

I Abinitio Molecular Dynamics Simulations

Integration of Newton's equation of motion

$$m_i \ddot{r}_i(t) = F_i, i = 1, \dots, N_{at}$$

Quantum mechanics (DFT)

- explicit treatment of electrons
→ wave functions / orbitals
- Basis: plane waves
→ Distributed 3D-FFT
→ Distributed Matrix Matrix Multiplies

Typical CPMD requirements

- time step: $\Delta t = 0.15$ fs
- single trajectory:
75 ps → 500.000 integration steps

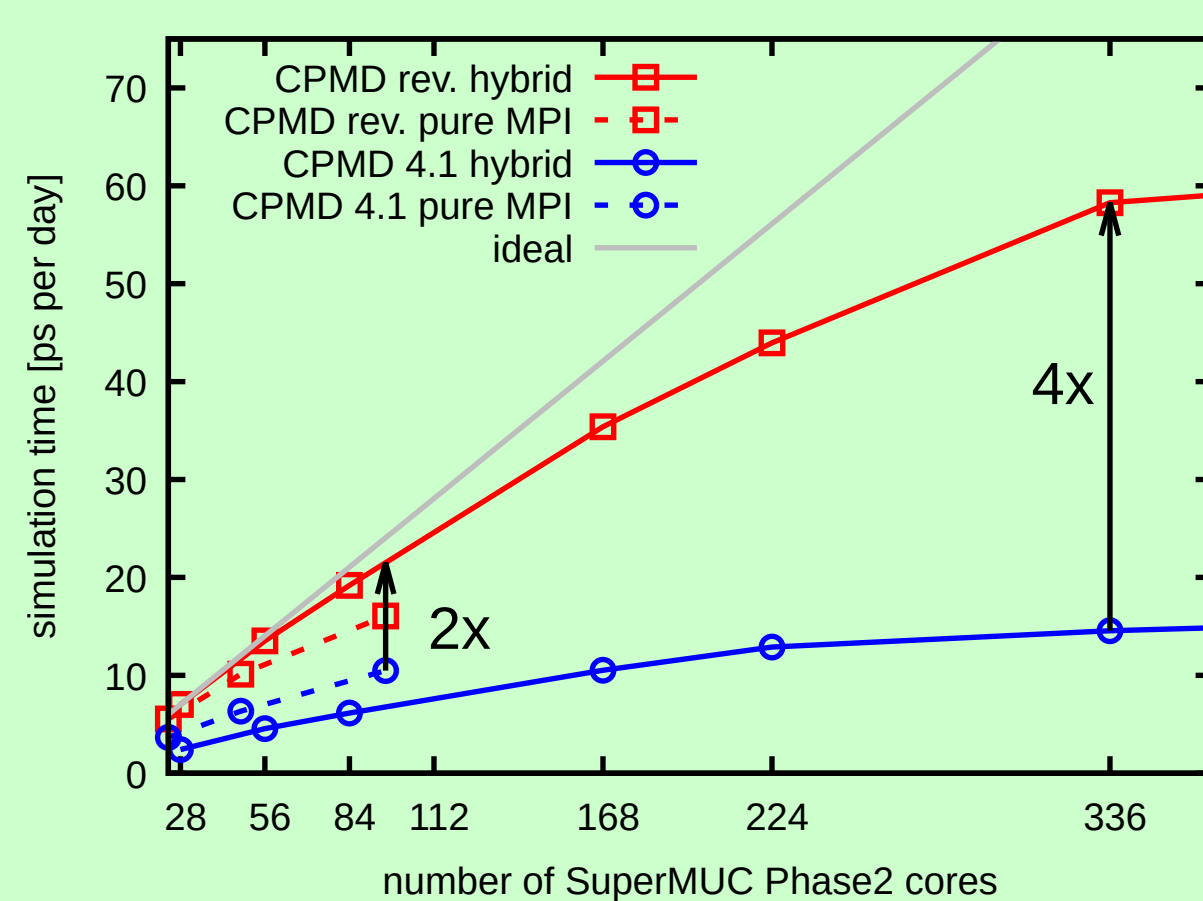
Rule of thumb

12.5 s per MD step }
1 ps per day } → 2 months for trajectory

Crucial

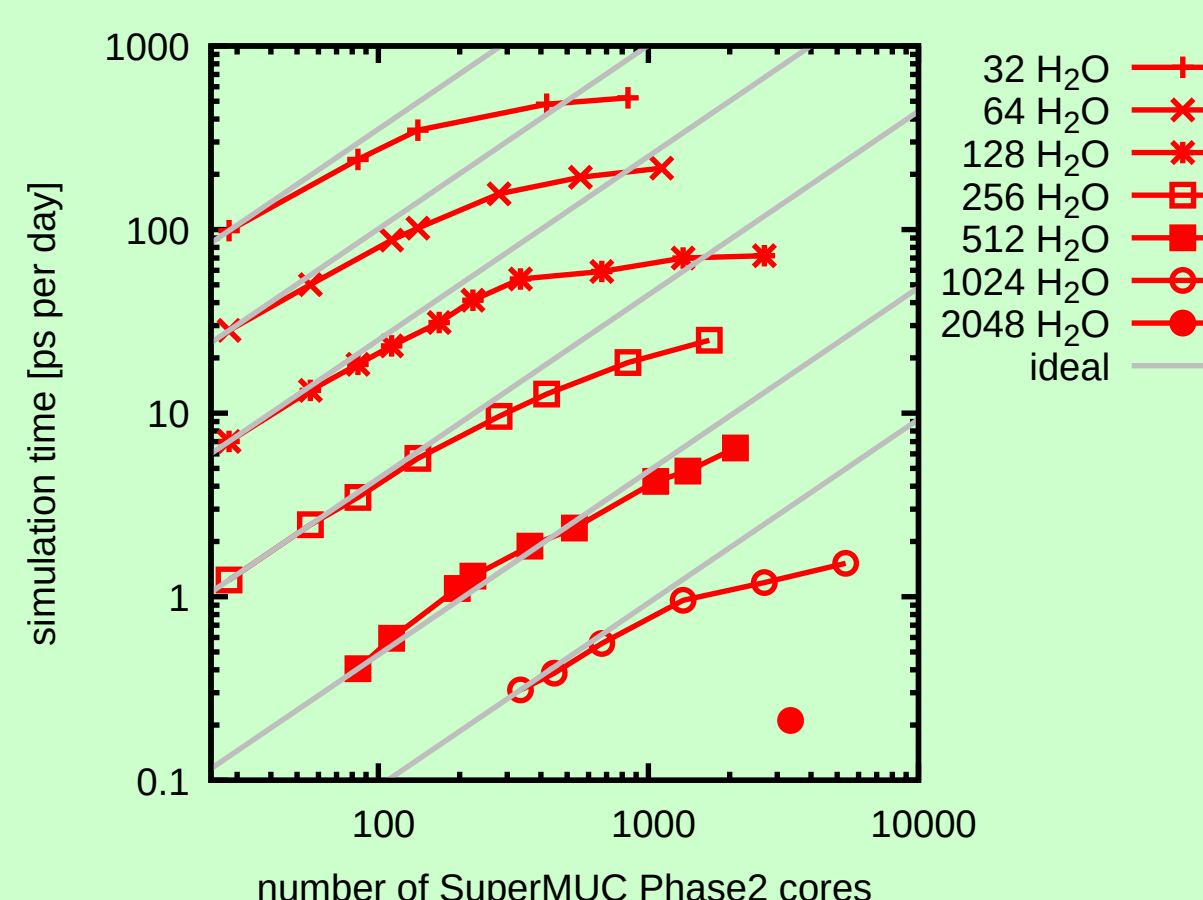
Time to solution! Strong scaling!

II Code Modernization



New Scaling

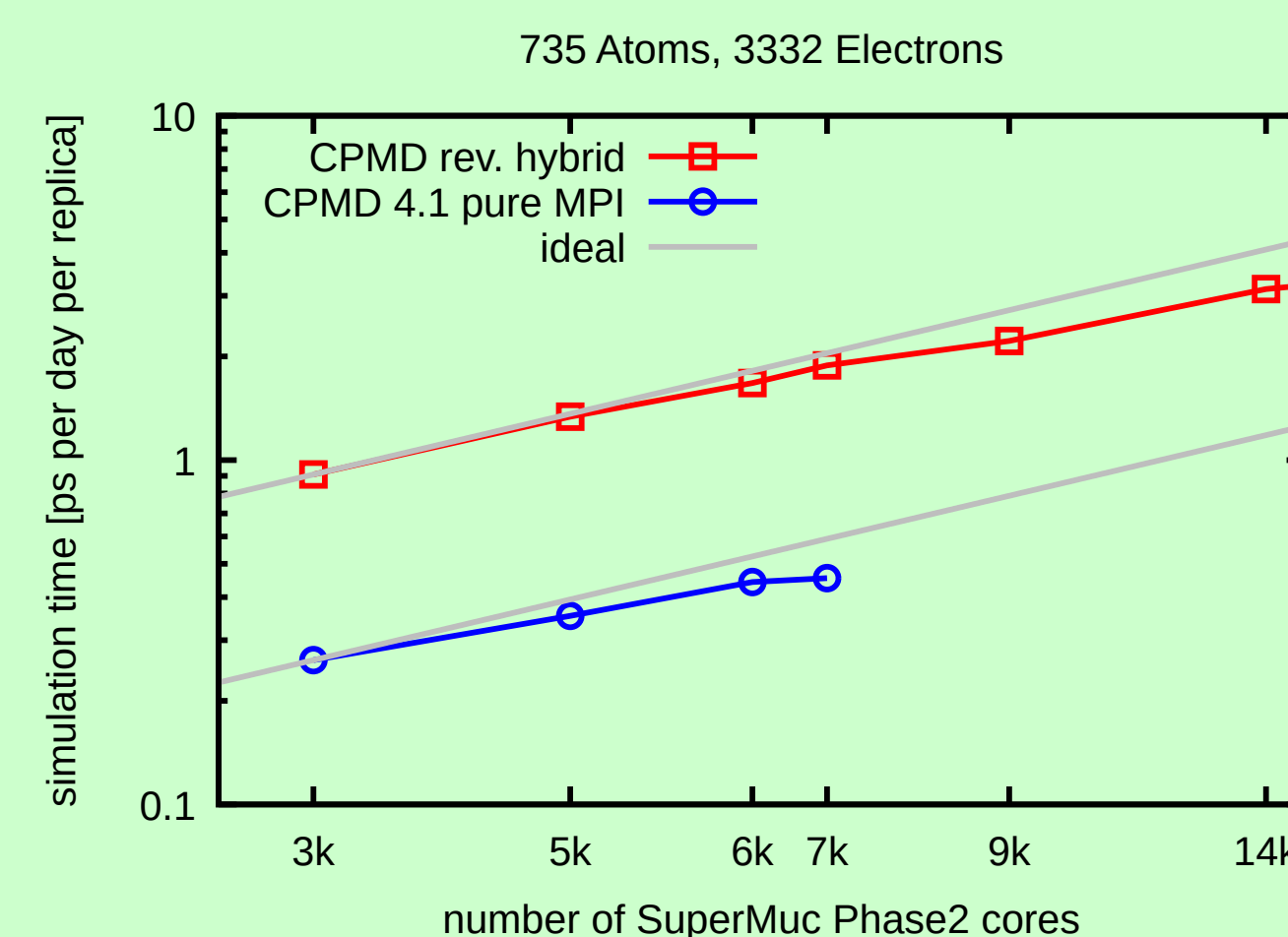
- strong scaling:
128 H₂O molecules
- scaleout:
32 - 2048 H₂O molecules



- enabling openMP
- revised communication scheme (3D-FFT)
- restructured data
- overlapping communication & computation

III HPC Application

Large Scale simulation 14000 cores Free Energy Sampling ("Umbrella Sampling")

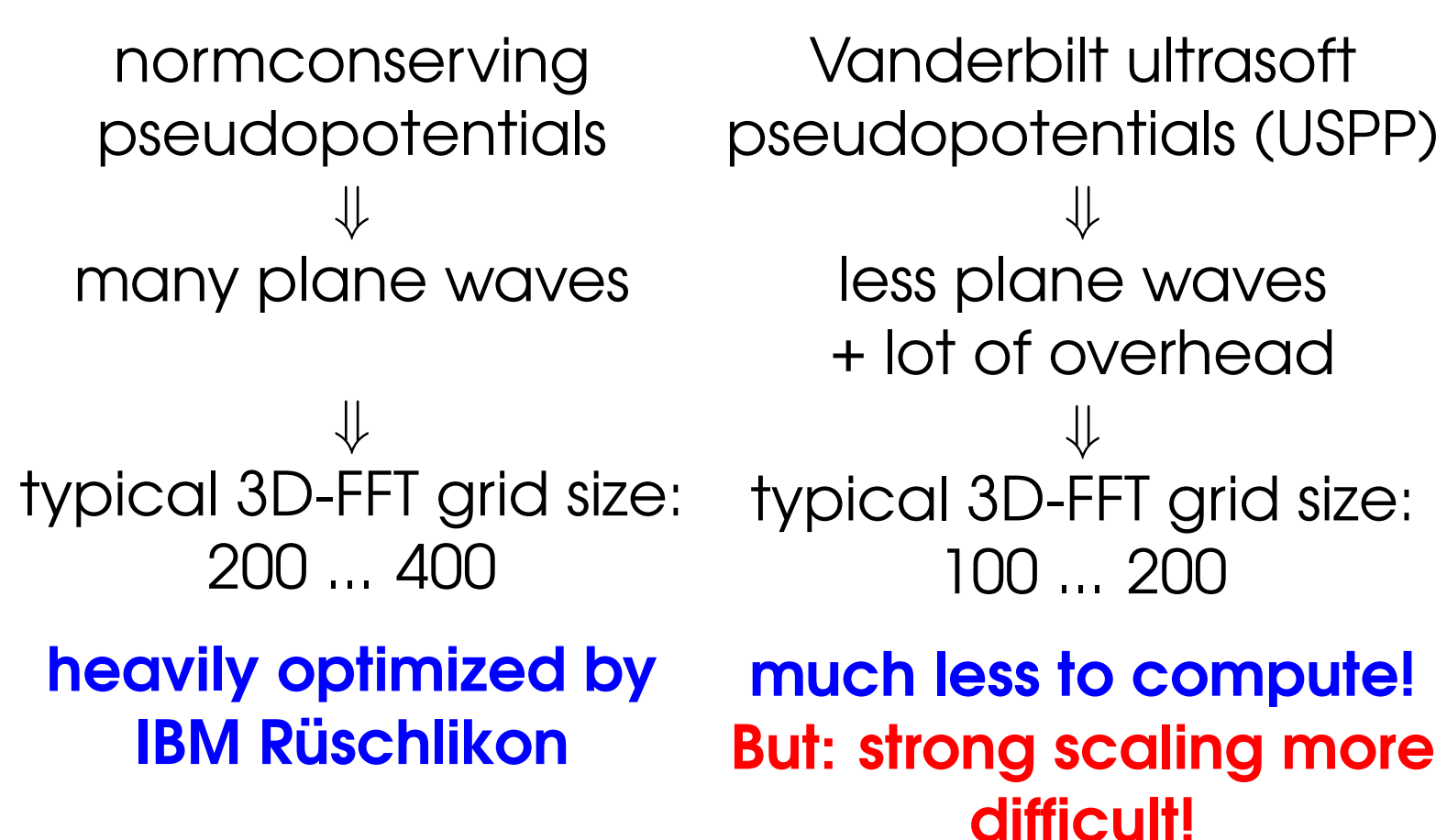


- 3332 electrons
- 735 atoms
6 species:
H: 344, O: 276,
Zn: 108, C: 4,
Cl: 2, S: 1
- 28 replicas

- Ideal scaling with loosely coupled replicas
- 7 x improved time to solution @ 2 x resource usage
9 nodes/replica => 18 nodes/replica
- Sustained single node performance:
50% peak @ 18 nodes per replica
- 12 x improved time to solution @ 5.3 x resource usage
9 nodes/replica => 48 nodes/replica

I.1 The CPMD Code

Two branches



I.2 CPMD Parallelization & scaling limitations

Primary MPI parallelization Real- & G-space Data Distribution

- 3D-FFTs
 - Distribution of real space 3d-FFT planes
 - "magic" core counts depended on the FFT grid
- Parallel Distributed Matrix Matrix Multiplications
 - a lot of small local dgemms
 - a lot of small mpi allreduce calls
 - hard to exploit peak flops
 - small messagesize
 - scaling limitation:
communication time goes up & computation time goes down

Secondary MPI parallelization: CP_GROUPS

- Replication of dataset, distribution of orbitals
- improved scalability
- frequent synchronization between groups
- overhead!

openMP: Extension to MPI

- fine grained loop parallelization
- threaded libraries
- main benefit: extending scalability by reducing MPI ranks

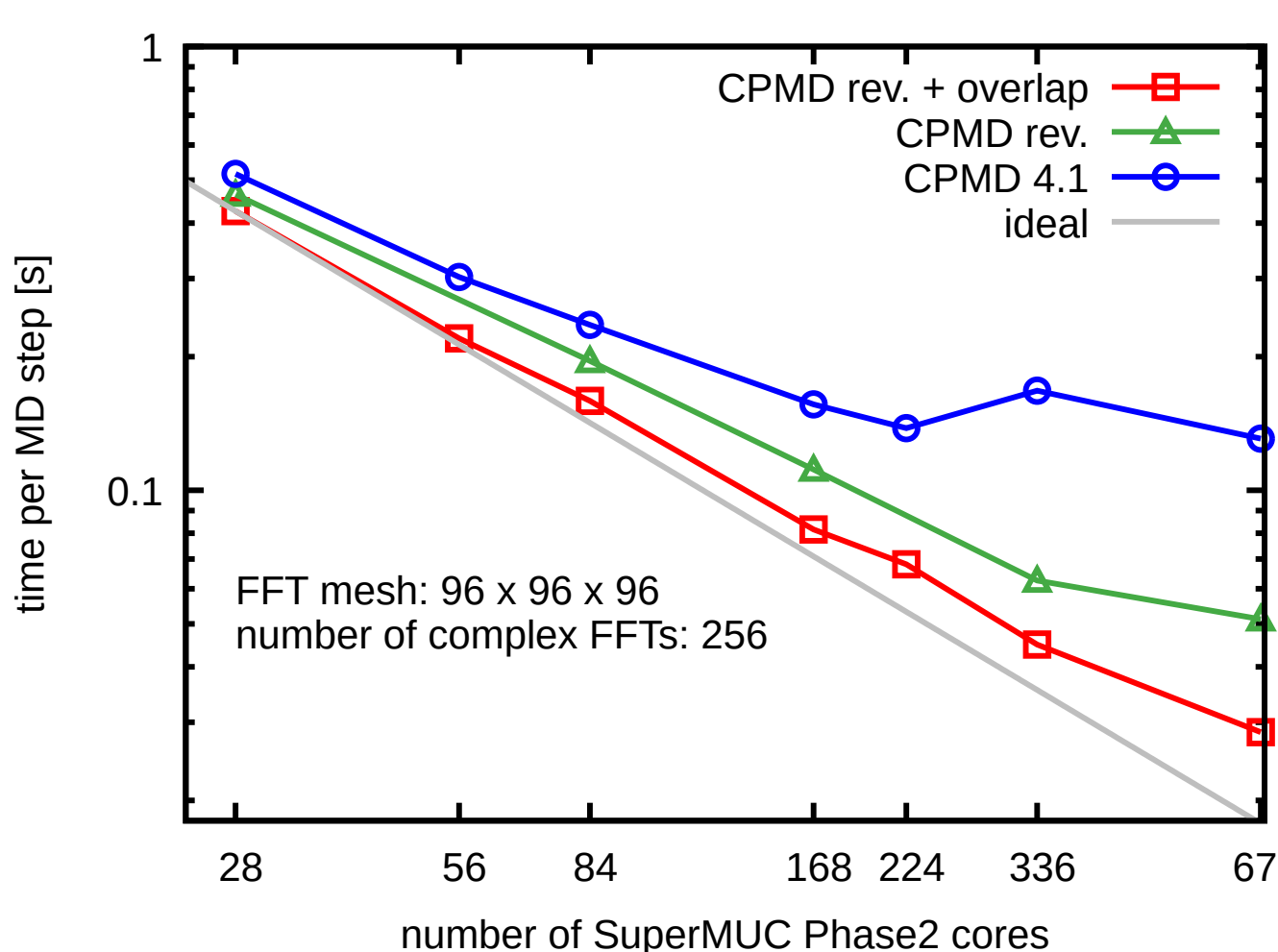
Special Subroutines for USPP

- openMP parallelization missing!
- Medium sized eigenvalue problem ((number of orbitals)²)
→ serial LAPACK routine!
- more frequent synchronization between CP_GROUPS needed | duplicated work
=> CP_GROUPS not as effective

Limited Scalability for USPP

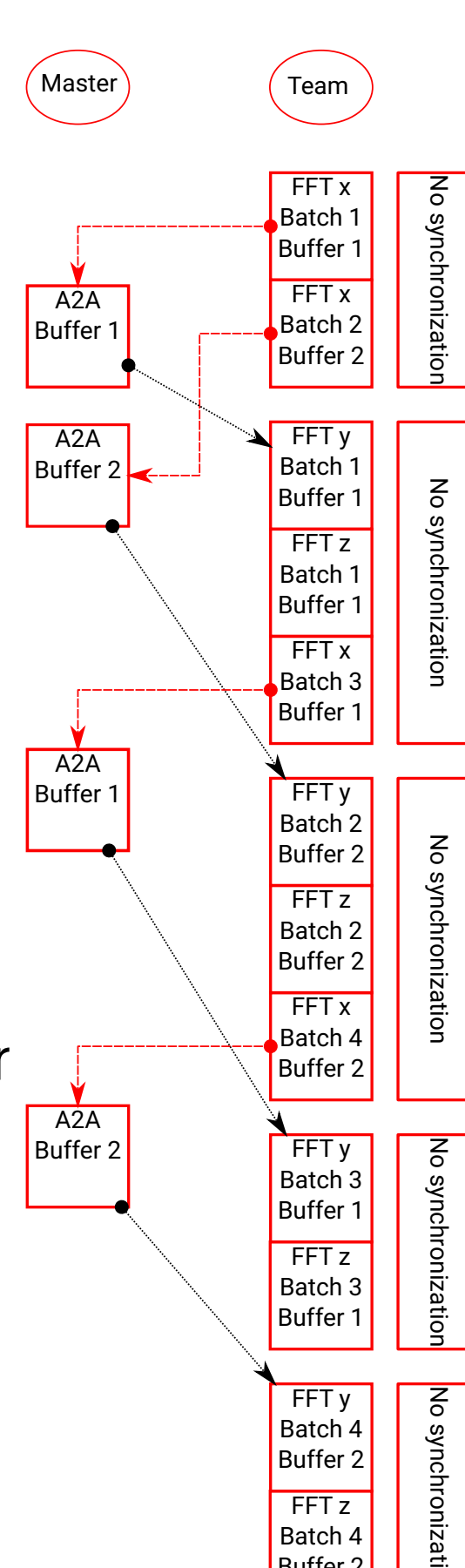
- "economical" USPP runs limited to number of core counts = real space planes
e.g. 90 (small) and 200 (large) simulations
- secondary parallelization not implemented | not effective!
- next gen super computers: scaling limited to 2-4 Nodes!
- inefficient on many core systems

II.1 Restructred 3D-FFT driver

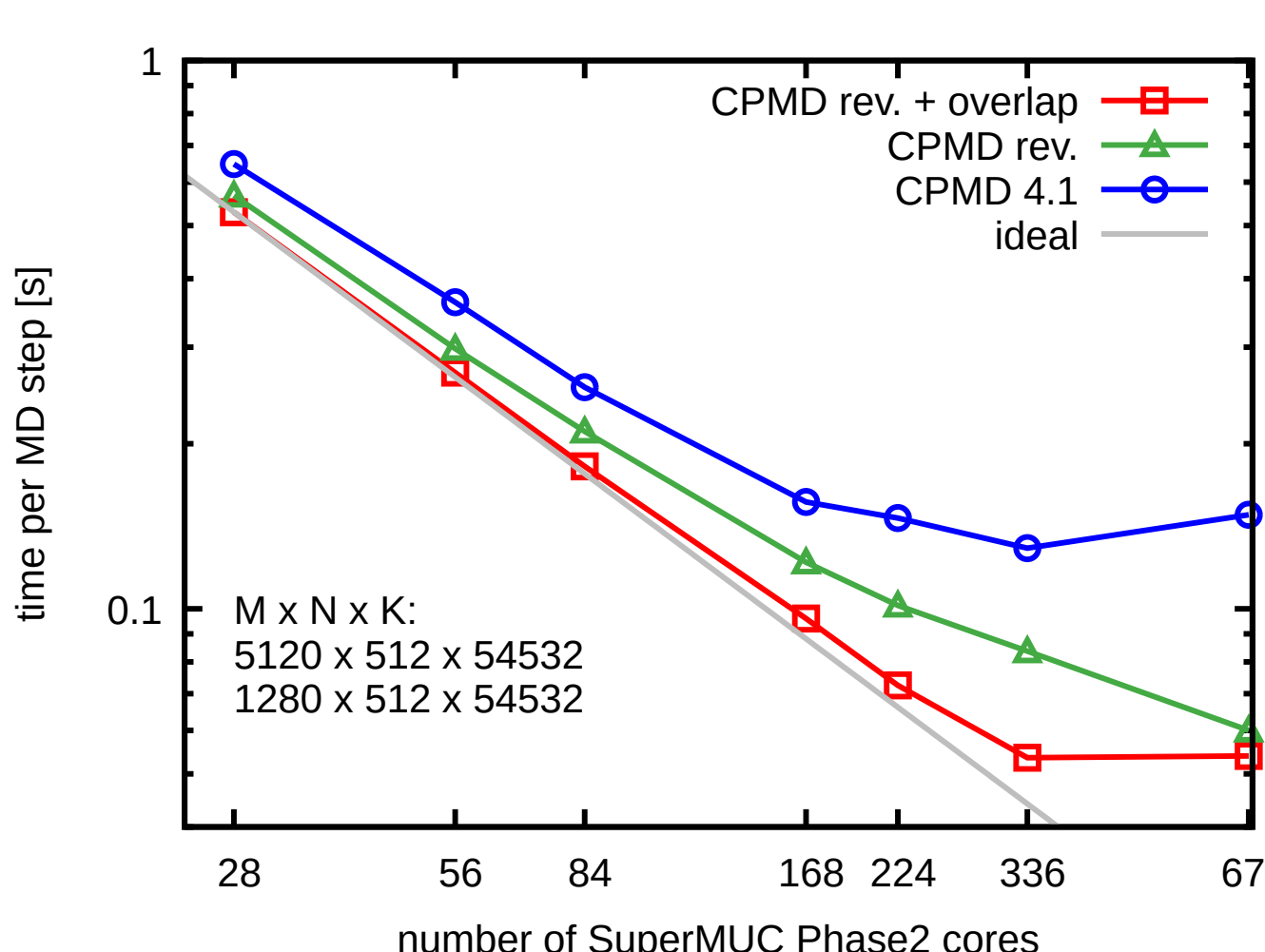


Batched FFT & overlapping com/comp

- delayed communication step until a user defined message size is reached
→ no overhead!
- overlap communication and computation (openMP)
- CP_GROUPS still possible on top
- autotuning incoming!



II.2 Restructred Distributed Matrix Matrix Multiply



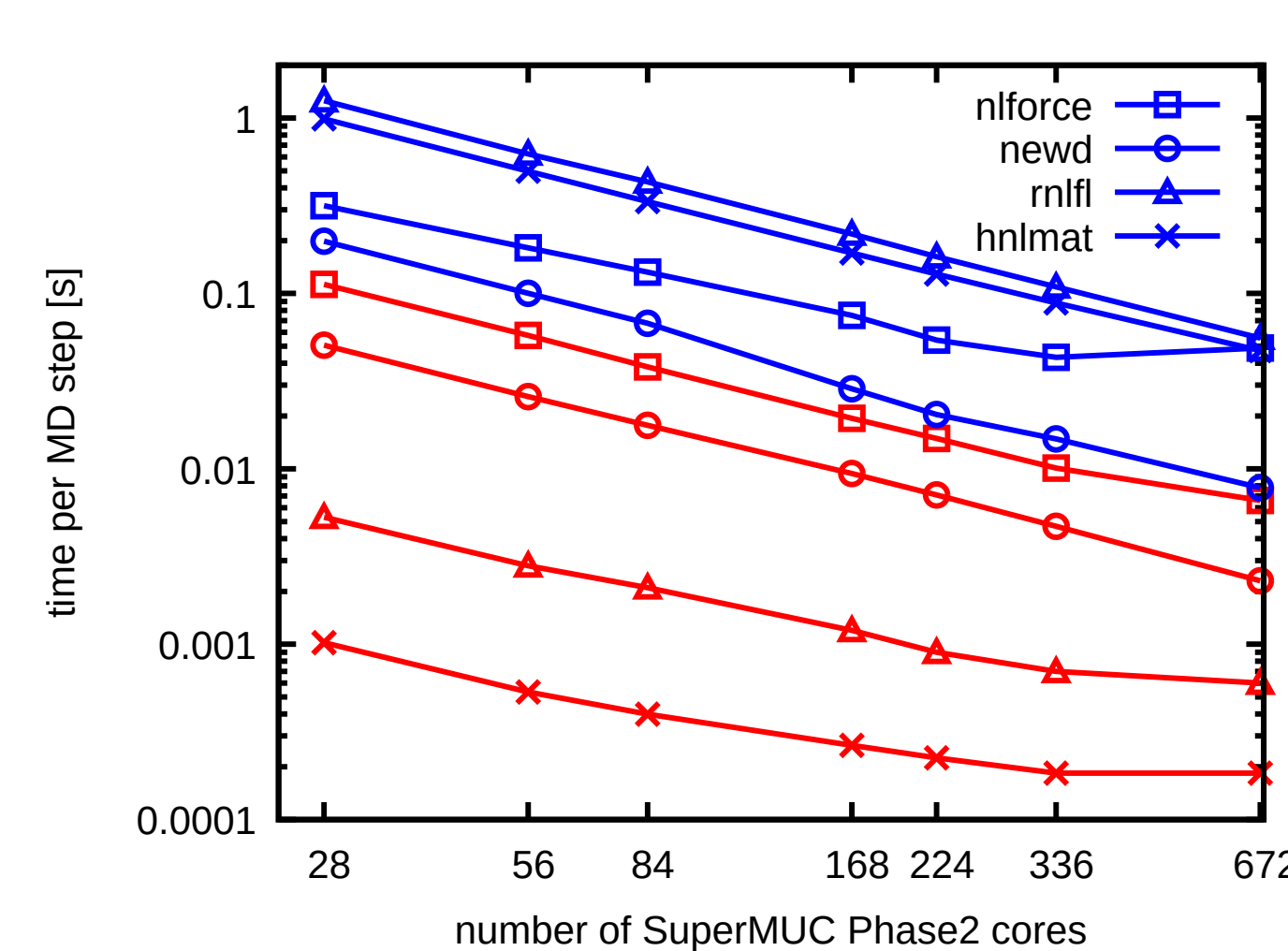
Pipeline Algorithm

- restructuring the data for a single large dgemm → one large mpi allreduce call
- break large matrix into smaller parts
- openMP master thread to communicate, others compute
- automatically, on the fly tuneable matrix sizes
- high flop rate and good overlap
- breakdown if communication » computation

Planned...avoid first mpi allreduce

- only needed to parallelize two follow up routines:
rnlfor & rnlfi
- heavy optimization of follow up routines
- remove inverse scaling mpi allreduce & usage of optimized but not parallelized rnlfor & rnlfi routines
- work in progress, first results: benefits even at the node level
- reduction of mpi allreduce by around 80%

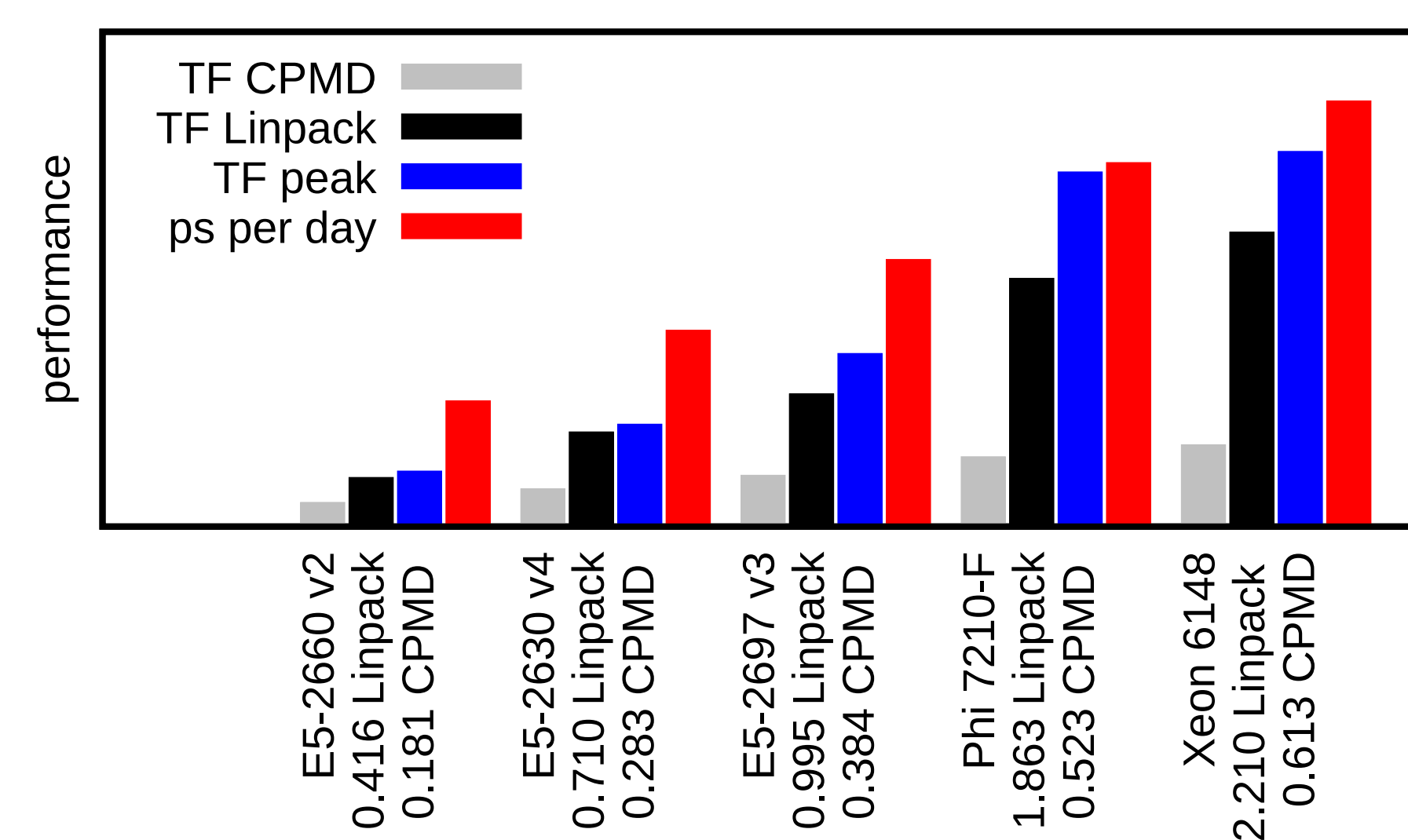
II.3 USPP specific routines



- enabling openMP (newd, hnlmat, rnlfi)
- optimizing and overlapping communication (nlforce)
- rearranging data & transforming loops to (small) dgemms (hnlmat, rnlfi)
- replace dspev with ELPA (not shown)
– still huge scaling problem
– evaluation of different eigenvalue solvers

II.4 Present & Upcomming Architectures

128 H₂O Molecules: Single Node Performance



- main development @ HSW & BDW
- promising out of the box performance on KNL & SKL
- scaleout on KNL hard (extremely high node to node latency)

Acknowledgments

- scaleout benchmarks, stability tests @ Leibnitz Supercomputings Centre of the Bavarian Academy of Sciences and Humanities (LRZ)
- code development, performance measurements @ Regional Computing Center (RRZE)
- funding KONWIHR @ RRZE
- fruitful discussions HPC Services Group @ FAU Erlangen, esp. Georg Hager