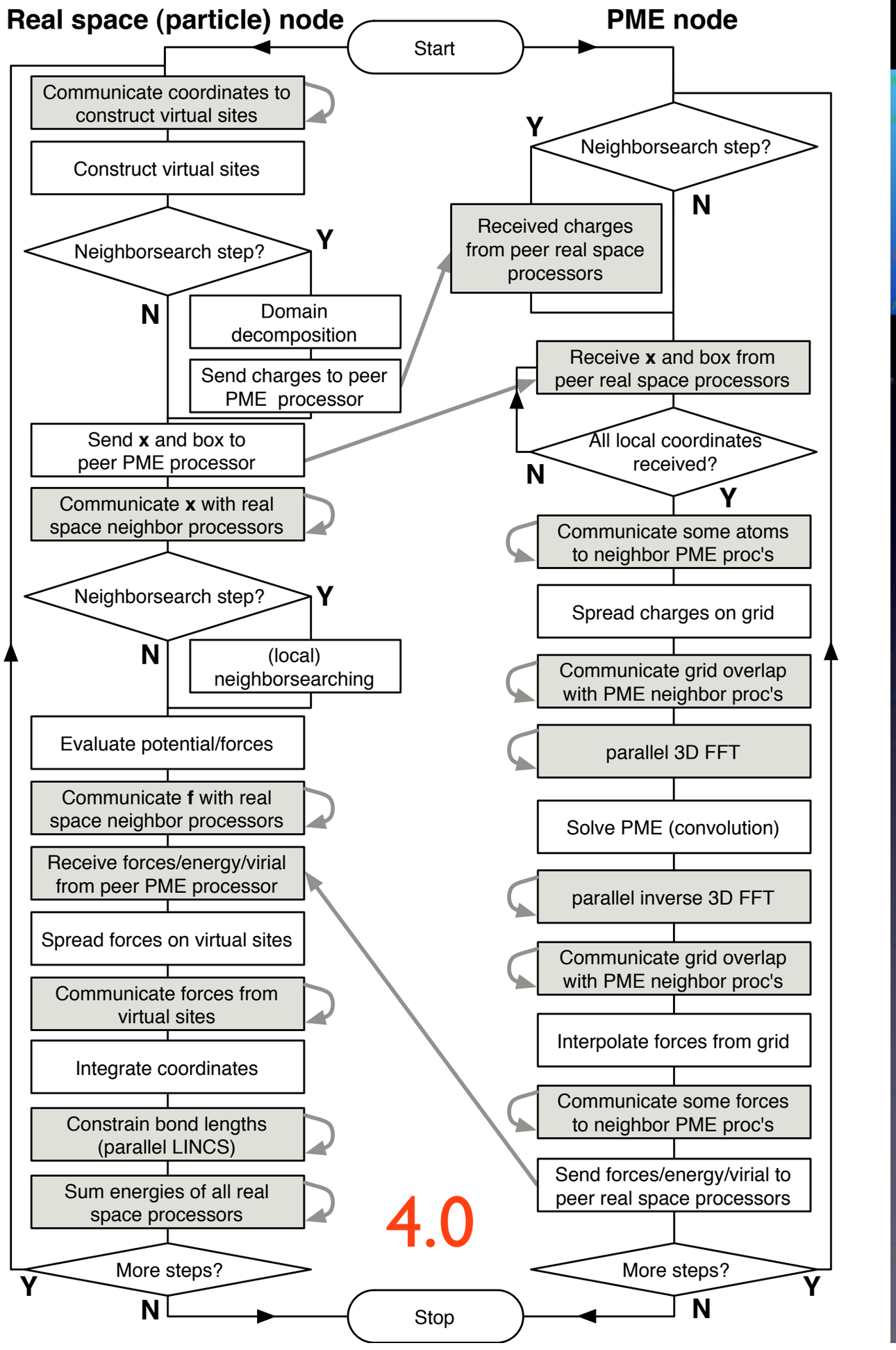


3.3

Flowcharts

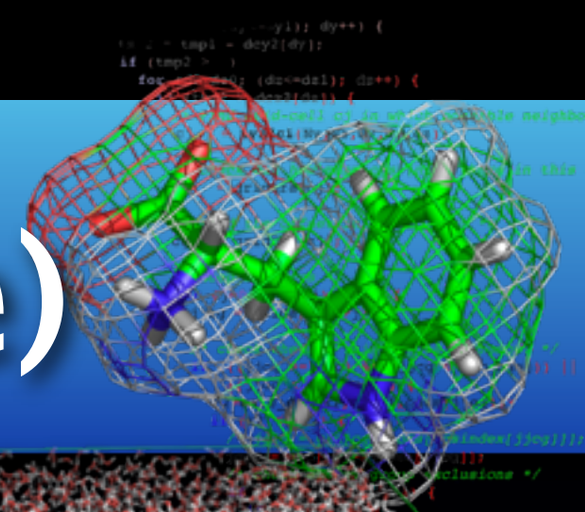


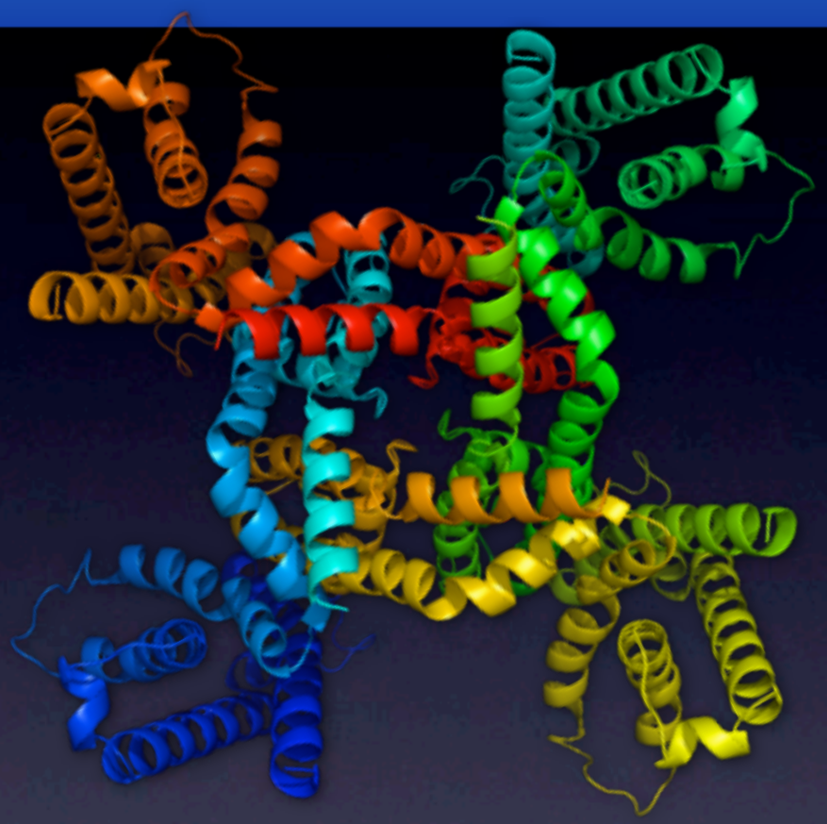
3.3

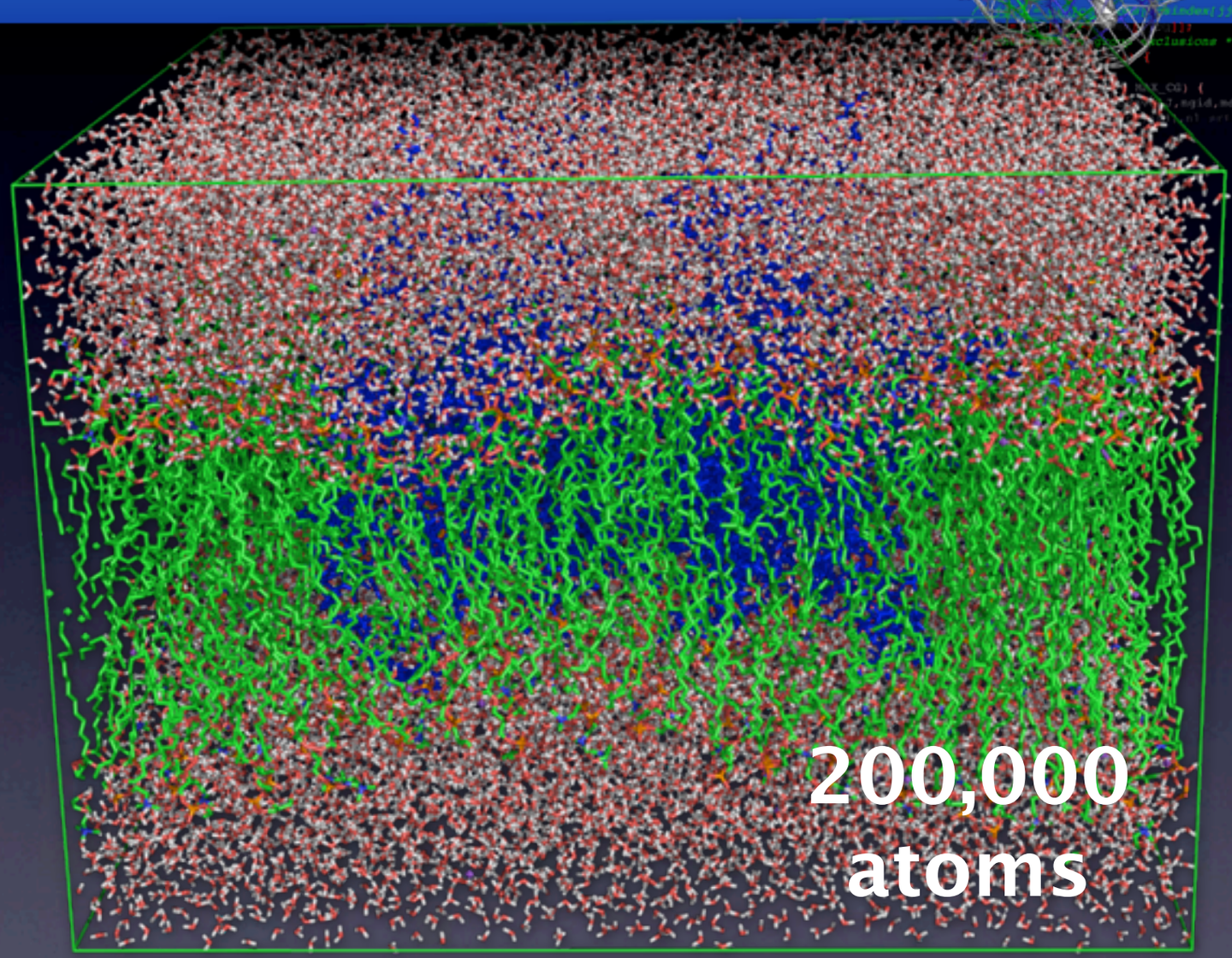


4.0

Performance (old slide)



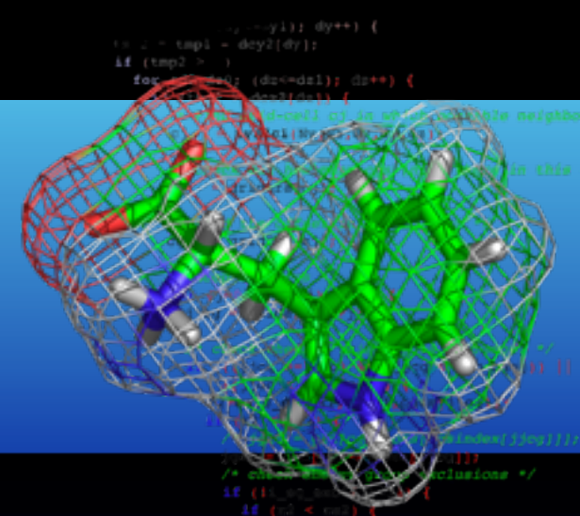




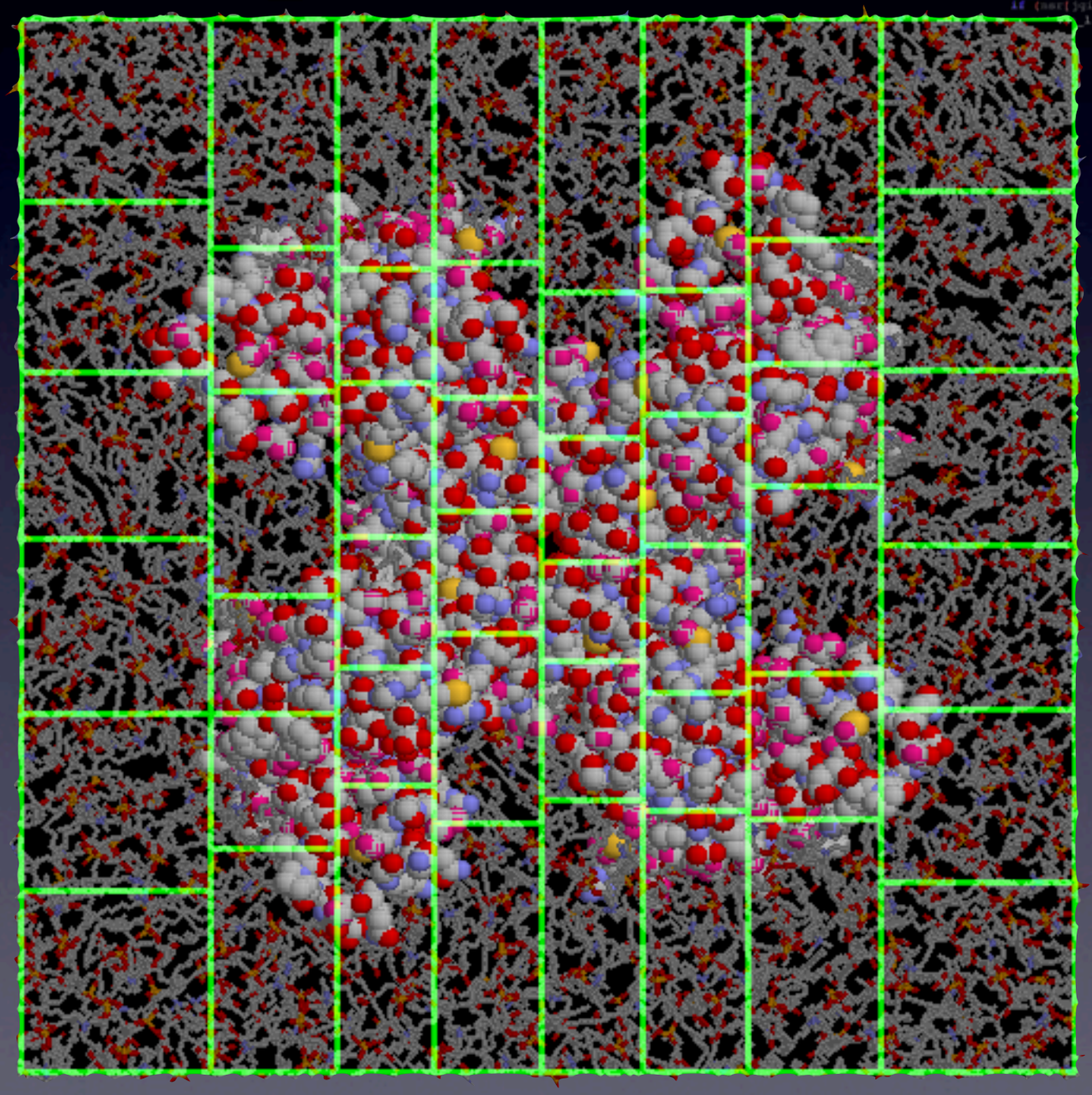
Cray XT4 @ CSC

1 µs in 3-4 weeks using 170 CPUs:
50x longer than previously possible

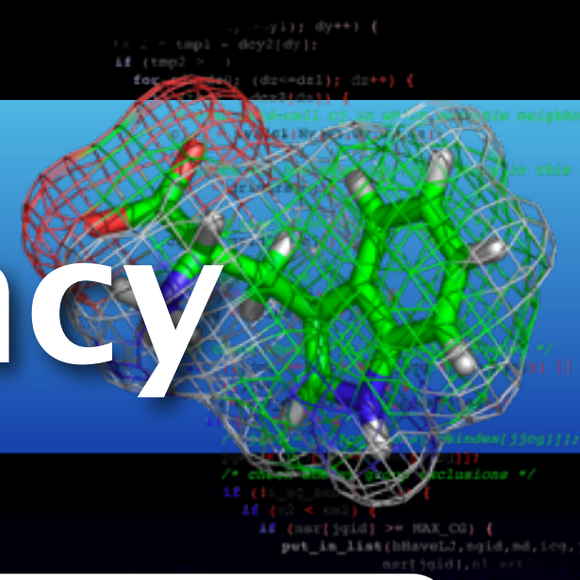
DLB in action



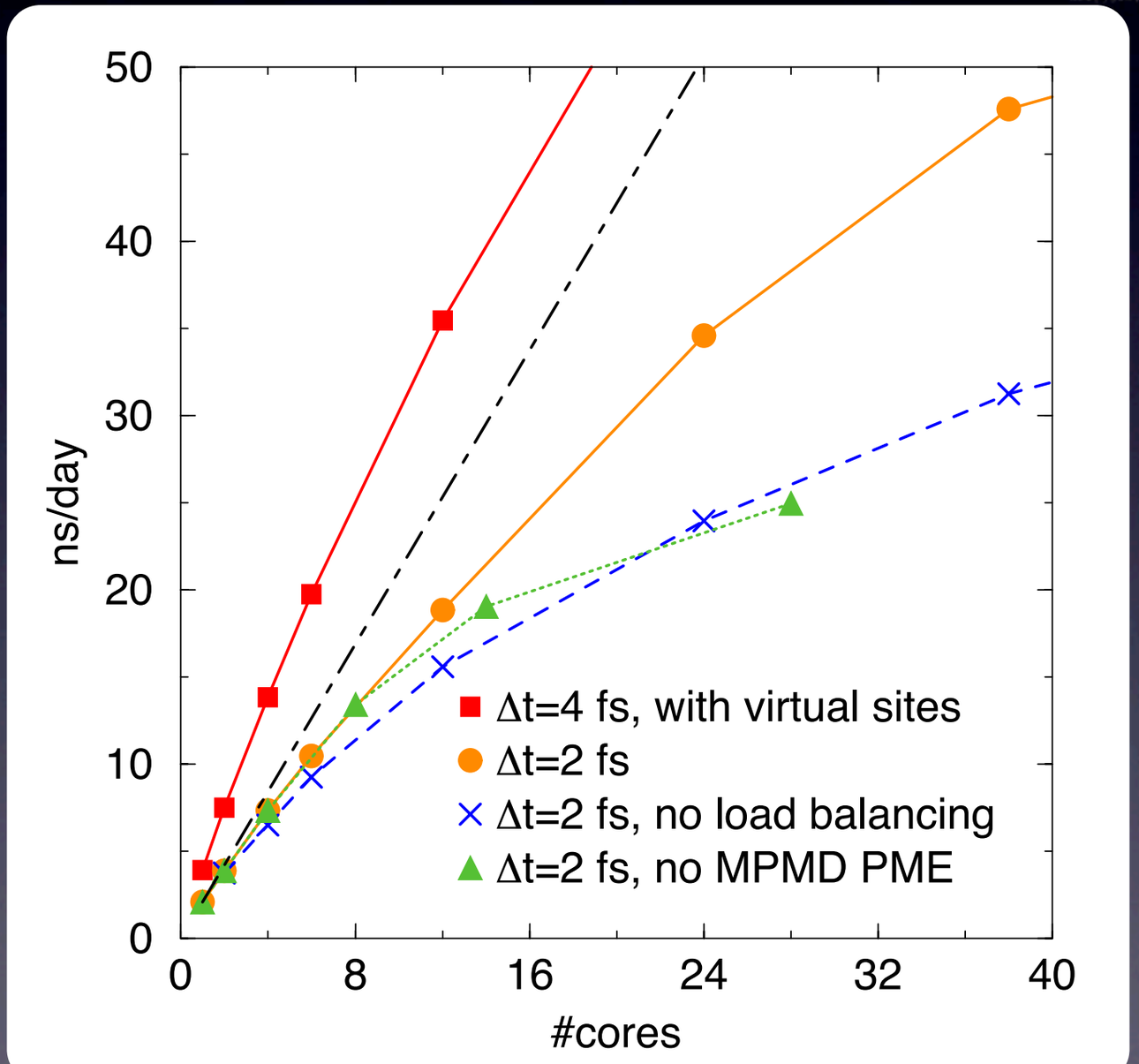
- 8x6=48 PP cores
- 16 PME cores
- protein: “slow”
- lipids: fast



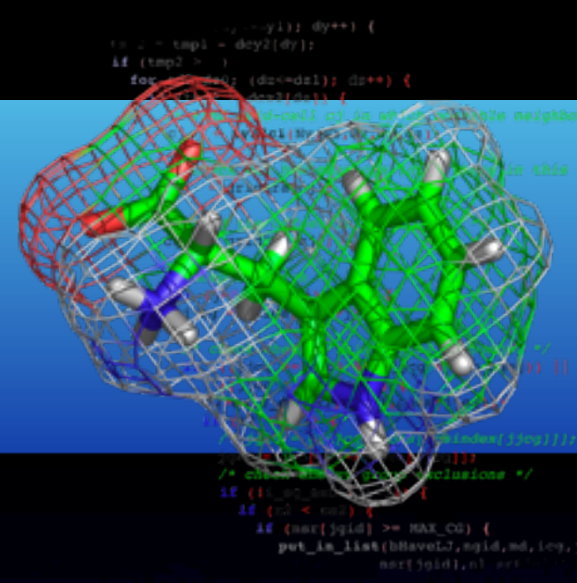
Algorithm efficiency



- Protein system:
  - T4-lysozyme
  - H_2O, Cl^-
  - 24199 atoms
  - 1 nm cut-off
  - PME

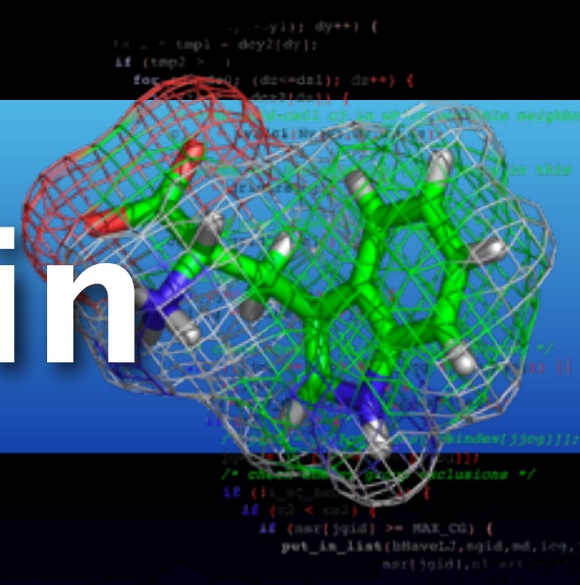


Scaling limits



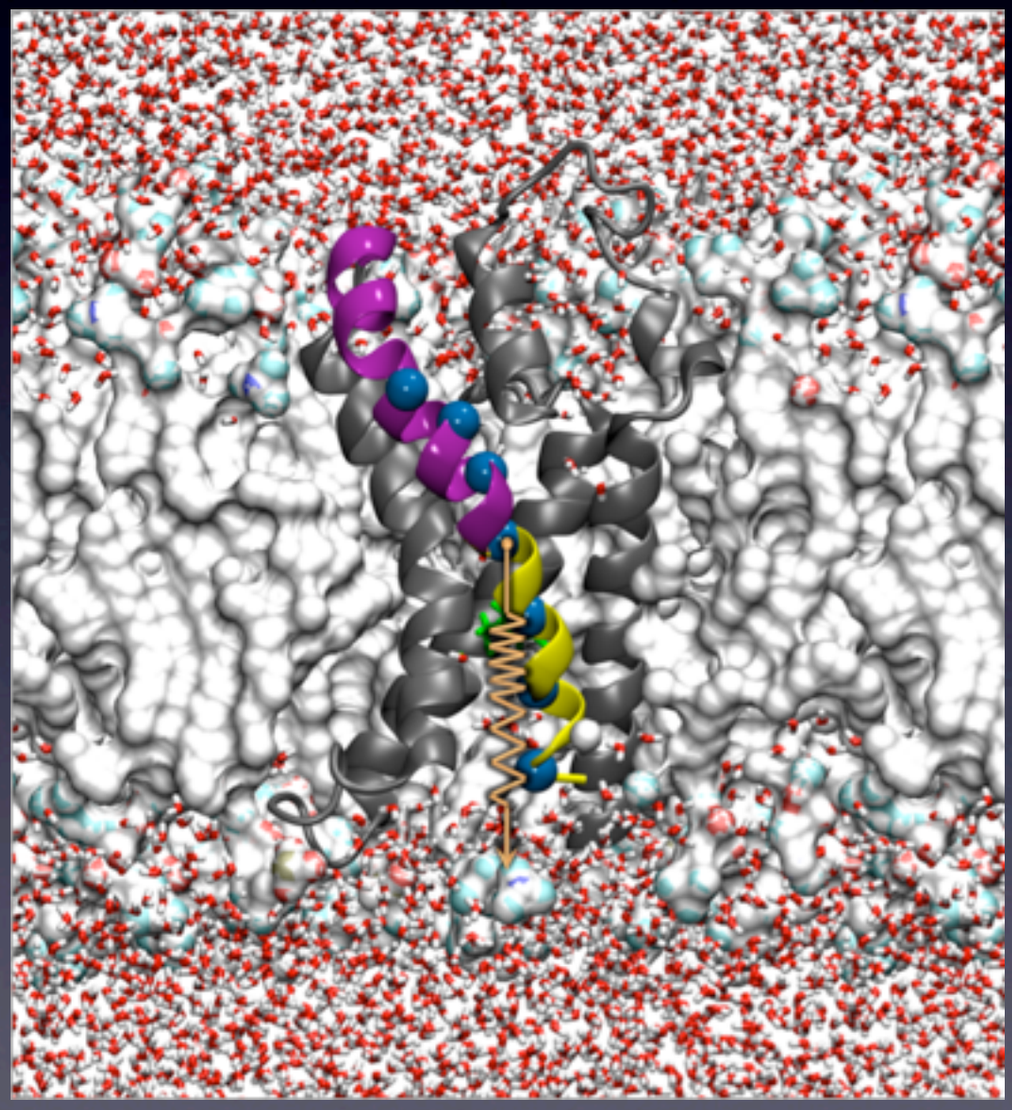
- Without Particle-Mesh-Ewald
  - Weak scaling: no limit
  - Strong scaling: ~300 atoms per core
- With Particle-Mesh-Ewald
  - “1D”-PME, 100’s of cores
  - GROMACS 4.1: “2D”-PME, 1000’s of cores
  - GROMACS 5: Multi-grid?

Membrane protein

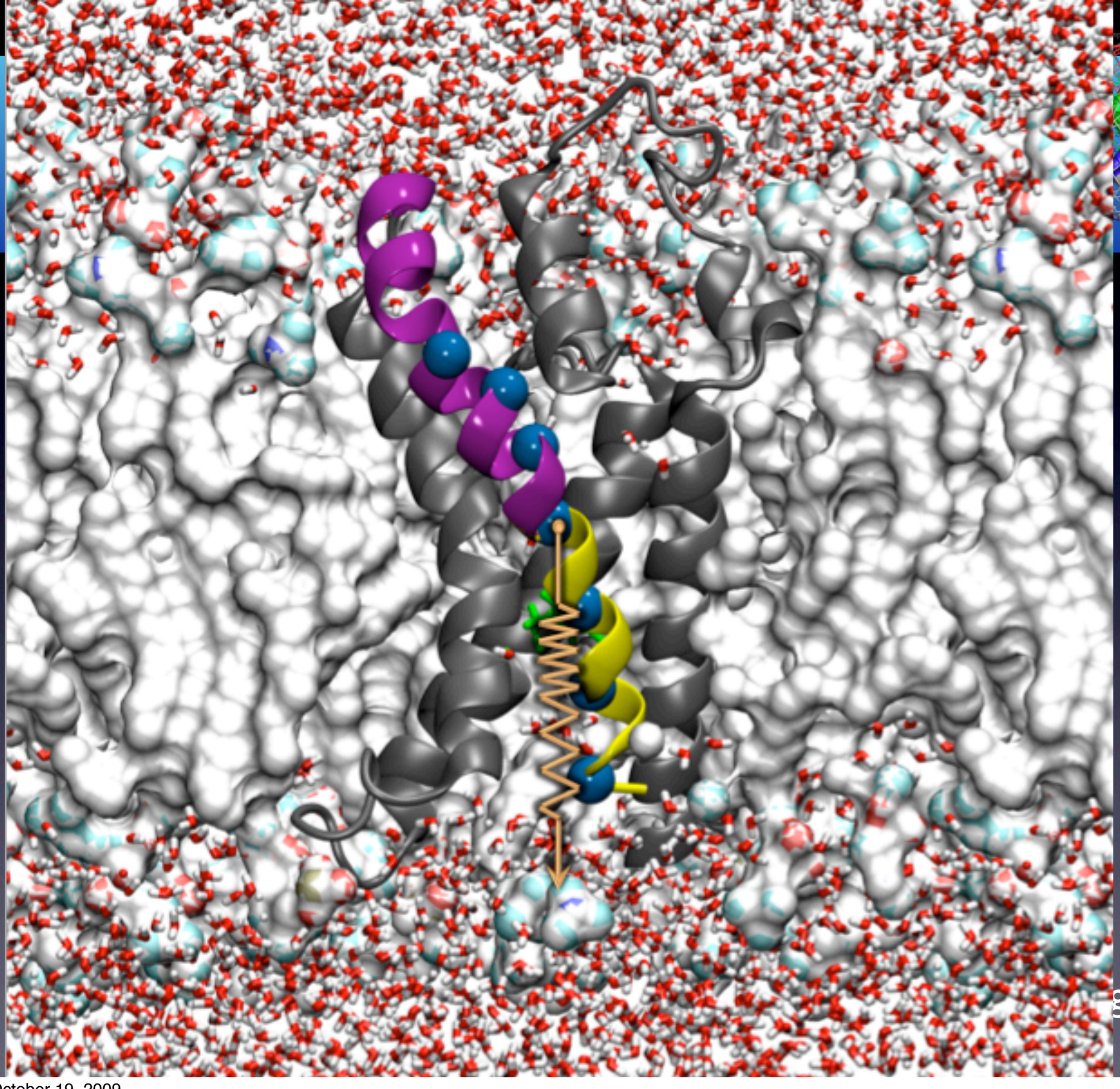


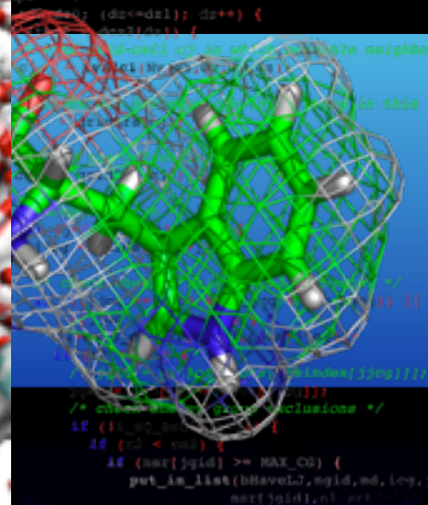
- Kv1.2/2.1 voltage-gated ion channel
- Open & closed state
- Contains a voltage sensor
- How does it work?

- Problem: transition is slow
  - Energy barrier: ~10 k_BT
- Long simulations: ms, μs



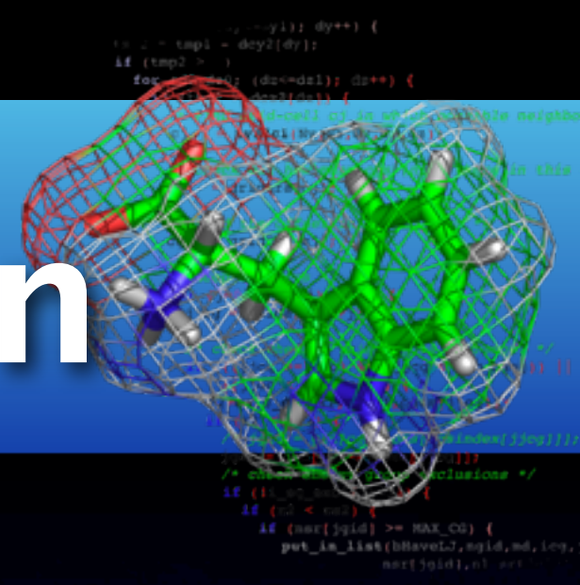
Sophie Schwaiger



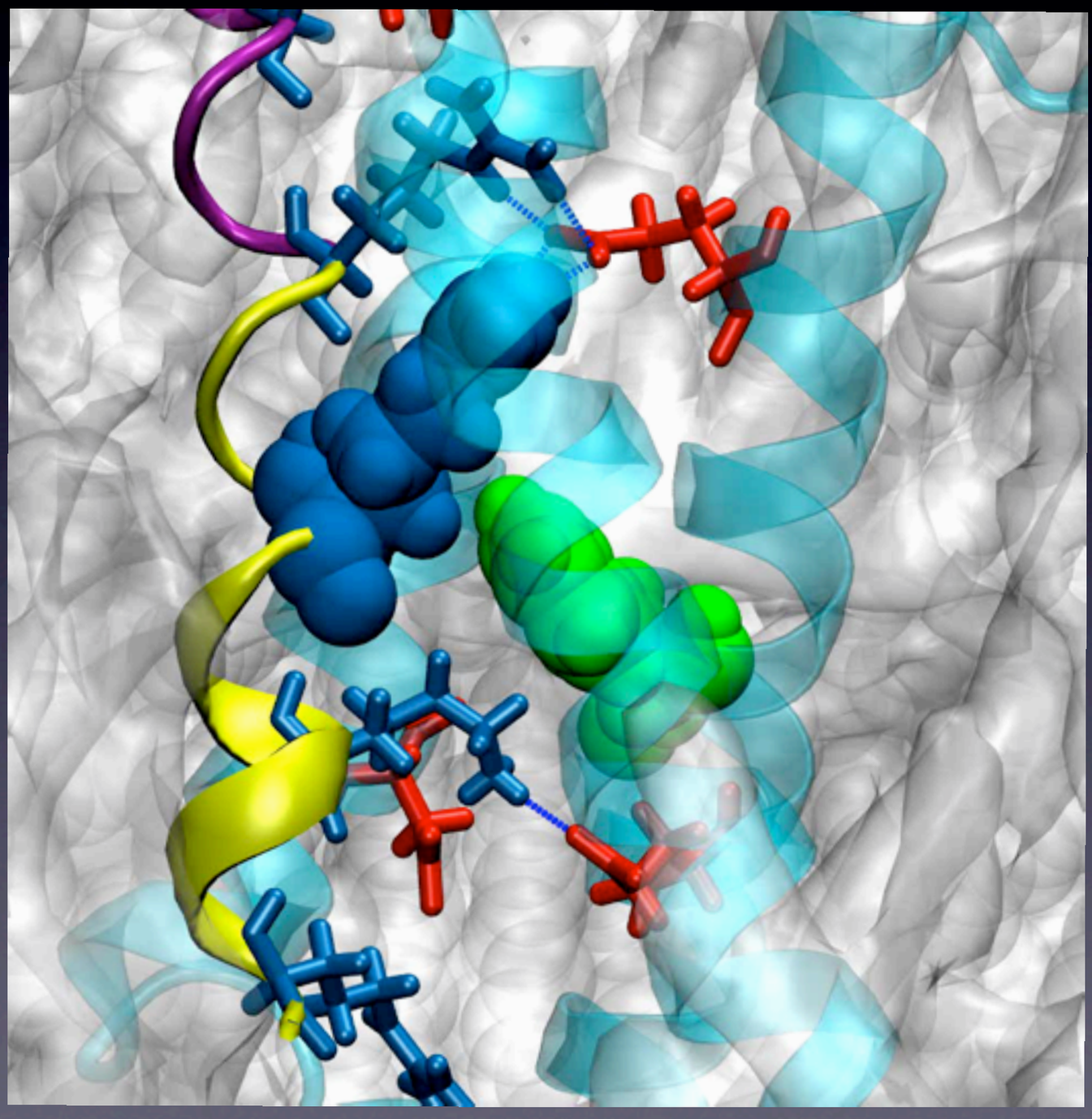


ger

0.5 µs simulation

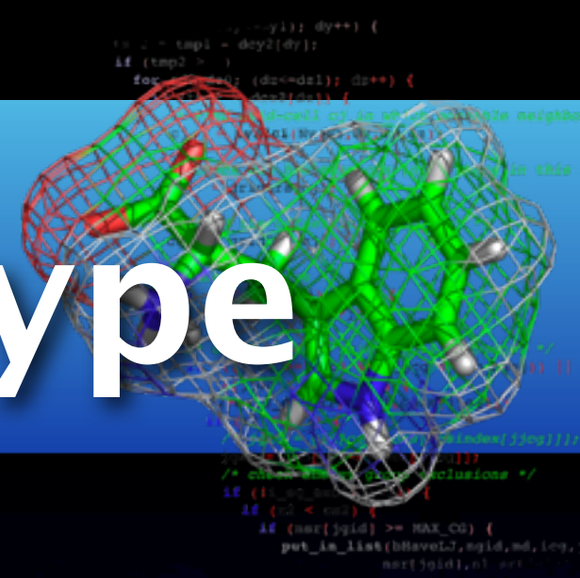


0.5 µs simulation

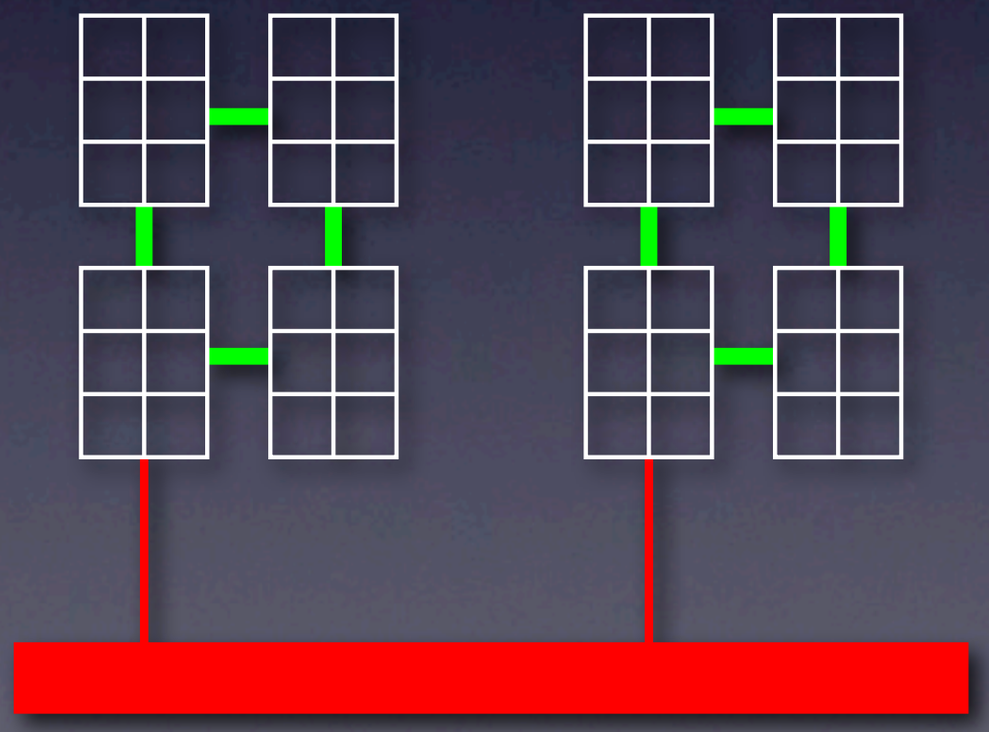


Recent GROMACS & hardware developments

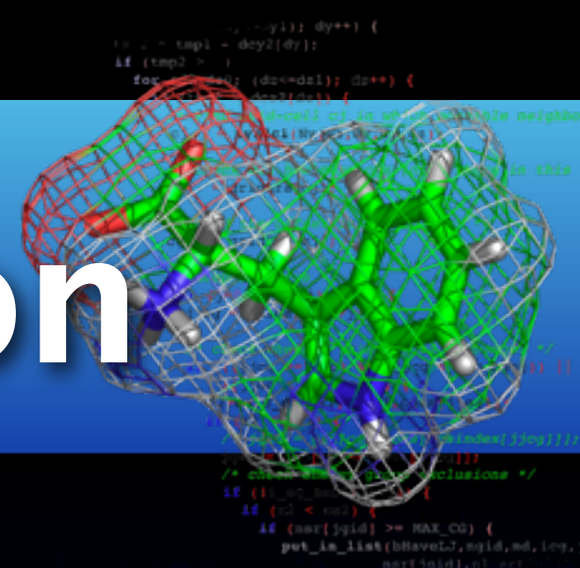
PDC PRACE prototype



- PRACE test machines with 4x6 AMD cores
- 24 core nodes connected with Infiniband
- Issue:
  24 cores share a network connection



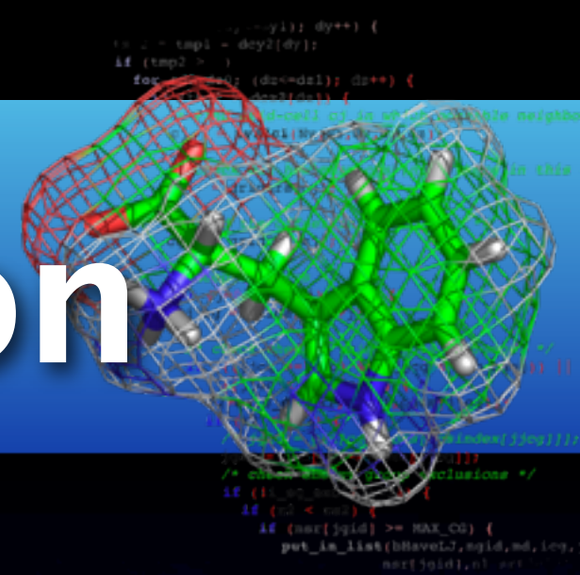
Global summation



- Most thermo/barostats need global summation
- But this can be relatively VERY expensive
- Avoid when possible!

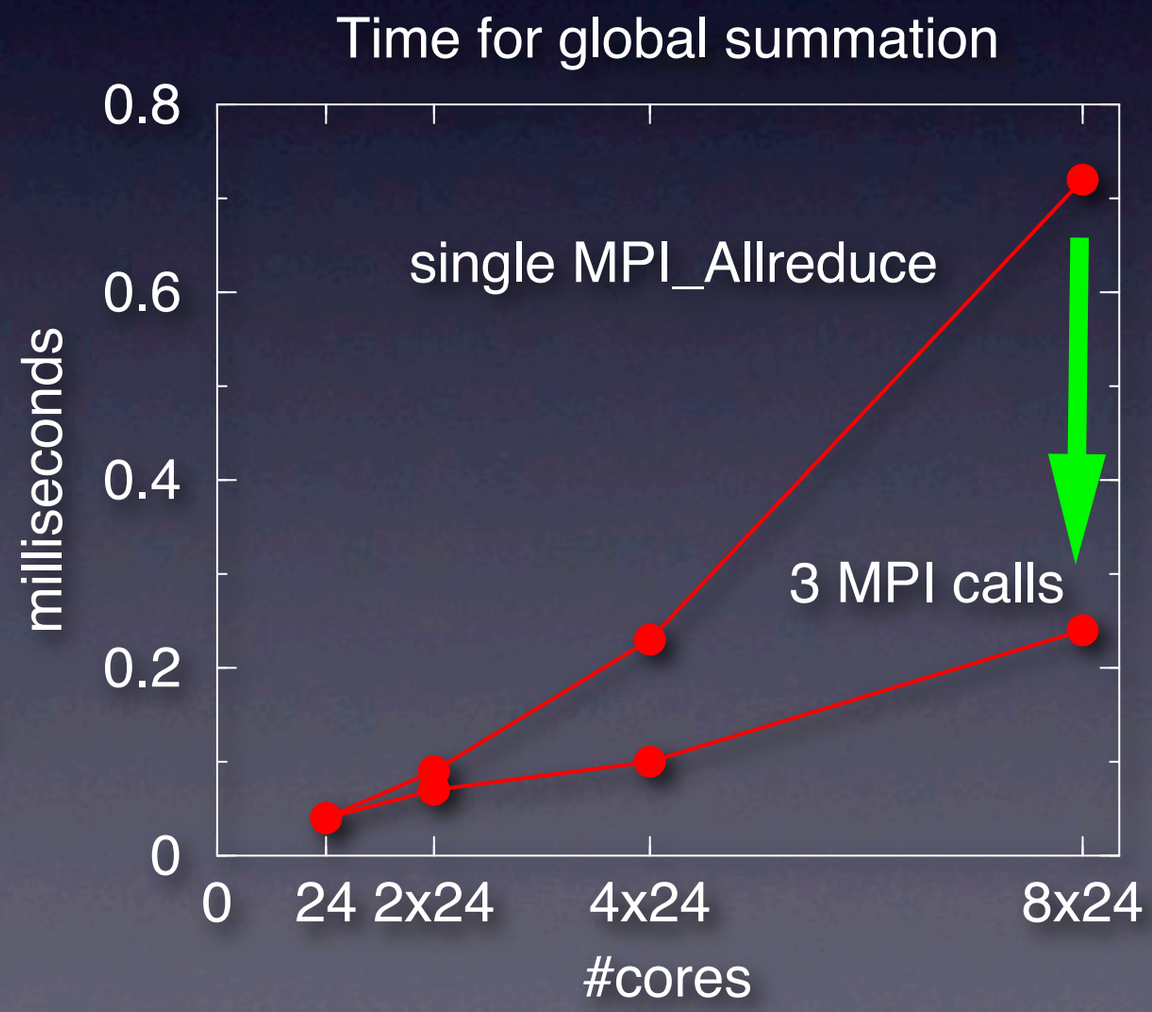
- GROMACS 3-step summation procedure:
  - MPI_Reduce, 24 cores
  - MPI_Allreduce, N nodes
  - MPI_Bcast, 24 cores

Global summation

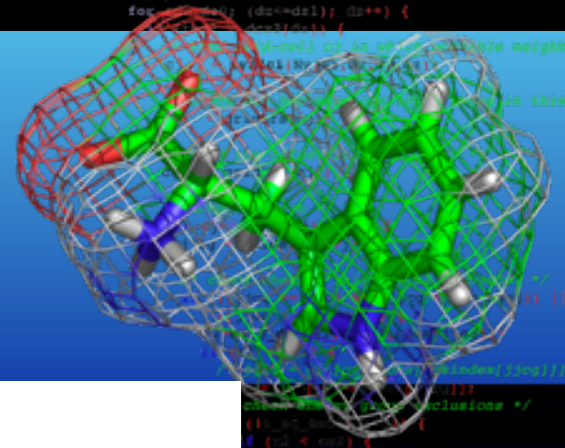


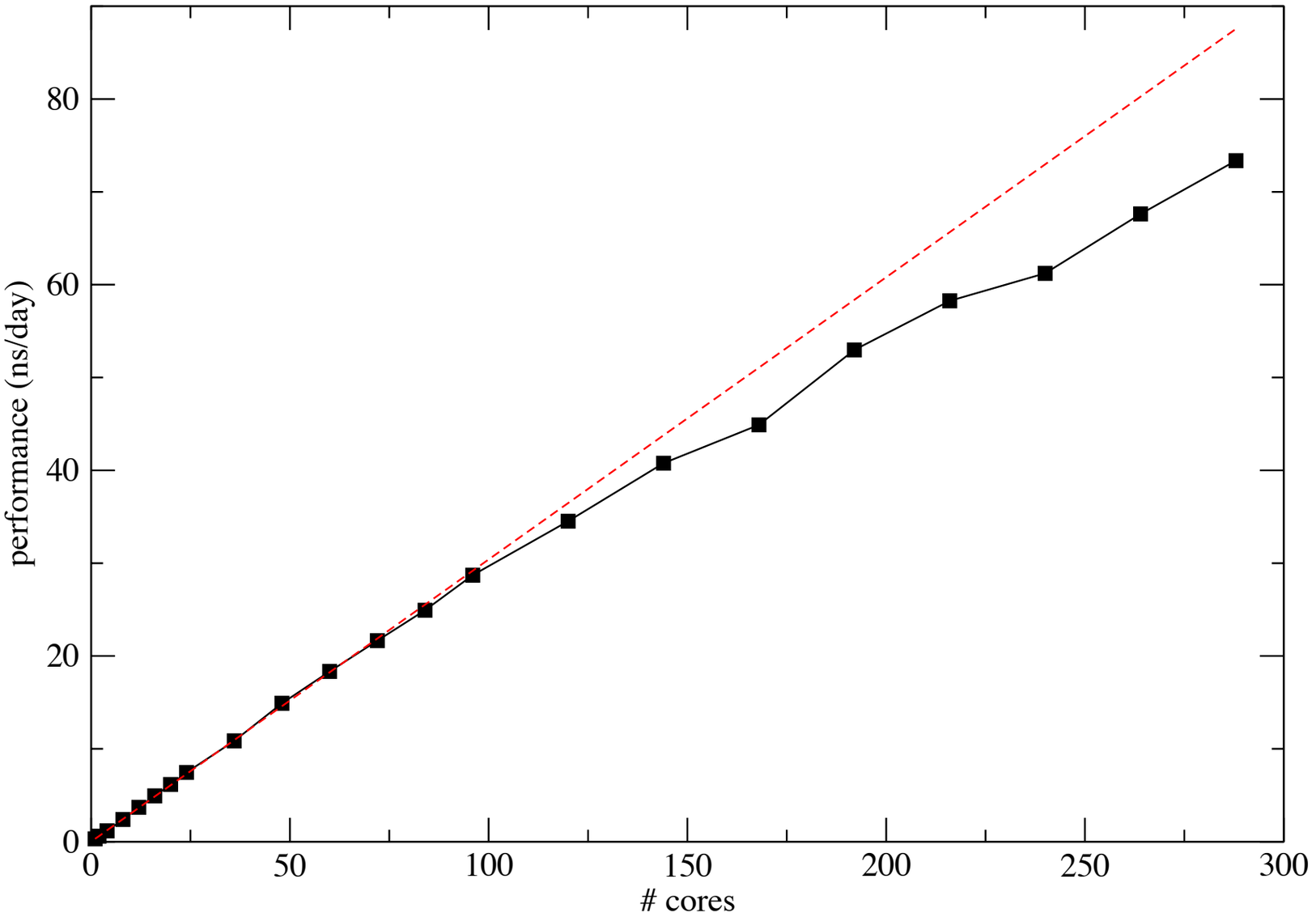
- Most thermo/barostats need global summation
- But this can be relatively VERY expensive
- Avoid when possible!

- GROMACS 3-step summation procedure:
  - MPI_Reduce, 24 cores
  - MPI_Allreduce, N nodes
  - MPI_Bcast, 24 cores

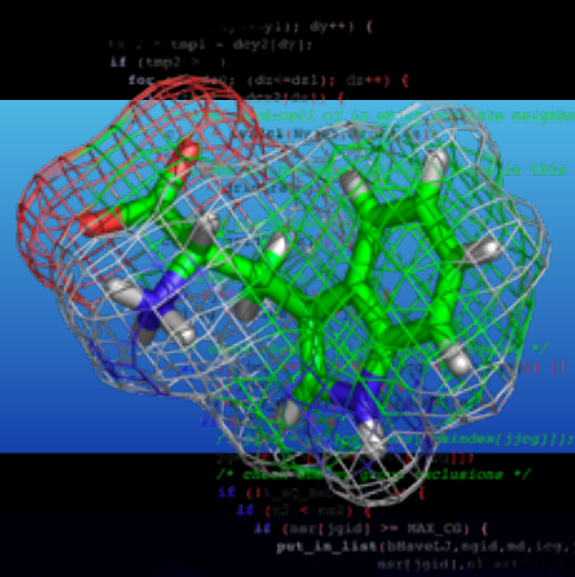


PRACE scaling

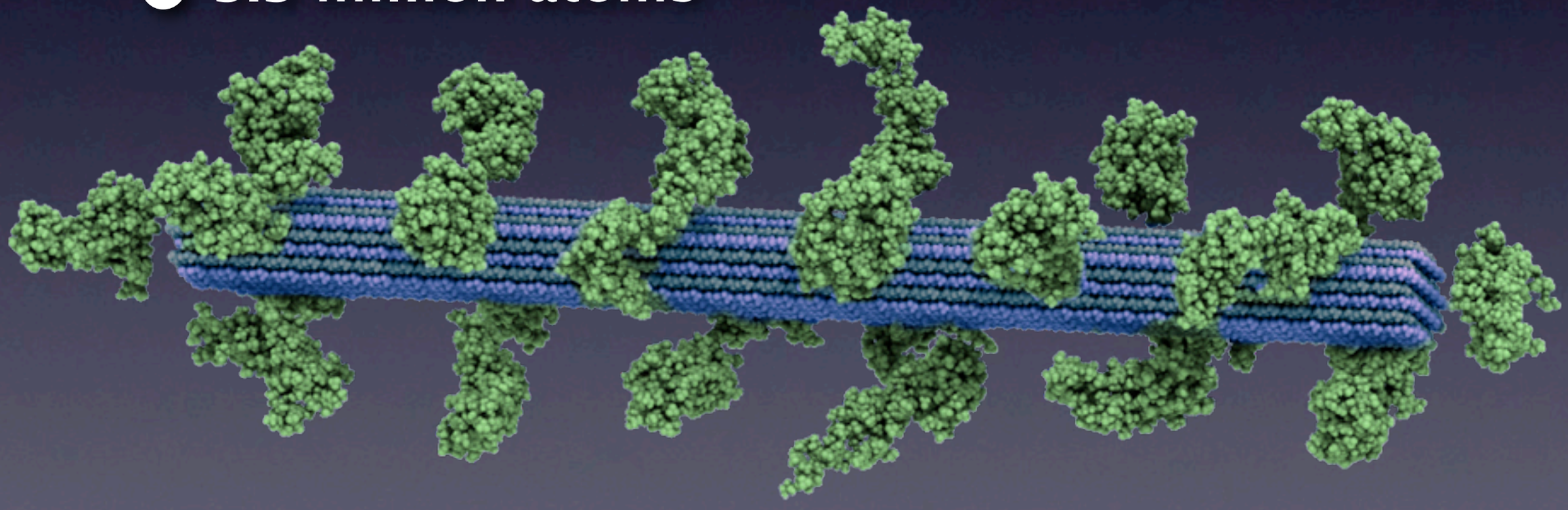




Multi-million atom biological system

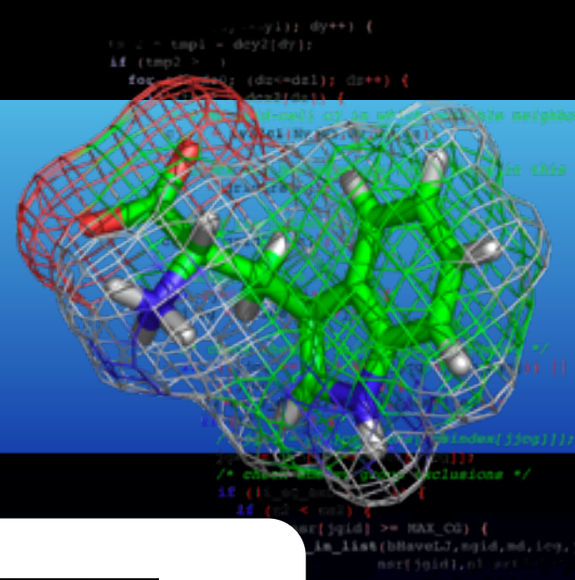


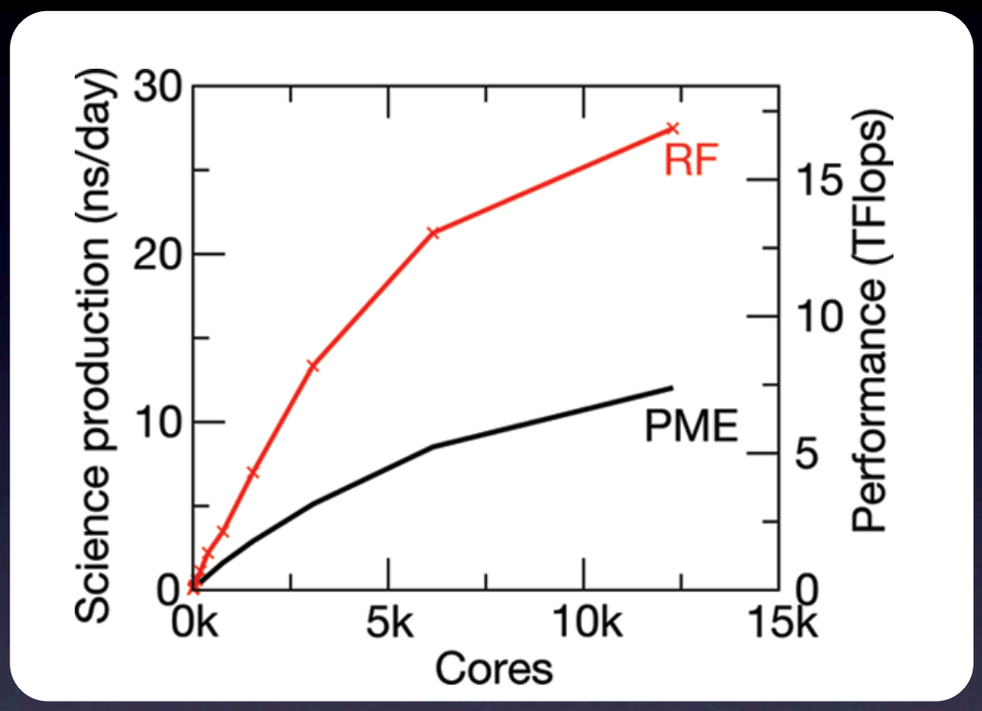
- Cellulose, H_2O, lignocellulosic biomass (biofuel)
- No charged groups -> reaction-field (no PME)
- 3.3 million atoms

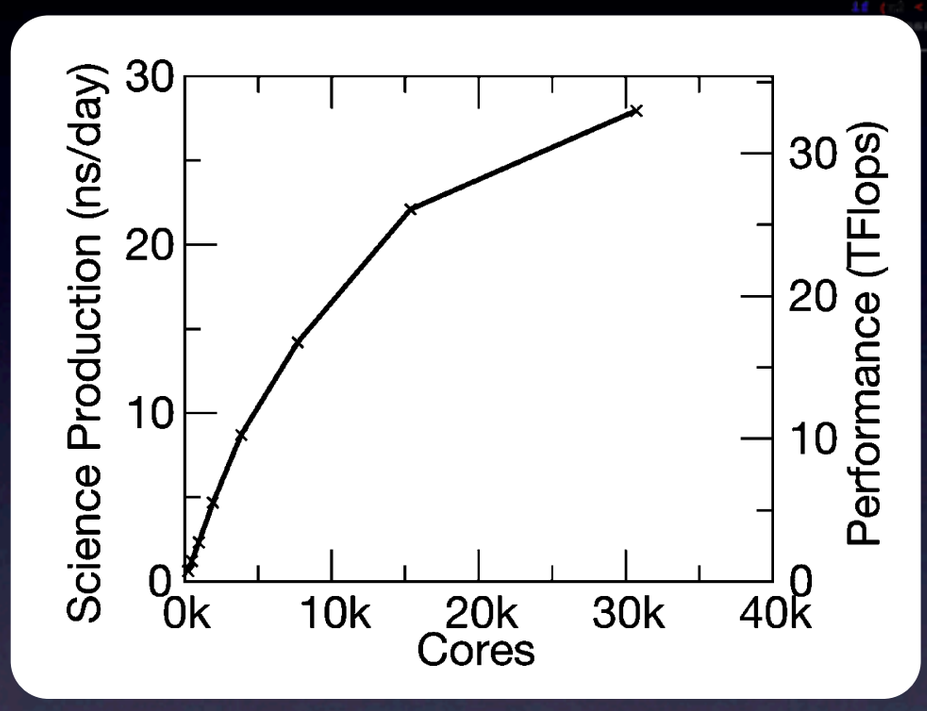


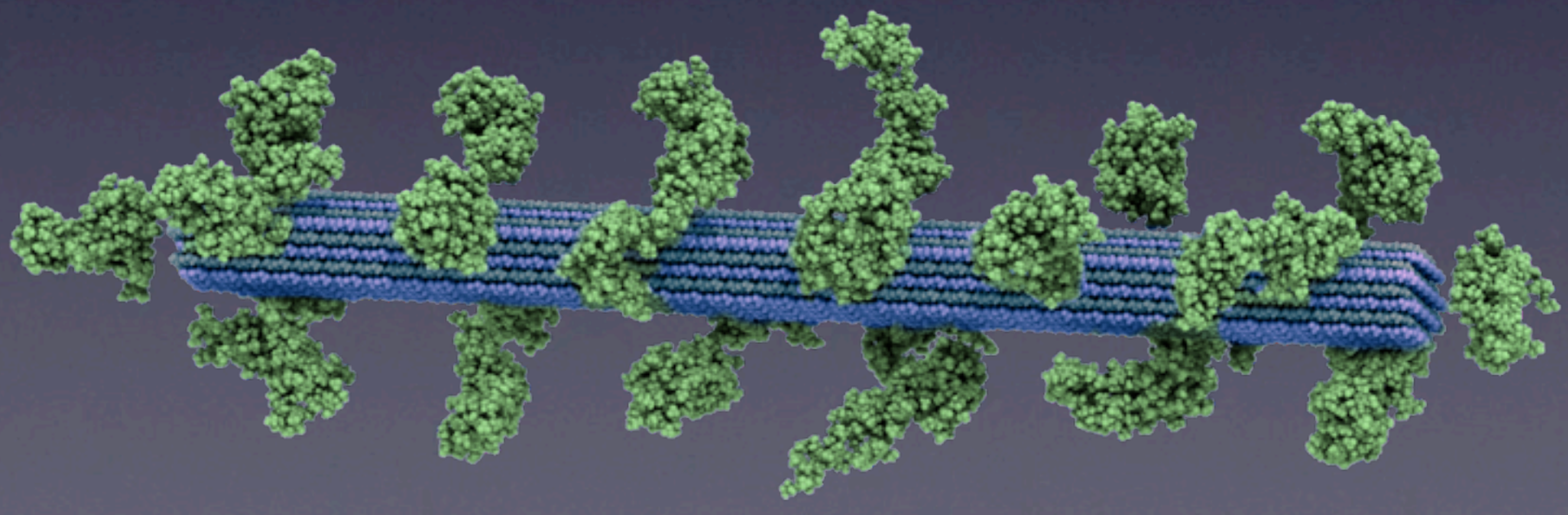
Schultz, Linder, Petridis, Smith; JCTC (2009)

10k scaling

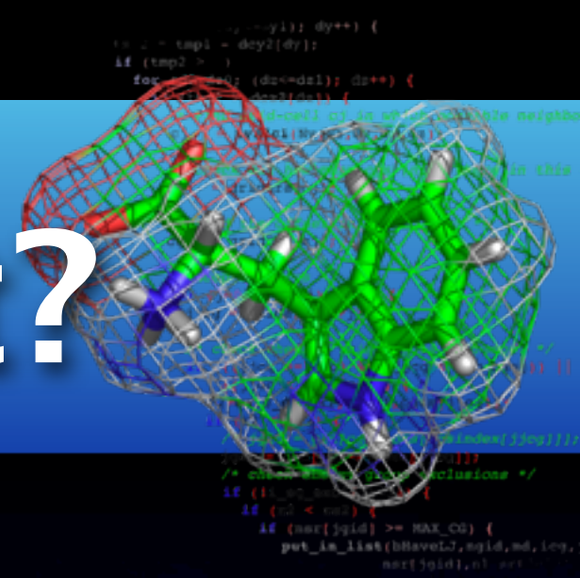



Science production (ns/day)
Performance (TFlops)
RF
PME
0
10
20
30
0
5
10
15
0k
5k
10k
15k
Cores


Science Production (ns/day)
Performance (TFlops)
0
10
20
30
0
10
20
30
0k
10k
20k
30k
40k
Cores



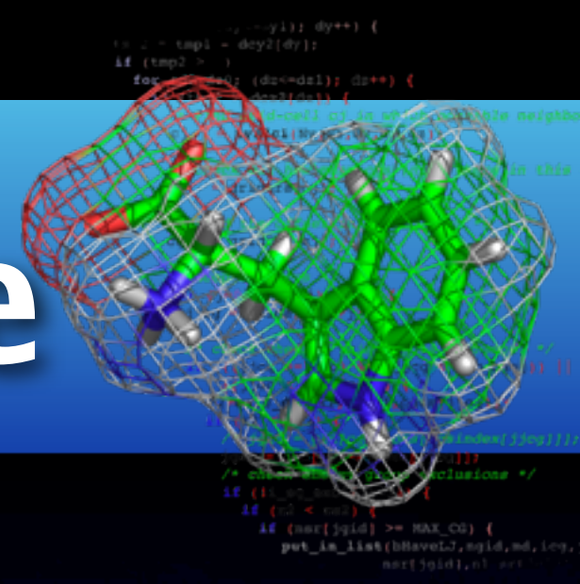
What is the limit?



100M atoms? 100k cores?

- GROMACS 4.0: linear scaling algorithms
- But still practical limitations:
  - File system access at start-up (fexist)
  - Data distribution at start-up
  - Still some O(#atoms) memory allocation

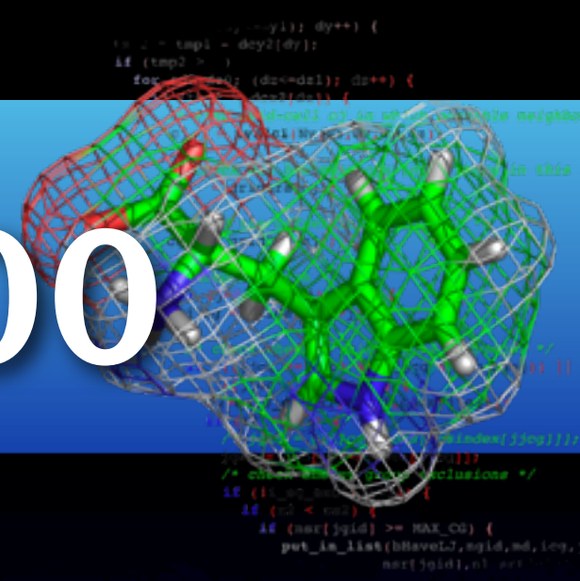
A large machine



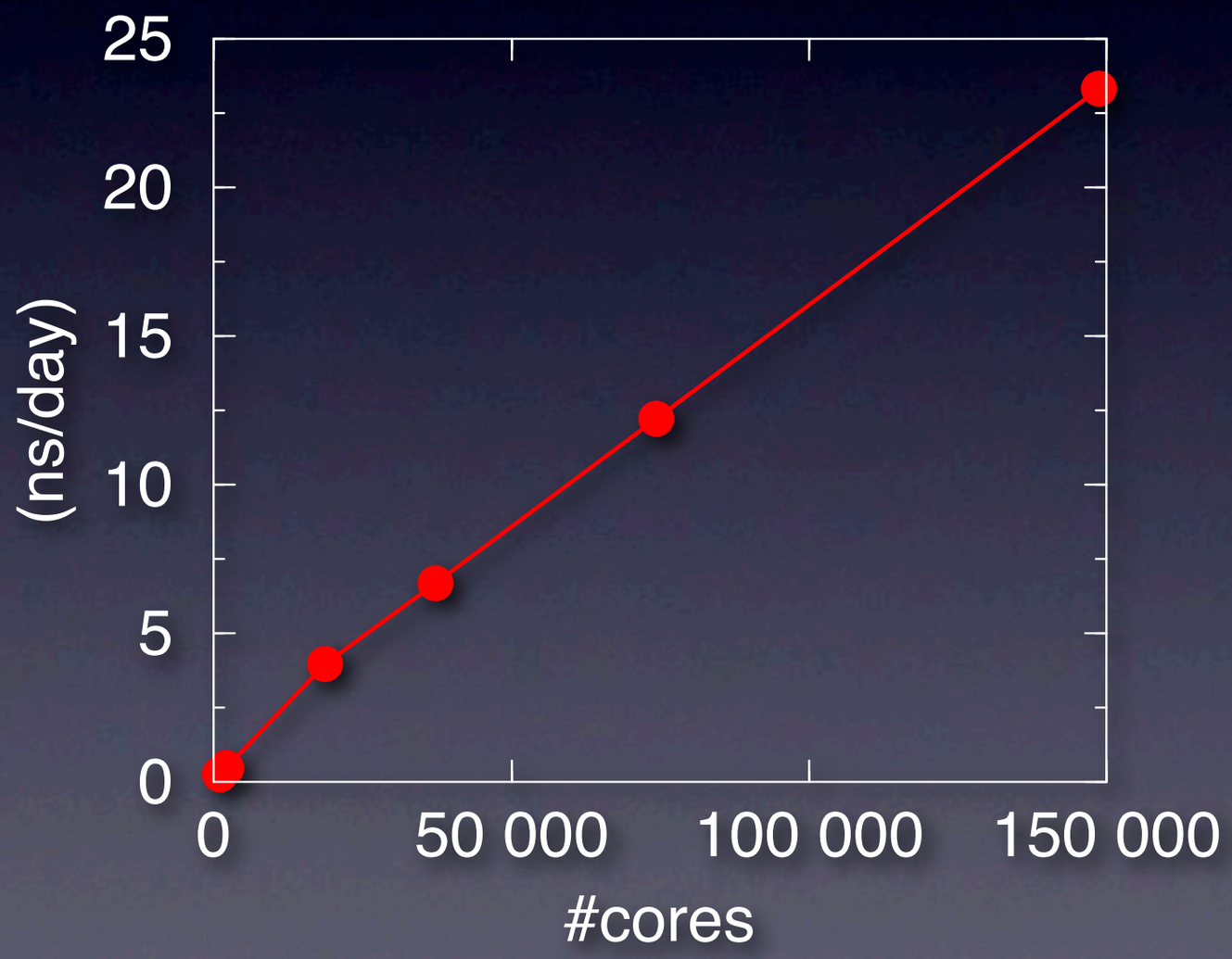
JaguarPF at Oak Ridge

- Cray XT5
- 150 000+ AMD Opteron 2.3 GHz cores
- SeaStar 2+ interconnect
- Upgrade planned to 450 000 cores

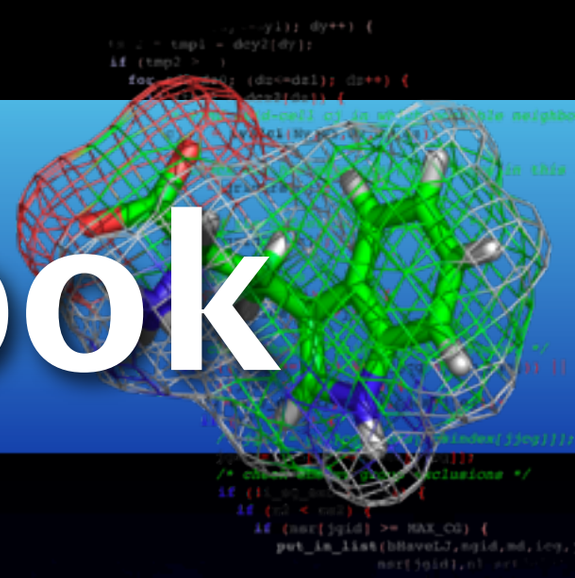
Scaling to 150 000



- peptides + H_2O
- 102M atoms
- Reaction-field
- 1.2 nm cut-off
- no DLB



GROMACS outlook



- Large systems:
  - Improve electrostatics scaling

- Medium systems:
  - Combine MPI with threads

- Small systems:
  - Distributed computing:
    GROMACS on Folding@Home