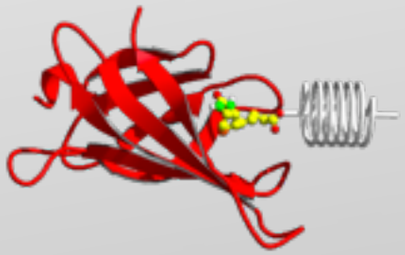
# Scaling of the GROMACS molecular dynamics code on SuperMUC

Carsten Kutzner, Helmut Grubmüller

MPI for biophysical Chemistry, Göttingen
Theoretical and Computational Biophysics Department
www.mpibpc.mpg.de/grubmueller



Sept 12

ParCo 2013 – International Conference on Parallel Computing, Munich

Mini-Symposium "Extreme Scaling on SuperMUC"

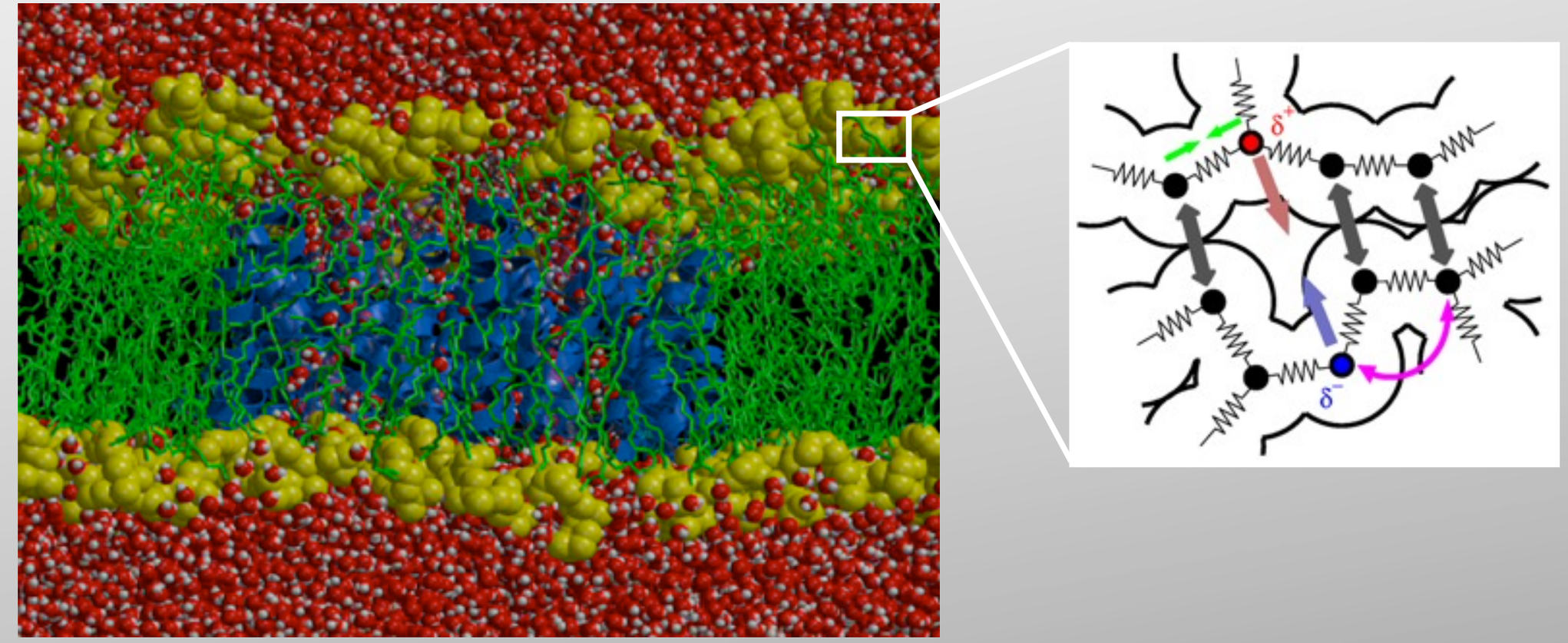
# Outline

- **GROMACS** molecular dynamics (MD)
  - challenges of parallel MD (mainly PME) and how GROMACS deals with them
- **Benchmark** setup
- **Results** – Scaling of GROMACS 4.6 on SuperMUC

# Molecular dynamics simulations with GROMACS

# Molecular dynamics simulations

- time-dependent motion of a set of i=1...n atoms
  - positions $\mathbf{r}_i$, charges $q_i$, masses $m_i$, velocities $\mathbf{v}_i$,
    and a "force field" / potential $U(\mathbf{r}_i, q_i, m_i, ...)$
  - calculate forces $\mathbf{F}_i = -\partial U/\partial \mathbf{r}_i$
    and solve Newton's eq. of motion $\mathbf{F}_i = m_i\, \mathbf{a}_i$
  - periodic boundaries




B.L. de Groot, H. Grubmüller, Science 294, 2353 (2001)

# Molecular dynamics simulations

- "force field" U defines potential energy

$$U = \sum_{\text{bonds}} \frac{1}{2} K_b (b - b_0)^2$$

$$+ \sum_{\text{angles}} \frac{1}{2} K_\Theta (\Theta - \Theta_0)^2$$

$$+ \sum_{\text{dihedrals}} \frac{1}{2} K_\varphi (1 + \cos(n\varphi - \delta))$$

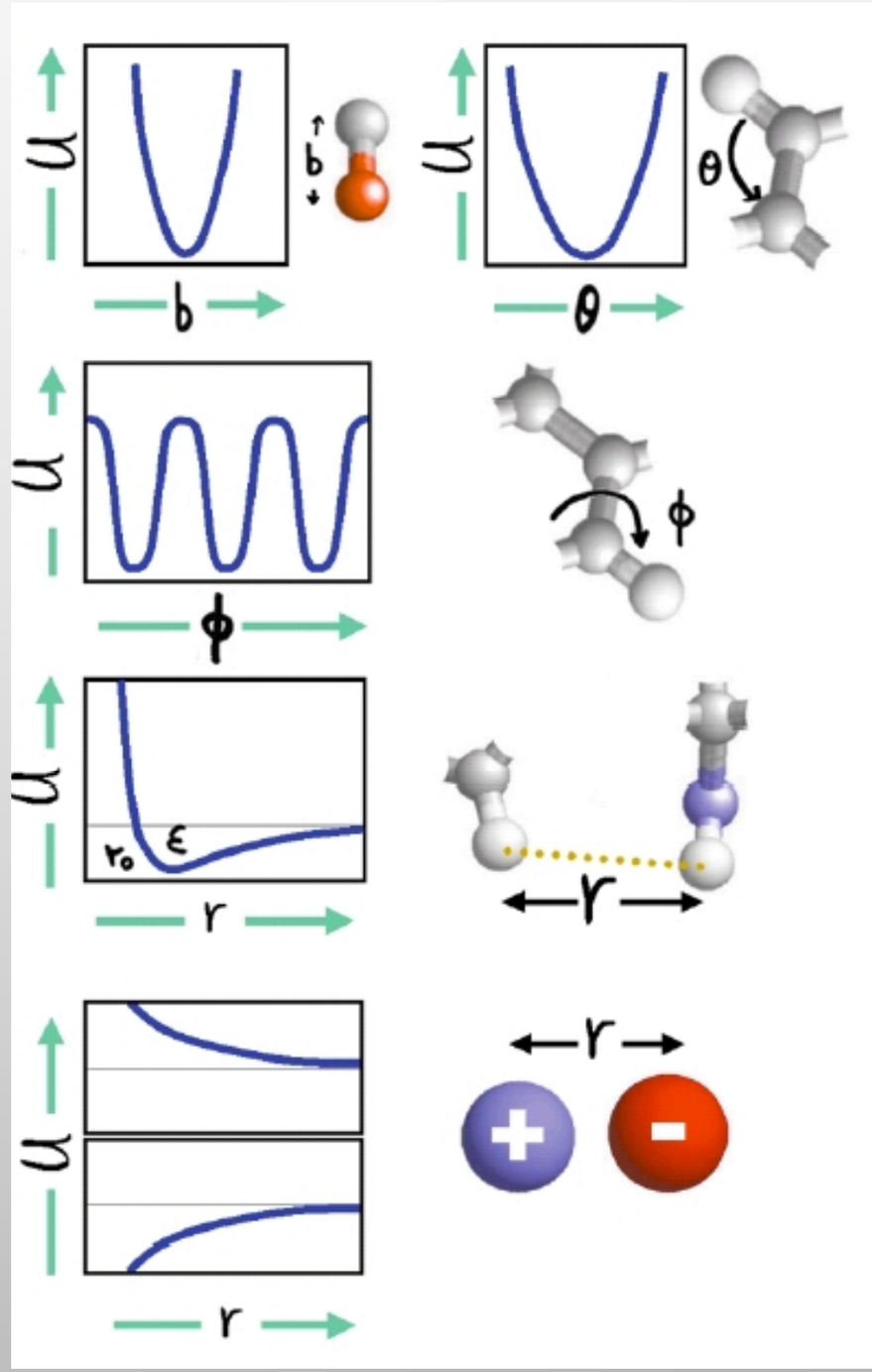
*cutoff* ⇨

$$+ \sum_{\text{pairs } i,j} \frac{C_{12}(i,j)}{r_{ij}^{12}} - \frac{C_6(i,j)}{r_{ij}^{6}}$$

*PME* ⇨

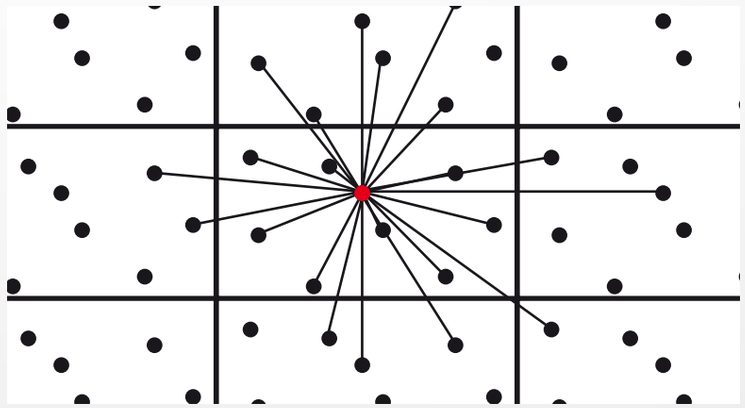
$$+ \sum_{\text{pairs } i,j} \frac{q_i q_j}{4\pi\varepsilon_0\varepsilon_r r_{ij}}$$ long-range!

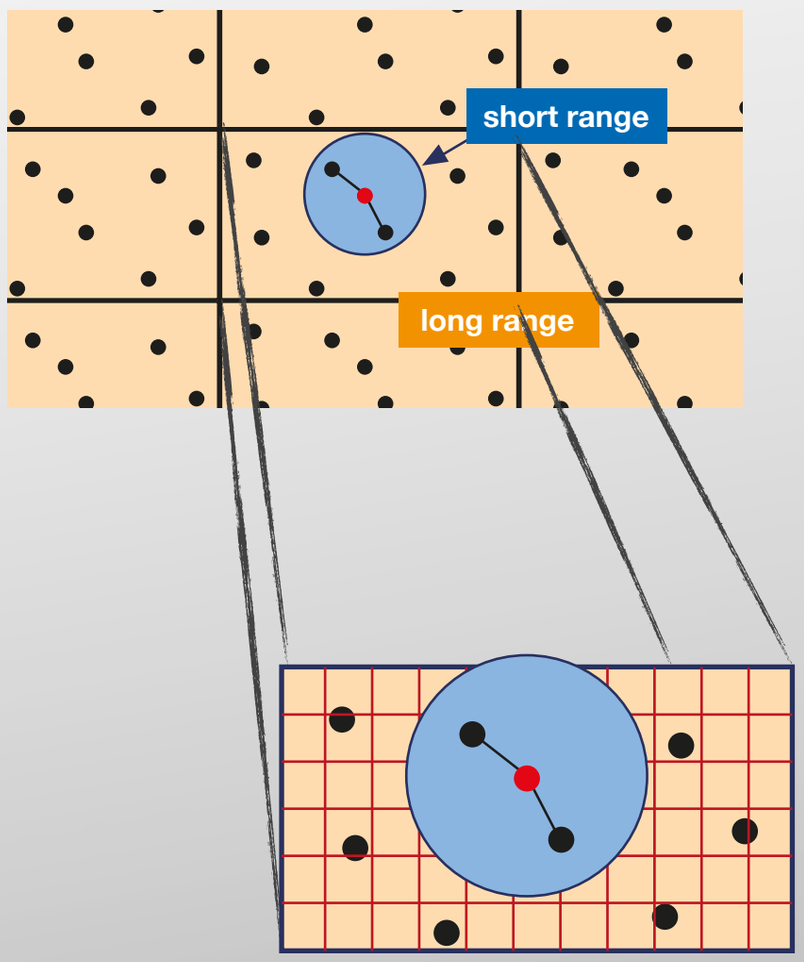
**>> 90% of calculation time!**



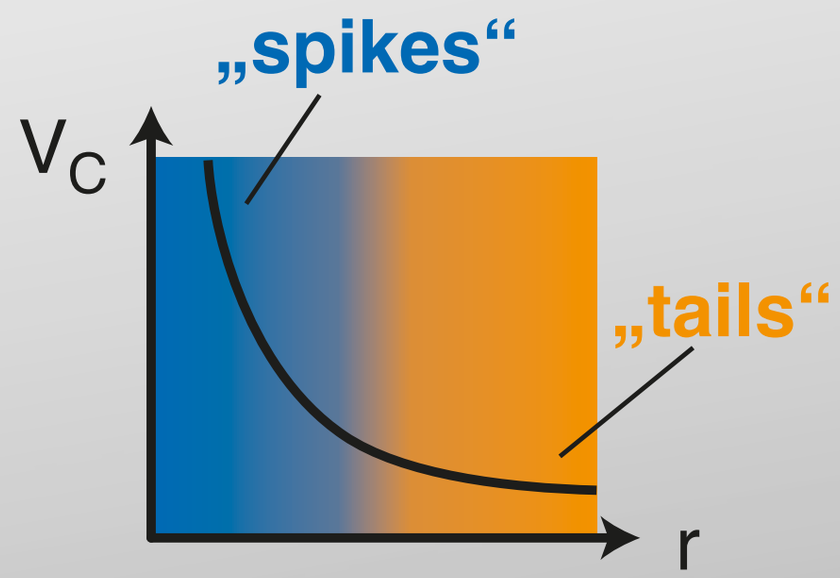

M.Levitt, Nat. Struct. Biol. 8, 392-393 (2001)

# Particle Mesh Ewald (PME) electrostatics

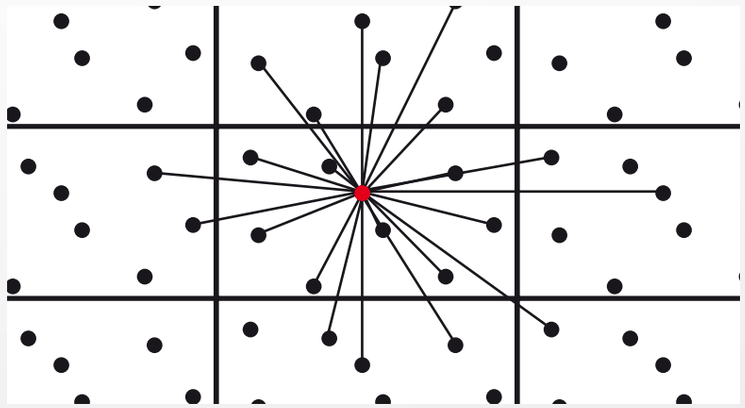


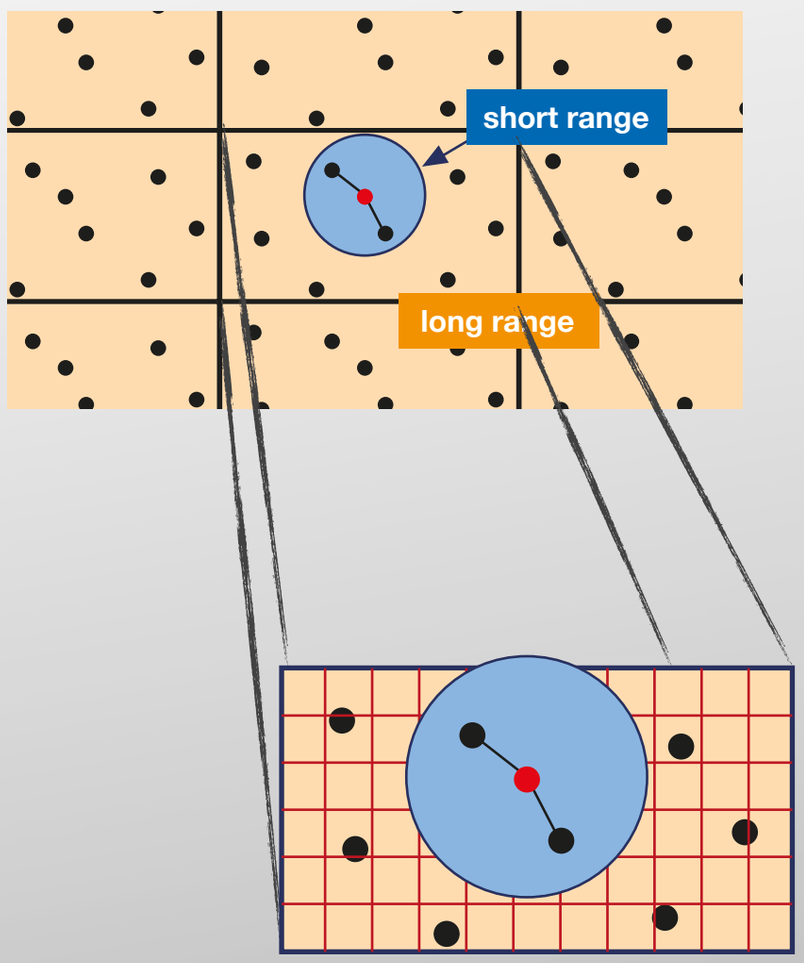



- Ewald summation splits Coulomb interactions in short range **SR** + long range **LR** part
  ("**spikes** + **tails**")
- calculate "**spiky**" **SR** part in direct space,
- calculate slowly varying **LR** part in reciprocal space.
- put charges on a Mesh → use FFT

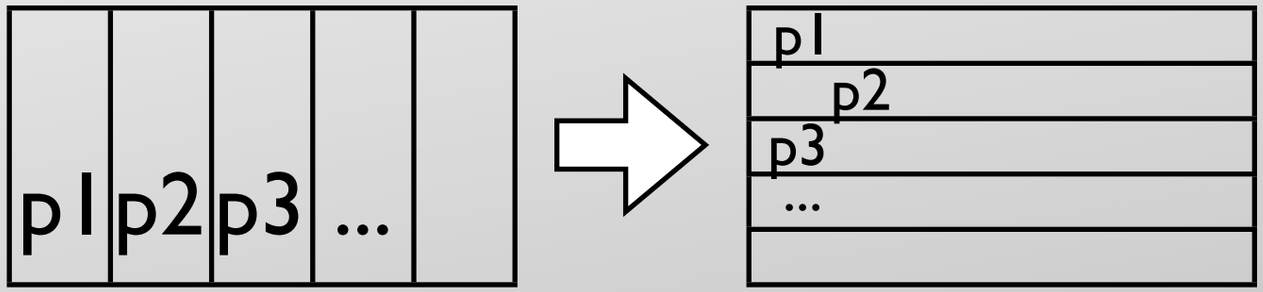


# Particle Mesh Ewald (PME) electrostatics

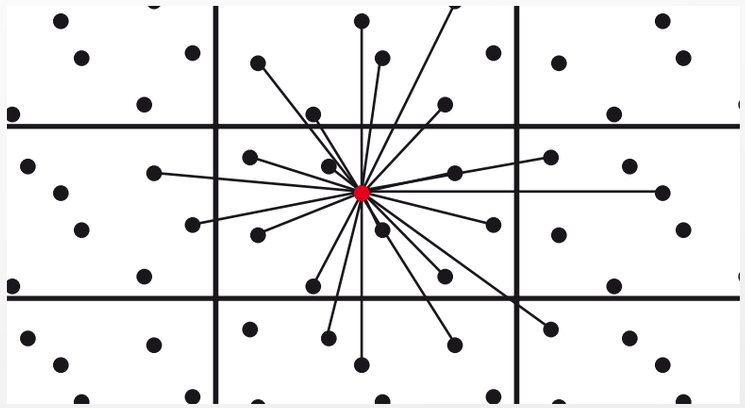


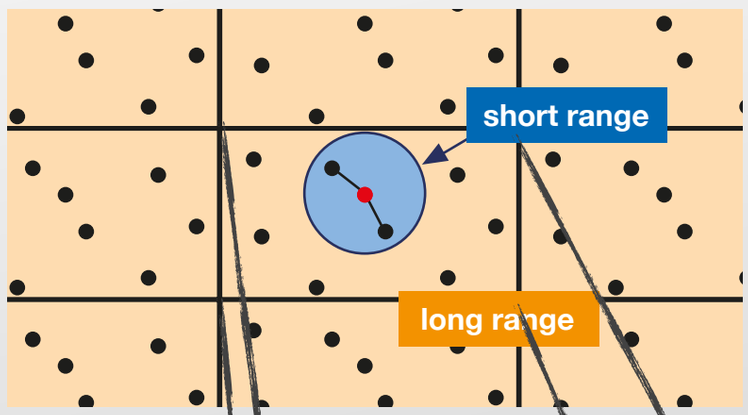



- Ewald summation splits Coulomb interactions in short range **SR** + long range **LR** part
  ("**spikes** + **tails**")
- calculate "**spiky**" **SR** part in direct space,
- calculate slowly varying **LR** part in reciprocal space.
- put charges on a Mesh → use FFT
  - **parallel** FFT requires **all-to-all** communication

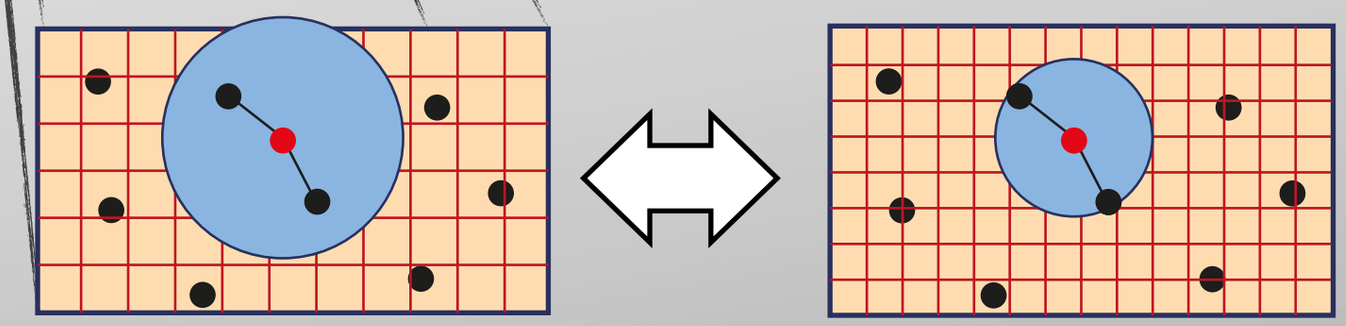


# Particle Mesh Ewald (PME) electrostatics





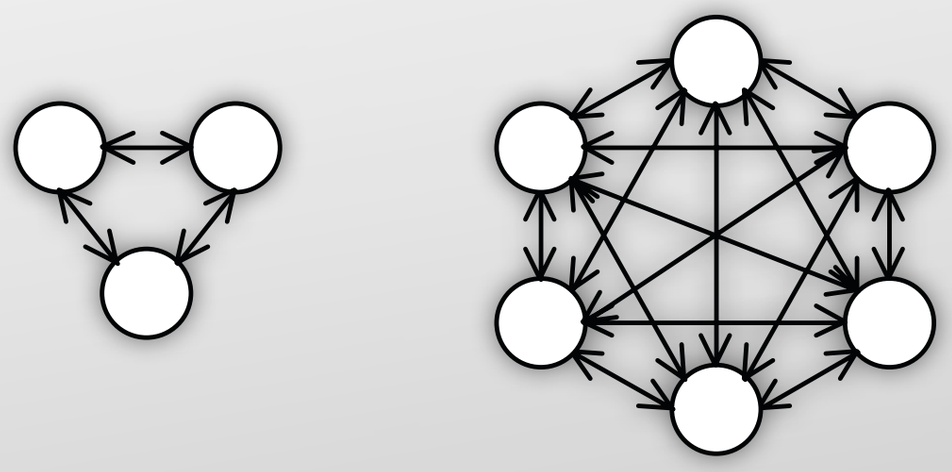

- Ewald summation splits Coulomb interactions in short range **SR** + long range **LR** part
  ("**spikes** + **tails**")
- calculate "**spiky**" **SR** part in direct space,
- calculate slowly varying **LR** part in reciprocal space.
- put charges on a Mesh → use FFT
  - **parallel** FFT requires **all-to-all** communication
- direct vs. reciprocal PME load can be shifted



# Parallel PME is a scaling bottleneck

- PME **calculation** cost is O(n•log n) with n atoms, but in parallel **communication** becomes most costly **(all-to-all)** at high number of processes N
- number of messages increases by $N^2$, therefore also total latency

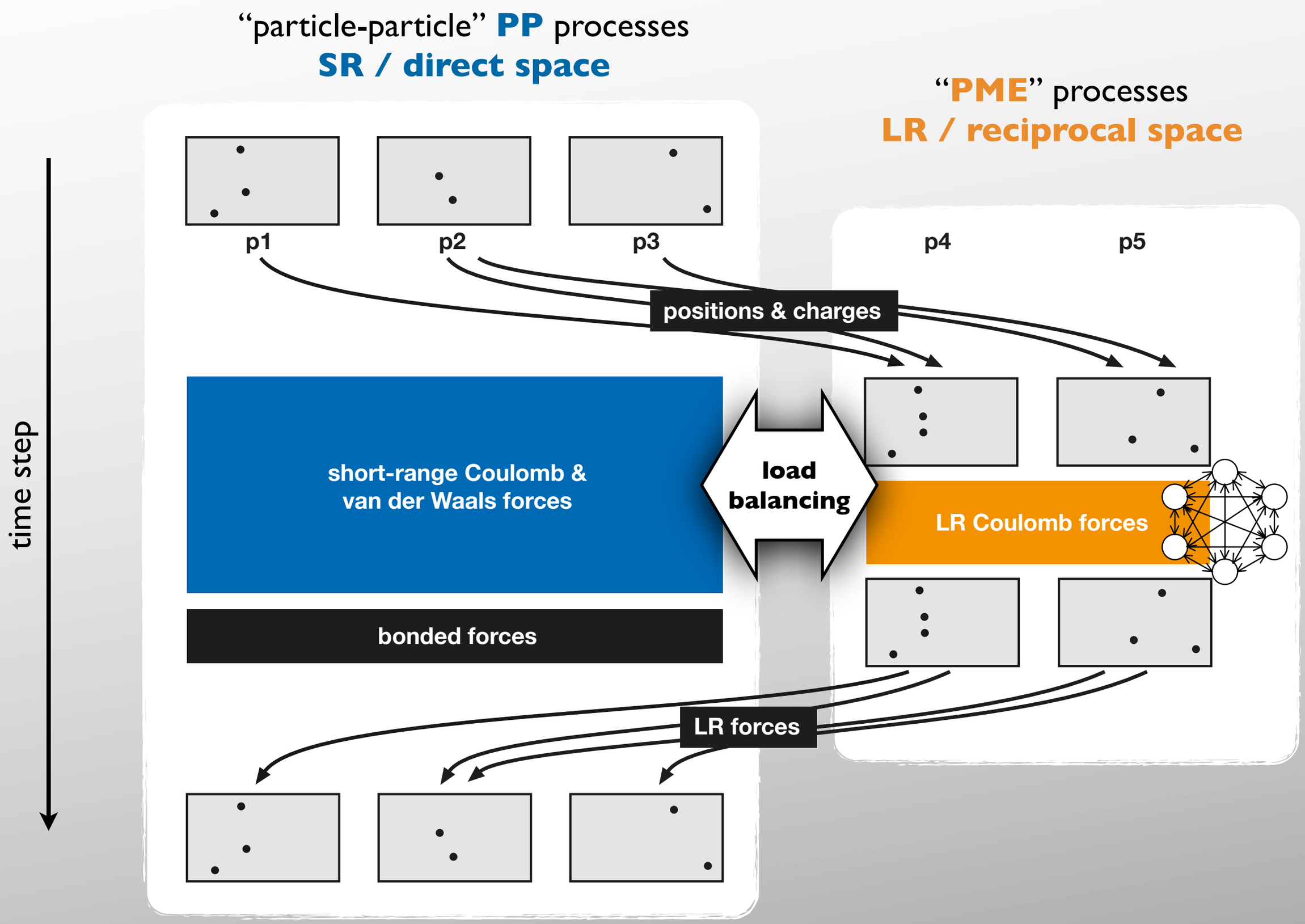
N•(N–1) messages



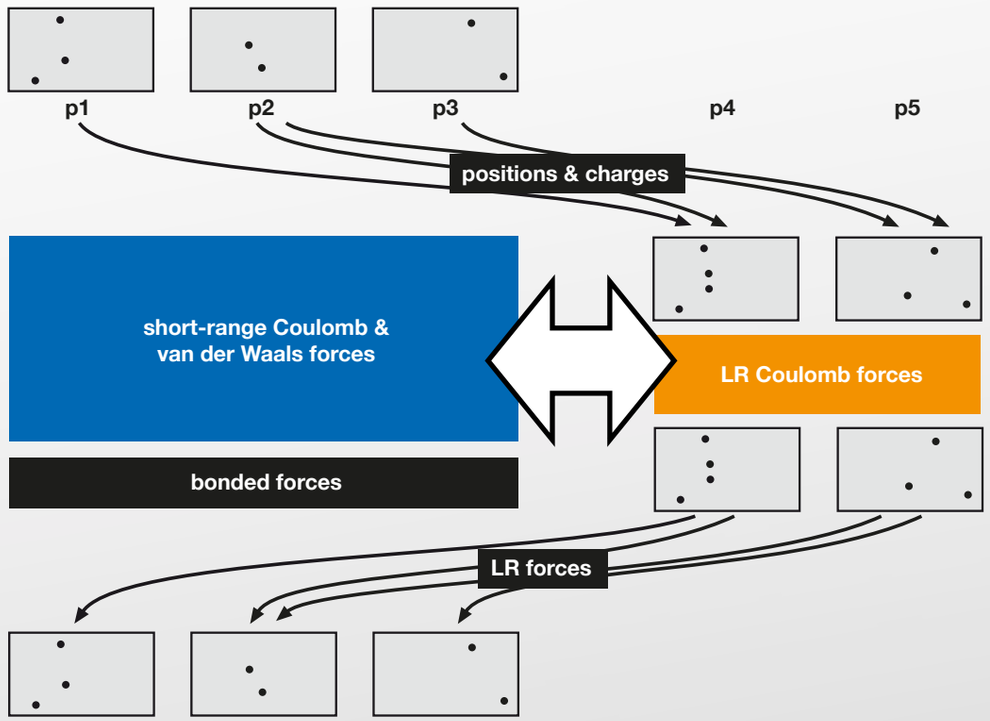
N=3 N=6

- GROMACS runs PME on a subset of the processors (typically 1/4 → number of messages reduced 16-fold!)

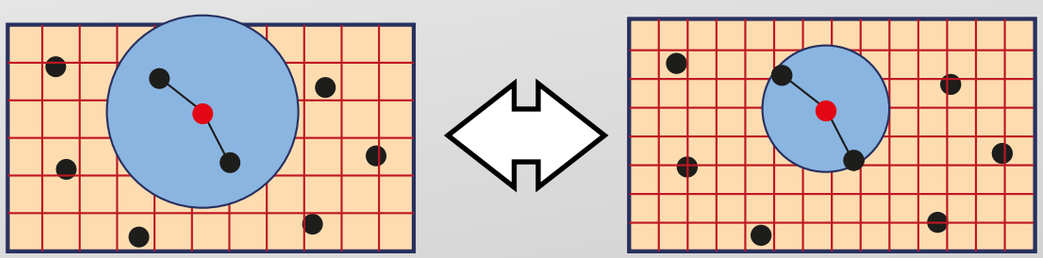
# Parallel PME in GROMACS



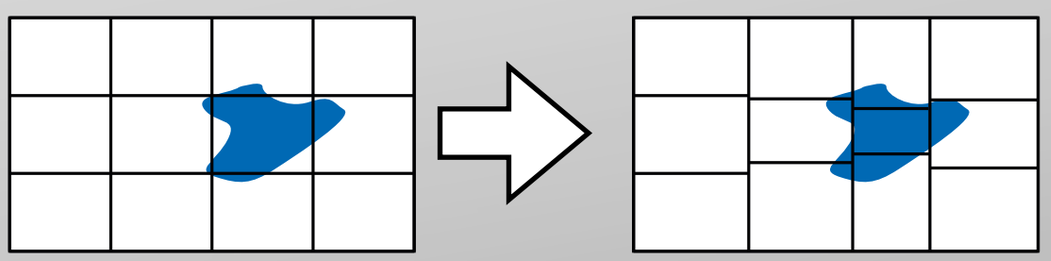
# 3 Load-balancing mechanisms




1. balance number of **PP** vs. **PME** processes

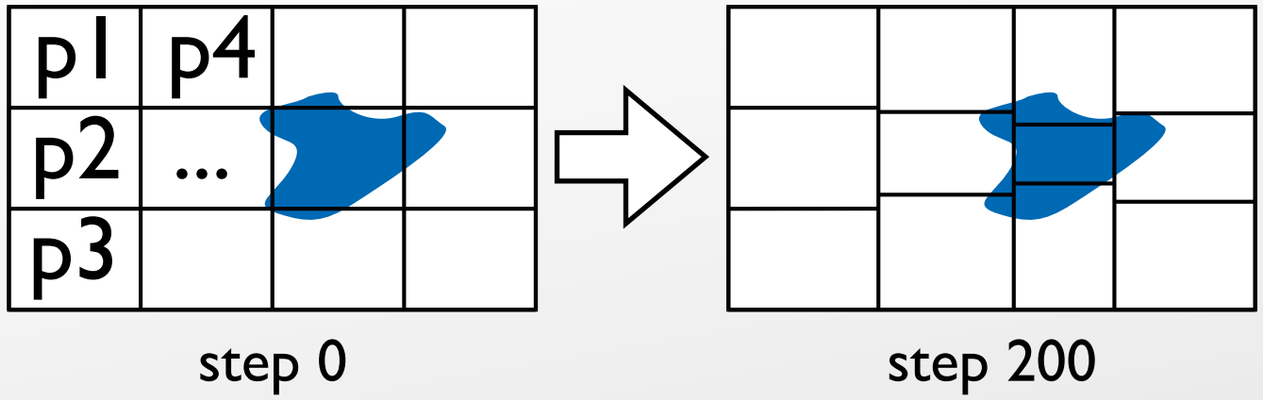


2. fine-tune **PP** vs. **PME** workload
   (balance cutoff : grid spacing)



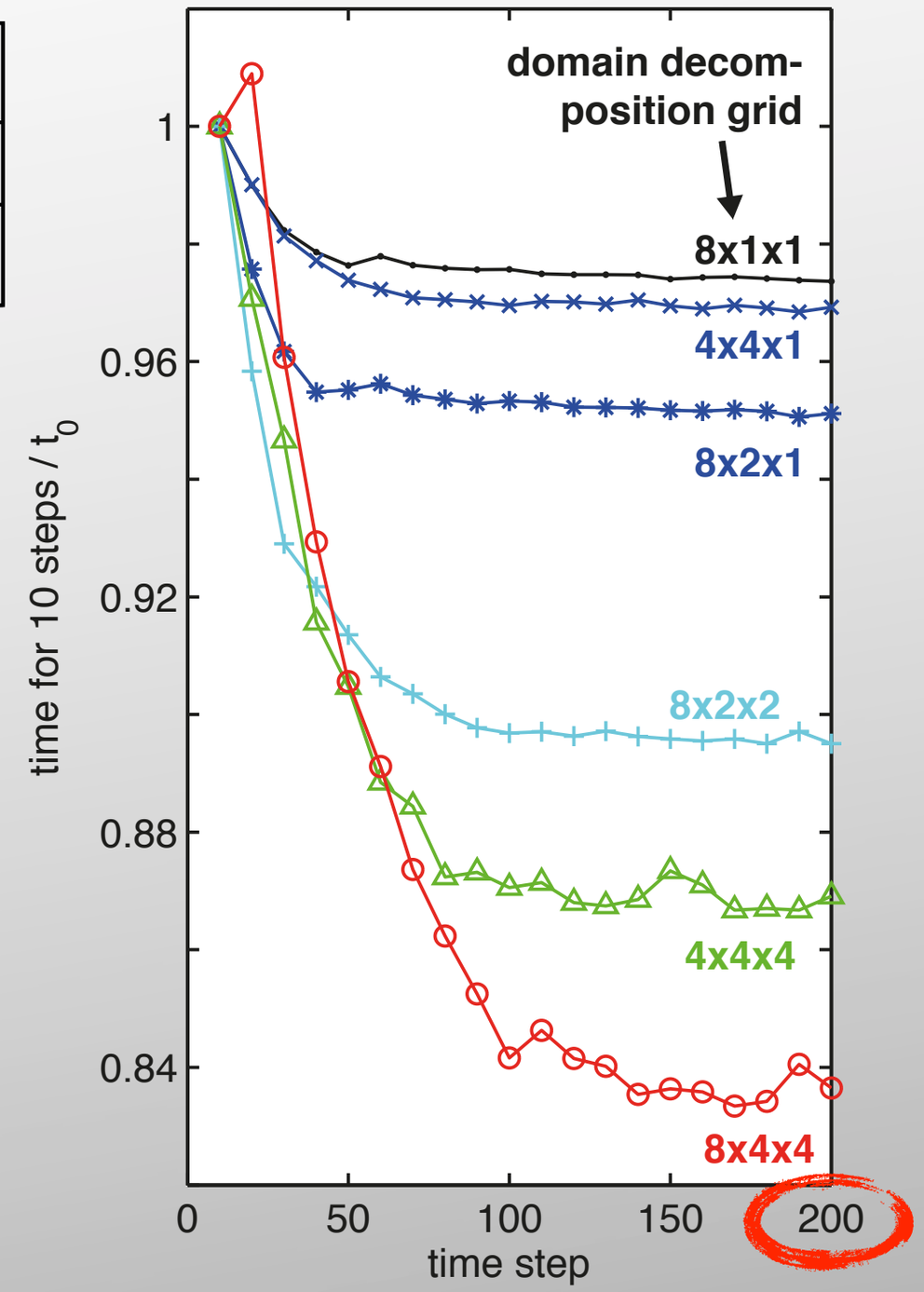
3. balance **direct space** workload
   between **PP** domains

# Dynamic load balancing (#3)




- domain decomposition for **direct space / short range parts**

- each MPI process gets assigned one of $N = n_x \times n_y \times n_z$ domains



# Find optimal PME : PP ratio (#1)

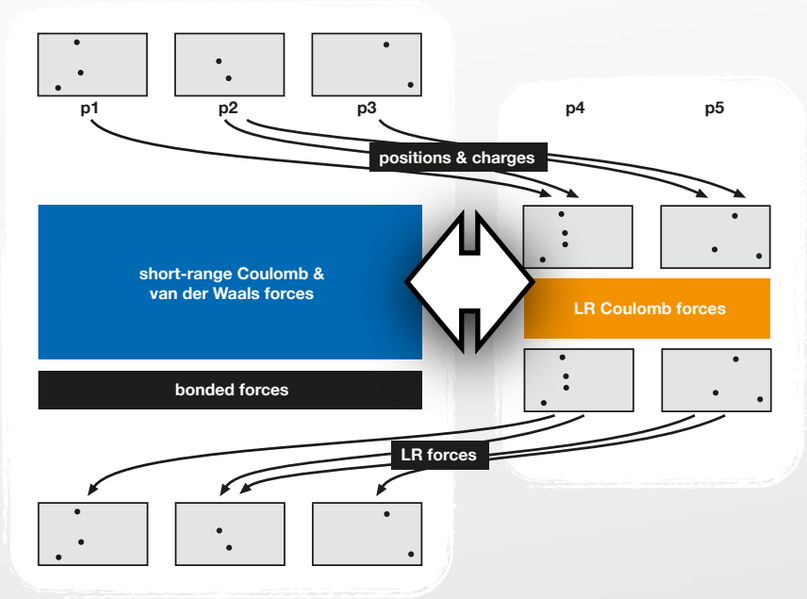
- GROMACS estimates **PP** : **PME** load, chooses near-optimal setting, e.g.

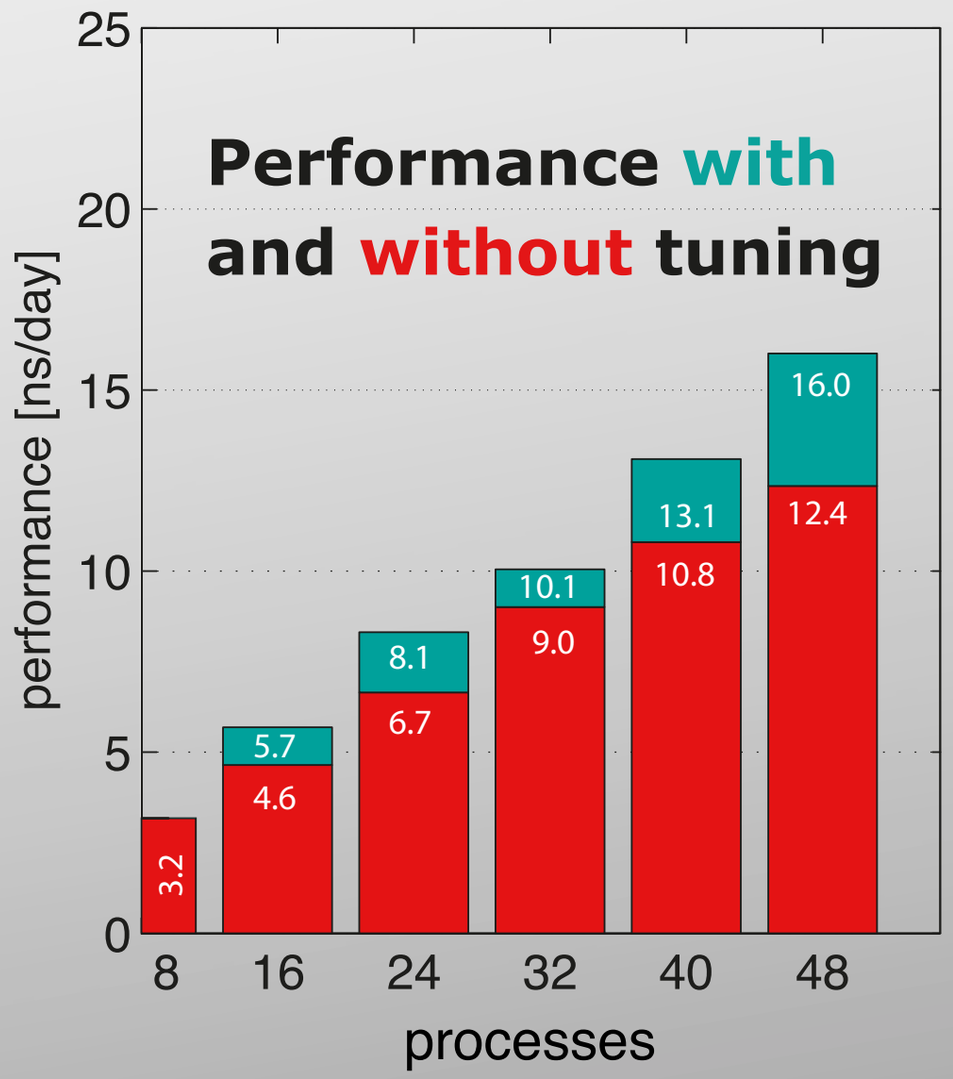
  **12 PP** + **4 PME** for 16 MPI processes

- use **g_tune_pme** to benchmarks settings around this value, e.g.

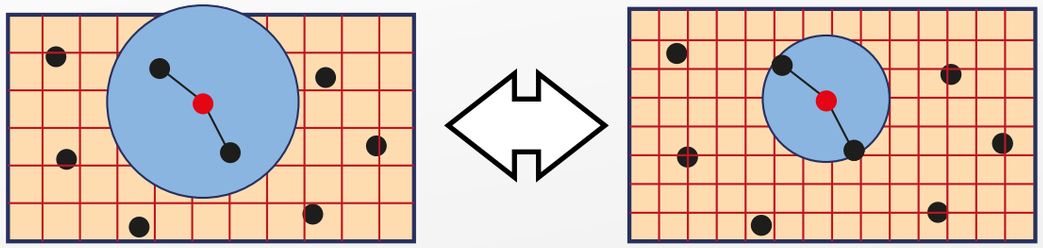
  **14** : **2**
  **13** : **3**
  **12** : **4** *
  **11** : **5**
  **10** : **6**
  **16** : **0** (no separate processes)

- 5–15 % extra performance!



# Fine-tune PME : PP workload (#2)



```
step  120: timed with pme grid 320 320 320, coulomb cutoff 1.200: 24209.0 M-cycles
step  200: timed with pme grid 288 288 288, coulomb cutoff 1.302: 22664.5 M-cycles
step  280: timed with pme grid 256 256 256, coulomb cutoff 1.465: 24579.1 M-cycles
step  360: timed with pme grid 224 224 224, coulomb cutoff 1.674: 33557.8 M-cycles
step  440: timed with pme grid 320 320 320, coulomb cutoff 1.200: 24507.7 M-cycles
step  520: timed with pme grid 300 300 300, coulomb cutoff 1.250: 24998.7 M-cycles
step  600: timed with pme grid 288 288 288, coulomb cutoff 1.302: 23082.2 M-cycles
step  680: timed with pme grid 280 280 280, coulomb cutoff 1.339: 23978.4 M-cycles
step  760: timed with pme grid 256 256 256, coulomb cutoff 1.465: 24737.4 M-cycles
step  840: timed with pme grid 240 240 240, coulomb cutoff 1.563: 28536.6 M-cycles
```

- **PP** : **PME** workload fine tuning needs time
- Reject the initial balancing phase when benchmarking!

```
mdrun -resetstep 1000
mdrun -resethway
```

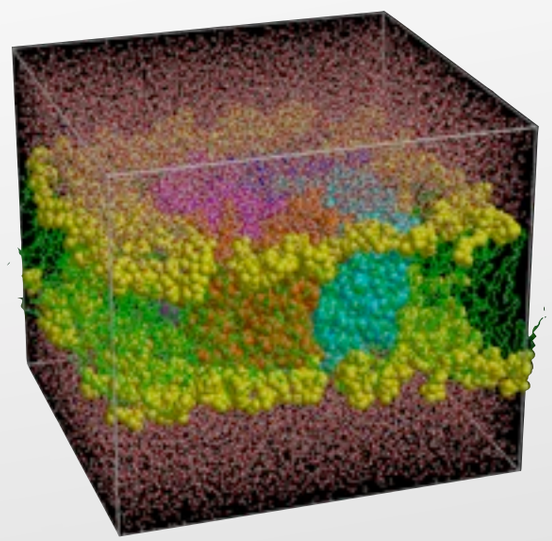
# GROMACS 4.6
# benchmarks on SuperMUC

# Benchmark protocol

- Gromacs 4.6 compilation:
  - with **IBM** MPI & icc12.1.6 ←→ **Intel** MPI 4.1 & icc 13.1.1
  - FFTW 3.3.2 (SSE2)
  - -O3 -mavx compiler flags
  - OpenMP support
    (each MPI process can use several OpenMP threads)
- 2.7 GHz clock rate
- vary number of MPI processes per node
  (32, 16, 8, 4, 2, 1)
- vary number of OpenMP threads
  per MPI process (1, 2, 4, 8, 16)
- using g_tune_pme,
  no timings taken during first half of benchmark

Find optimal performance for any number of nodes!

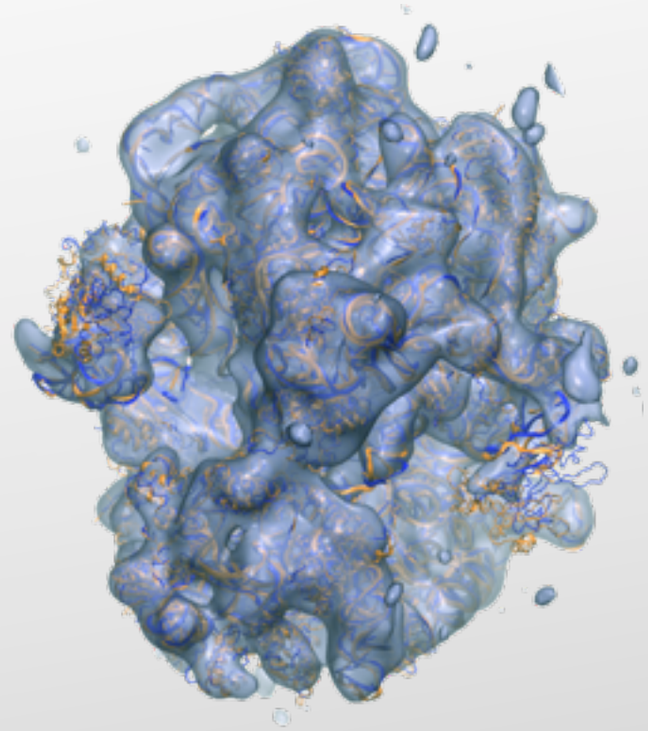
# Three "real world" benchmark systems:



"Aquaporin-1 channel"

- **81,743 atoms**
- 2 fs time step
- cutoffs @ 1.0 nm
- PME grid spacing 0.120 nm
- 10,000 steps

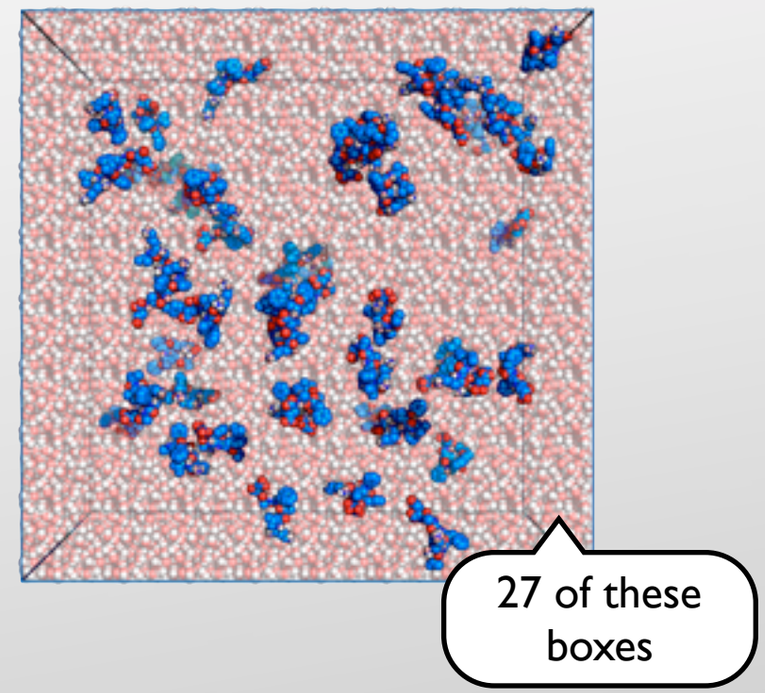
- de Groot, Grubmüller, Science 294, 2353 (2001)



"Ribosome"

- **2,136,412 atoms**
- 4 fs time step
- cutoffs @ 1.0 nm
- PME grid spacing 0.135 nm
- 2,000 steps

- Fischer, Konevega, Wintermeyer, Rodnina, Stark, Nature 466 (2010), 329–333

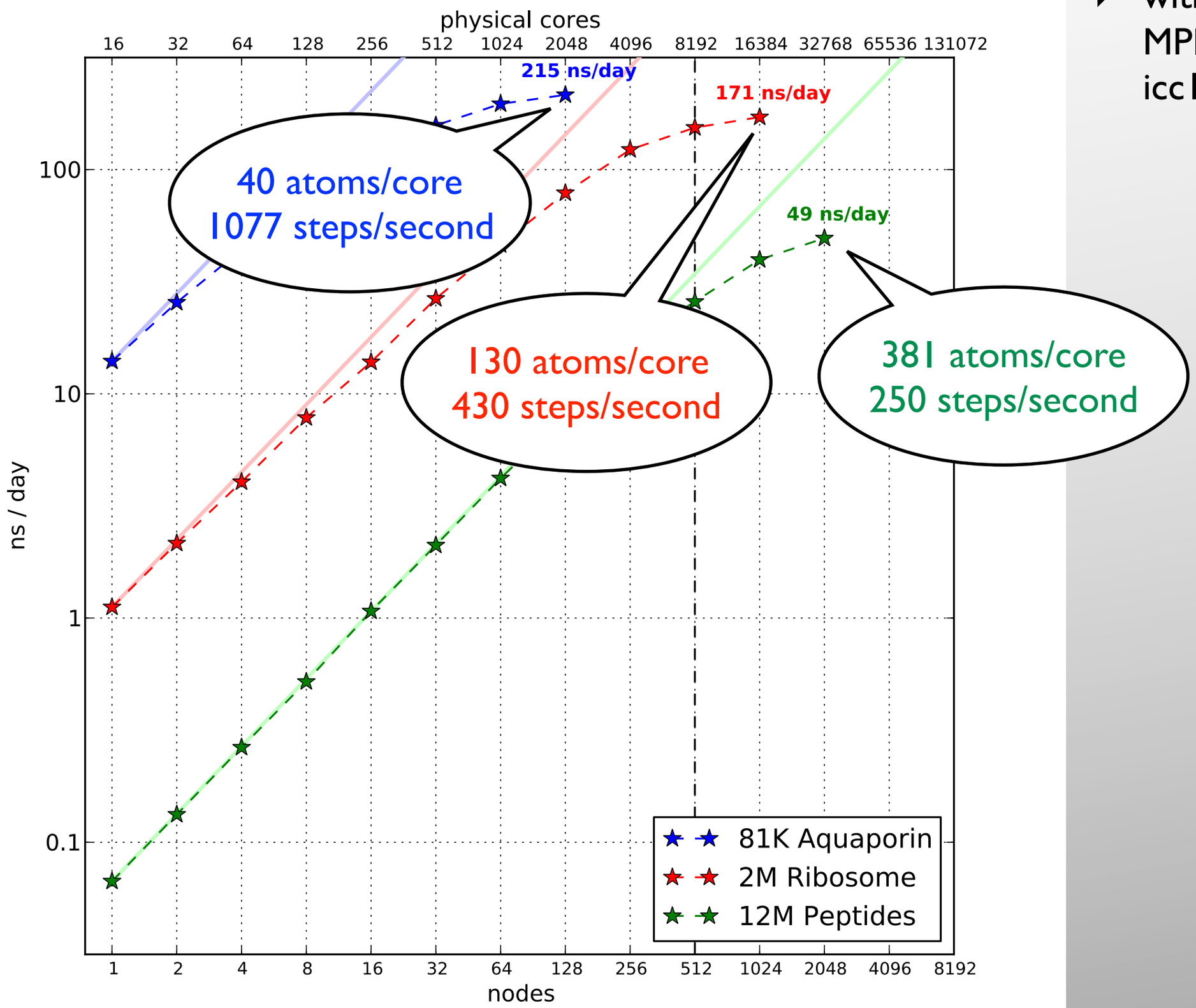



"Peptide aggregation"

- **12,495,503 atoms**
- 2 fs time step
- cutoffs @ 1.2 nm
- PME grid spacing 0.160 nm
- 500 steps

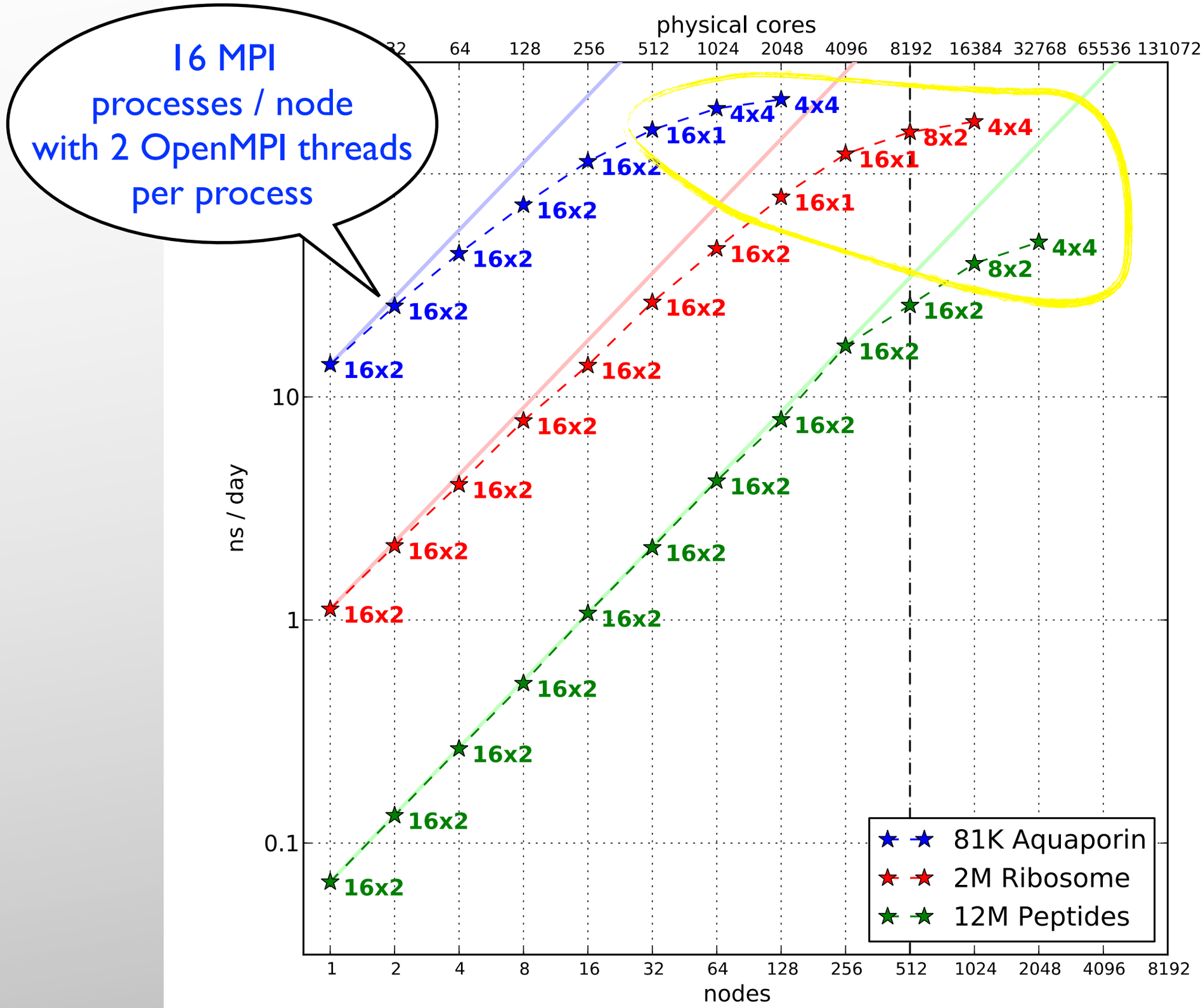
- Matthes, Gapsys, de Groot, J. Mol. Biol. 421, 390–416 (2012)

# GROMACS performance on SuperMUC



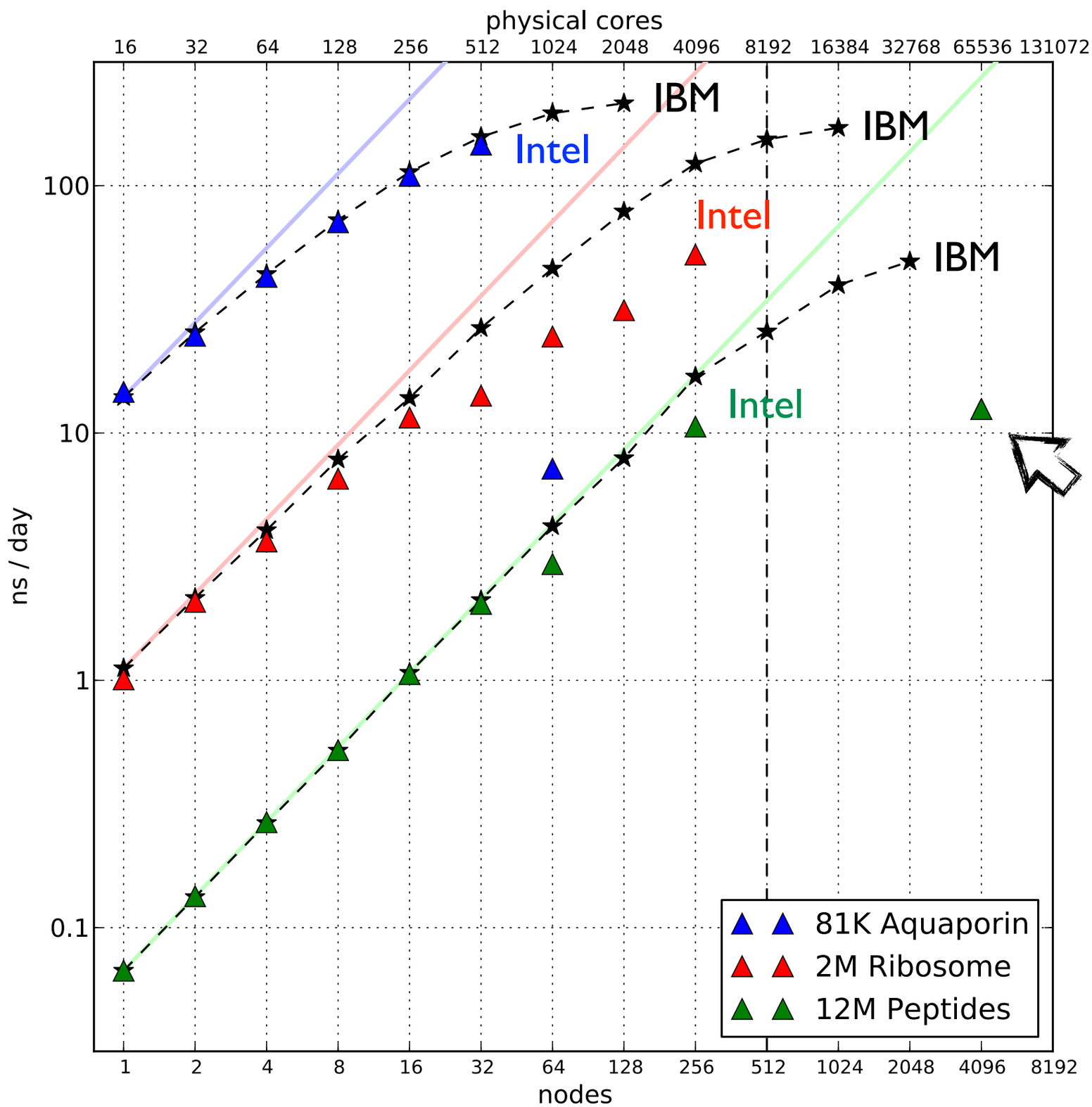

- with **IBM** MPI & icc12.1.6

# GROMACS performance on SuperMUC




- with **IBM** MPI & icc12.1.6
- 16 MPI x 2 OpenMP is fastest, except at high parallelization

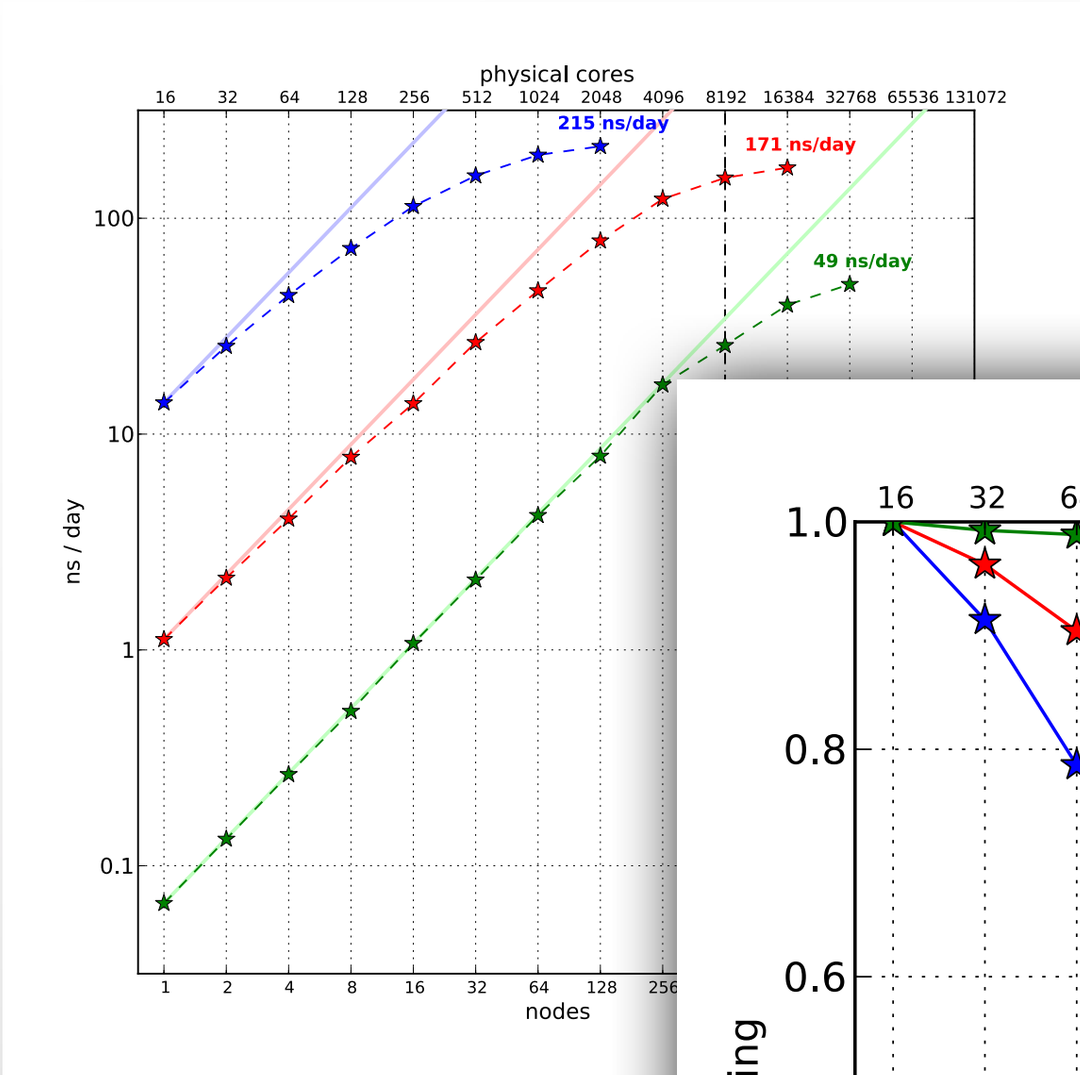
# Still some issues using Intel MPI

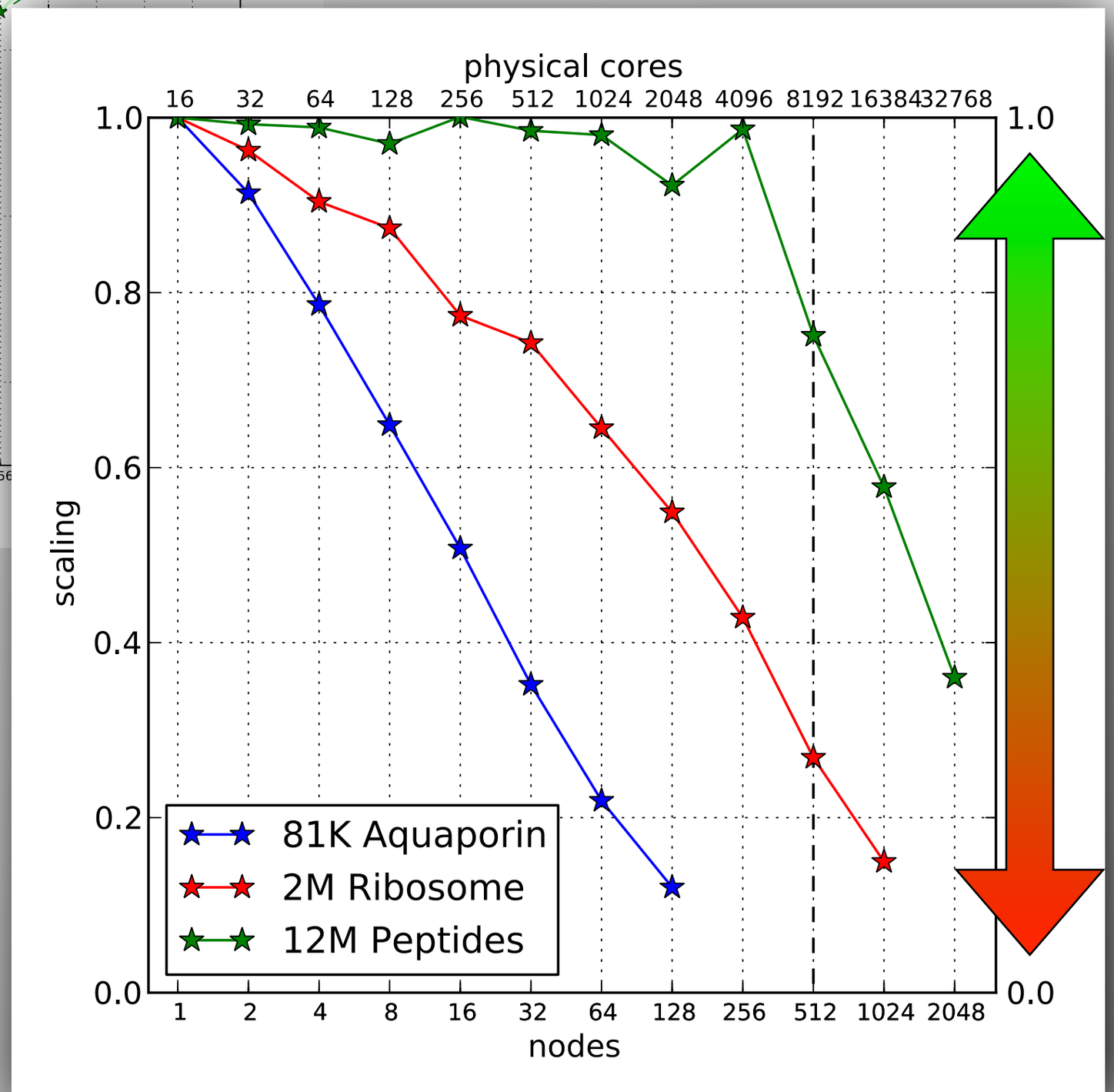



- with **Intel** MPI & icc13.1.1

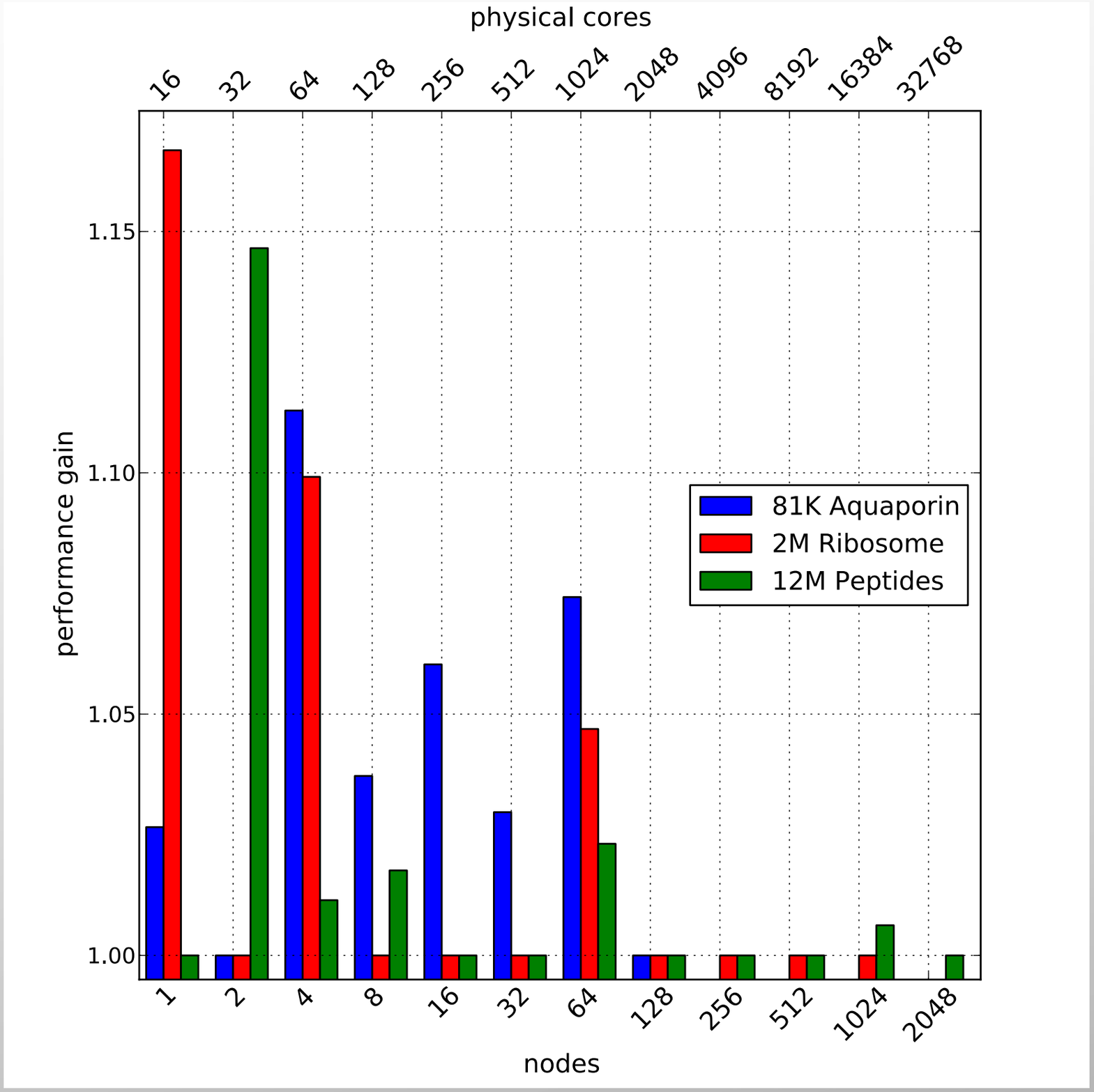
# Scaling

- $S(N) = t_1 / (N \cdot t_N)$

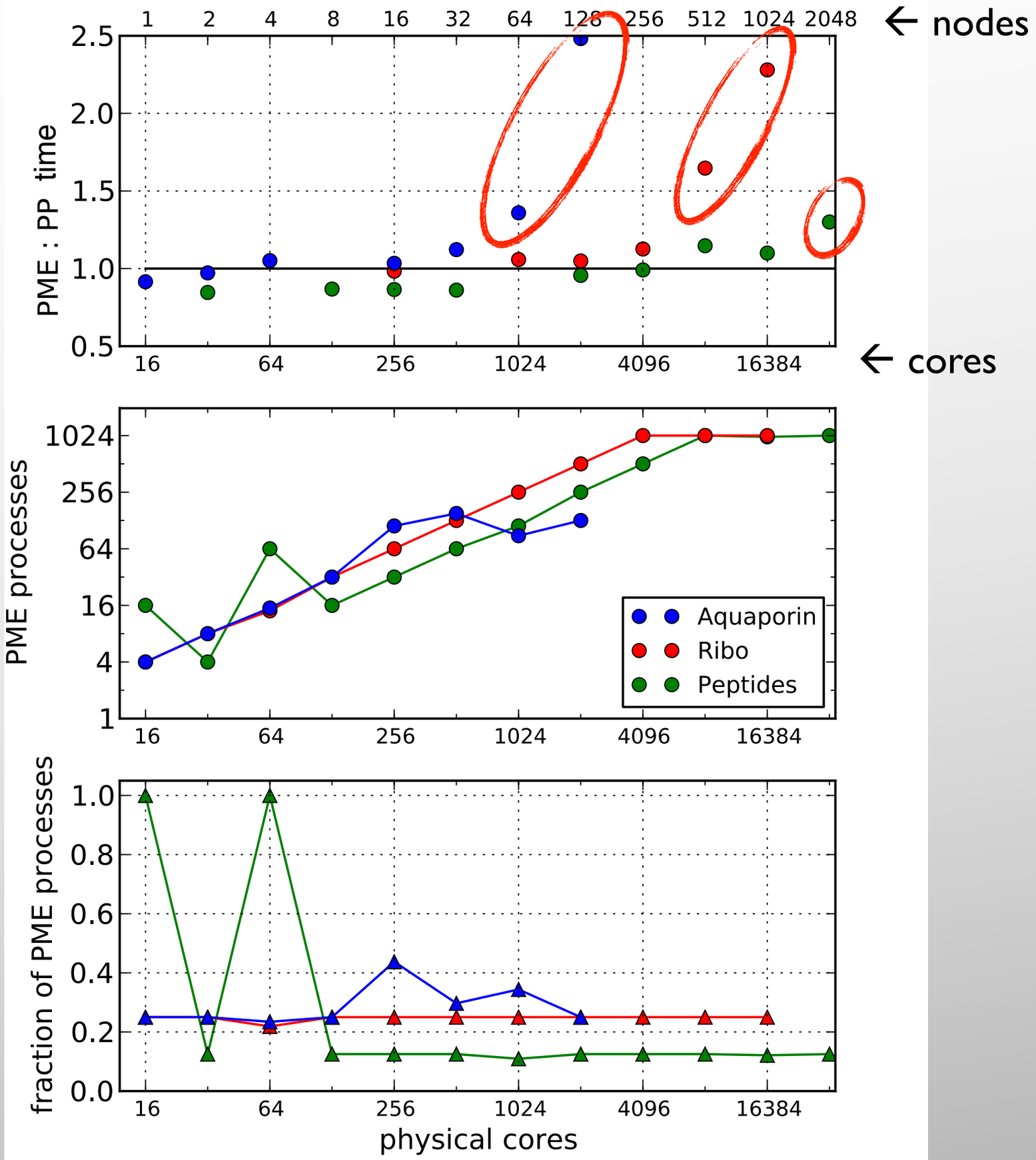


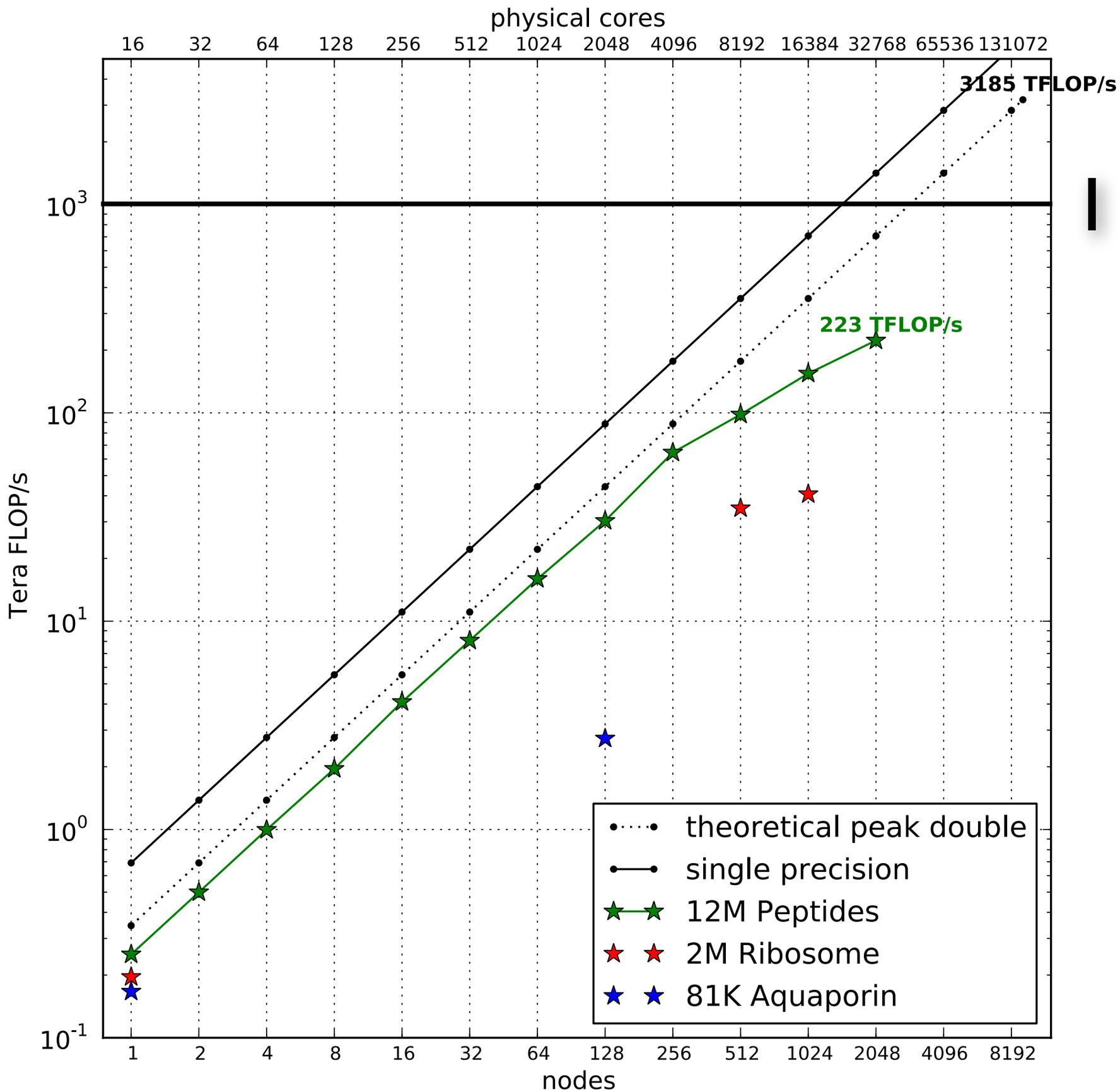
# Performance gain due to g_tune_pme




- with **IBM** MPI & icc12.1.6

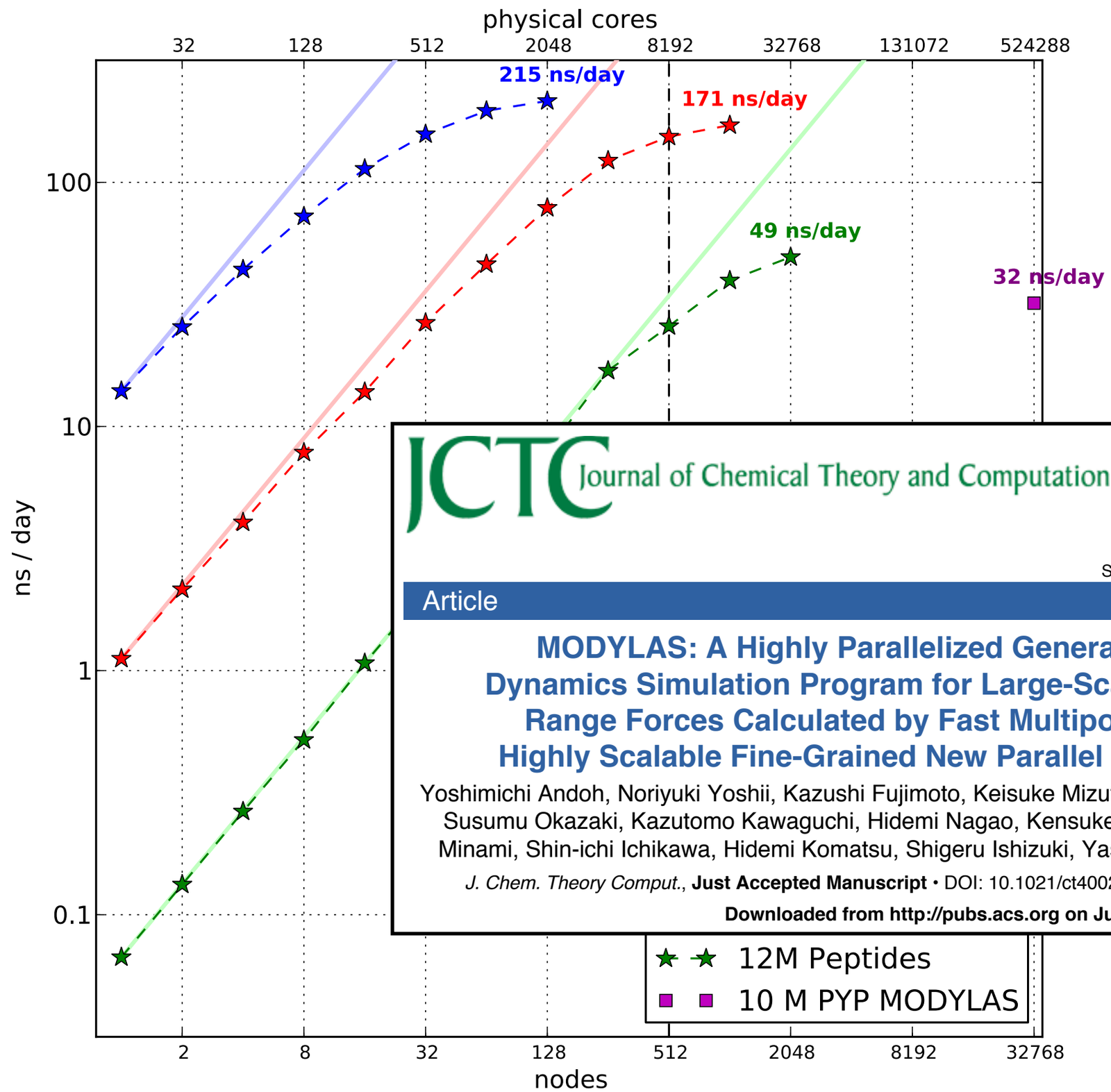



# PME limits the scaling


physical cores
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
3185 TFLOP/s
223 TFLOP/s
Tera FLOP/s
$10^3$
$10^2$
$10^1$
$10^0$
$10^{-1}$
theoretical peak double
single precision
12M Peptides
2M Ribosome
81K Aquaporin
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
nodes

# 1 PFLOP/s

# GROMACS performance on SuperMUC




JCTC Journal of Chemical Theory and Computation

Subscriber access provided by MPI FUR BIOPHYS CHEM

Article

**MODYLAS: A Highly Parallelized General-Purpose Molecular Dynamics Simulation Program for Large-Scale Systems with Long-Range Forces Calculated by Fast Multipole Method (FMM) and Highly Scalable Fine-Grained New Parallel Processing Algorithms**

Yoshimichi Andoh, Noriyuki Yoshii, Kazushi Fujimoto, Keisuke Mizutani, Hidekazu Kojima, Atsushi Yamada, Susumu Okazaki, Kazutomo Kawaguchi, Hidemi Nagao, Kensuke Iwahashi, Fumiyasu Mizutani, Kazuo Minami, Shin-ichi Ichikawa, Hidemi Komatsu, Shigeru Ishizuki, Yasuhiro Takeda, and Masao Fukushima

*J. Chem. Theory Comput.*, **Just Accepted Manuscript** • DOI: 10.1021/ct400203a • Publication Date (Web): 11 Jun 2013

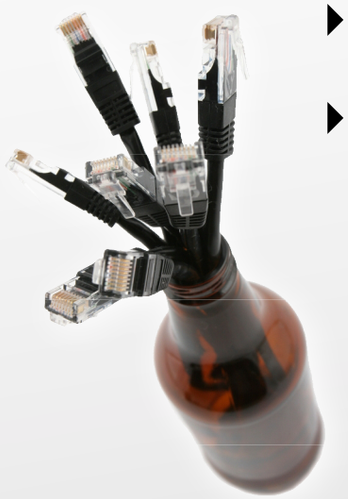
**Downloaded from http://pubs.acs.org on June 14, 2013**

## Modylas on K-computer

10 Petaflops,
88.128 Sparc64 CPUs
@2 GHz and 8 cores

# Conclusions

- Scaled GROMACS to 4096 nodes (65,000 cores, 8 islands)



  - 12 M atom system reached 49 ns/day on 32,000 cores, 223 TFLOP/s
  - PME / all-to-all is major scaling bottleneck,
    - <= 1024 MPI processes for PME!

# Outlook

- resolve problems when using Intel MPI
- scale even larger system to whole SuperMUC? → Petaflop?
- can we get time steps < 1 ms at high parallelization?

- Long term: replace PME electrostatics by FMM

SPPEXA — JÜLICH FORSCHUNGSZENTRUM

http://www.mpibpc.mpg.de/grubmueller/sppexa

# Acknowledgments / people involved


- Ferdinand Jamitzky, Nicolay Hammer, Christoph Bernau, Matthias Brehm
  - SuperMUC *extreme scaling workshop organization, GROMACS on SuperMUC experiences*

- Florian Merz (IBM), Markus Rampp (RZG)
  - *IBM MPI settings & performance analysis tools*

- Heinrich Bockhorst, Klaus-Dieter Oertel
  - *Intel MPI performance tools & debugging*

- Rossen Apostolov, Berk Hess
  - *getting optimal GROMACS performance on SuperMUC*

# Job file IBM MPI

```
#!/bin/bash
#
#@ job_type = parallel
#@ class = general
#@ node = 64
### schedule the job to A to B islands
#@ island_count=1
#@ tasks_per_node = 16
#@ wall_clock_limit = 1:00:00
#@ job_name = RIBO_n64_mpi16_th2
#@ network.MPI = sn_all,not_shared,us
### Energy saving options
#@ energy_policy_tag = NONE
#@ output = job$(jobid).out
#@ error = job$(jobid).err
#@ notification=always
#@ queue
. /etc/profile
. /etc/profile.d/modules.sh

#setup of environment
module load mpi.ibm
module load lrztools

export MP_BULK_MIN_MSG_SIZE=32768
export OMP_NUM_THREADS=2
export MP_TASK_AFFINITY=core:2
export MPIRUN=mpiexec
export PROCS=1024
export MDRUN=/gpfs/work/pr86se/lu78tis/gromacs/4.6/463-ibmmpi-fftw332-icc121/bin/mdrun

# $MPIRUN -n ${PROCS} ${MDRUN} -dlb yes -s Ribo10kBench.tpr -noconfout -maxh 0.25 -gcom 1000 -npme 0

/gpfs/work/pr86se/lu78tis/gromacs/4.6/463-ibmmpi-fftw332-icc121/bin/g_tune_pme -np ${PROCS} -npstring -n
-dlb yes -s Ribo10kBench.tpr -steps 1000 -resetstep 1000 -noconfout -r 1 -ntpr 1 -npme all
```

use RDMA for messages larger than this

# Job file Intel MPI

```
#!/bin/bash
#
#@ job_type = MPICH
#@ class = general
#@ node = 64
### schedule the job to A to B islands
#@ island_count=1
#@ tasks_per_node = 16
#@ wall_clock_limit = 1:00:00
#@ job_name = RIBO_n64_mpi16_th2_MPICH
#@ network.MPI = sn_all,not_shared,us
### Energy saving options
#@ energy_policy_tag = NONE
#@ output = job$(jobid).out
#@ error = job$(jobid).err
#@ notification=always
#@ queue
. /etc/profile
. /etc/profile.d/modules.sh
#setup of environment
module unload mpi.ibm
module load mpi.intel/4.1.1
##################################################
module load lrztools

export OMP_NUM_THREADS=2
if [ "$OMP_NUM_THREADS" -gt 1 ] ; then
  module load mpi_pinning/hybrid_blocked
else
  module load mpi_pinning/mpp
fi
export I_MPI_PIN_DOMAIN=auto
export I_MPI_PIN_CELL=unit

export I_MPI_DEBUG=5
cpuinfo
##################################################
export MPIRUN=mpiexec
export PROCS=1024
export MDRUN=/gpfs/work/pr86se/lu78tis/gromacs/4.6/                    cc131-3/bin/mdrun
export I_MPI_DAPL_DIRECT_COPY_THRESHOLD=262114

/gpfs/work/pr86se/lu78tis/gromacs/4.6/462-impi41-fftw332-icc131-3/bin/g_tune_pme -np ${PROCS} -npstring -n -dlb
yes -s Ribo10kBench.tpr -steps 2000 -resetstep 2000 -noconfout -r 1 -ntpr 1
```

workaround since mpiexec defines some I_MPI variables in the wrong way

use RDMA for messages larger than this