Multi-scale molecular simulations of biological systems: Parallelization of RAPTOR for Blue Gene/Q

Adrian W. Lange
MiraCon
ALCF
03/06/13

Multi-state empirical valence bond

Goal:

- To simulate chemically reactive biological systems, particularly proton transport through water and biomolecules

Problem:

- Atomistic force fields typically ignore reactivity

Compromise:

- Approximate reactivity with MS-EVB

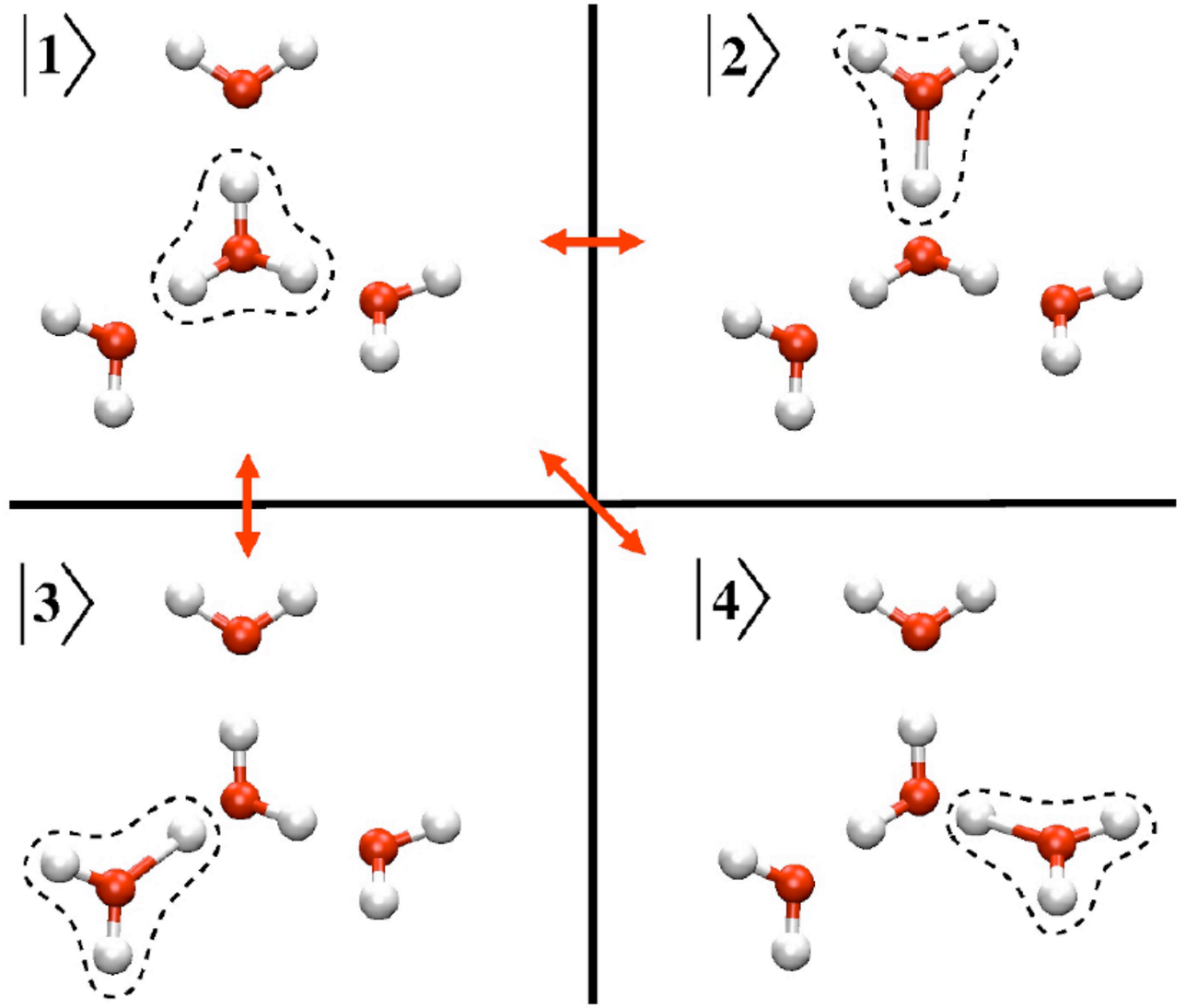
$$|\Psi\rangle = \sum_I c_I \, |\psi_I\rangle$$

$$\mathbf{Hc} = E\mathbf{c}$$

$$E^{\mathrm{MS-EVB}} = \min(E)$$

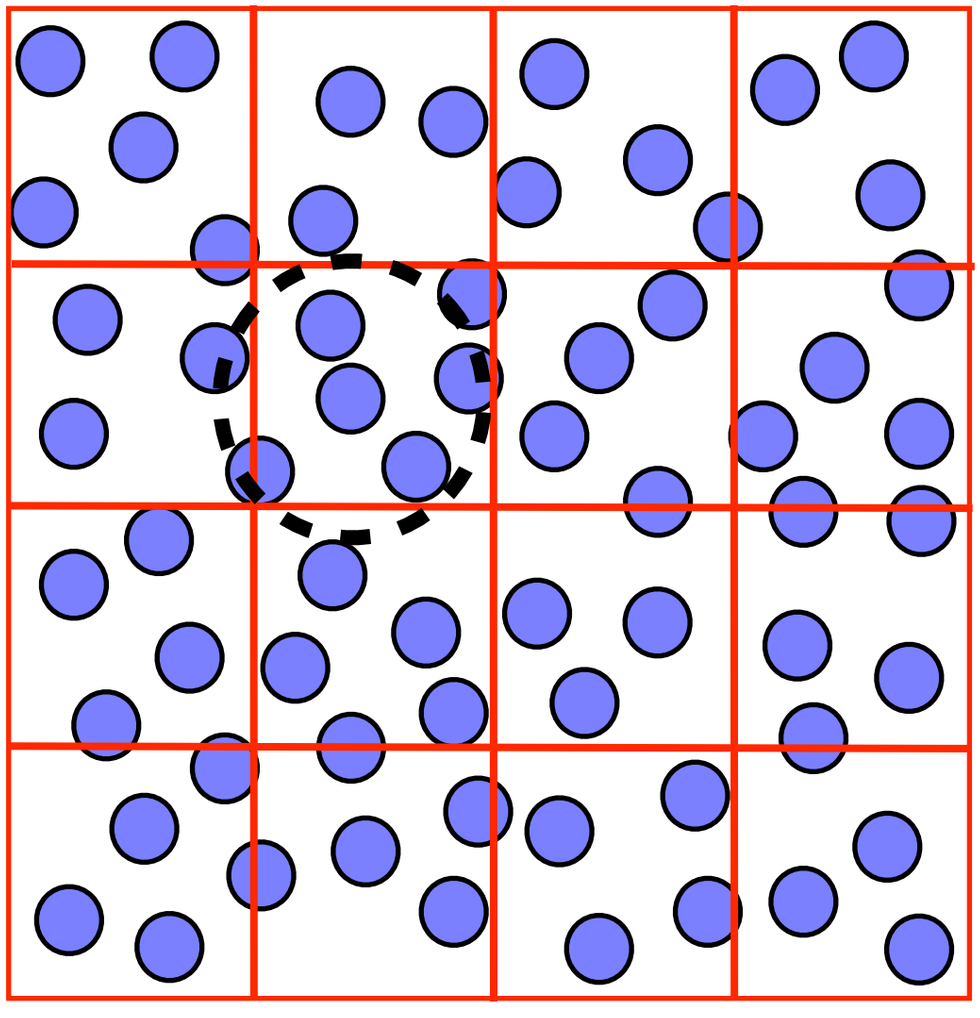
$H_{II} =$ Energy of state I

$H_{IJ} =$ Coupling b/w I and J



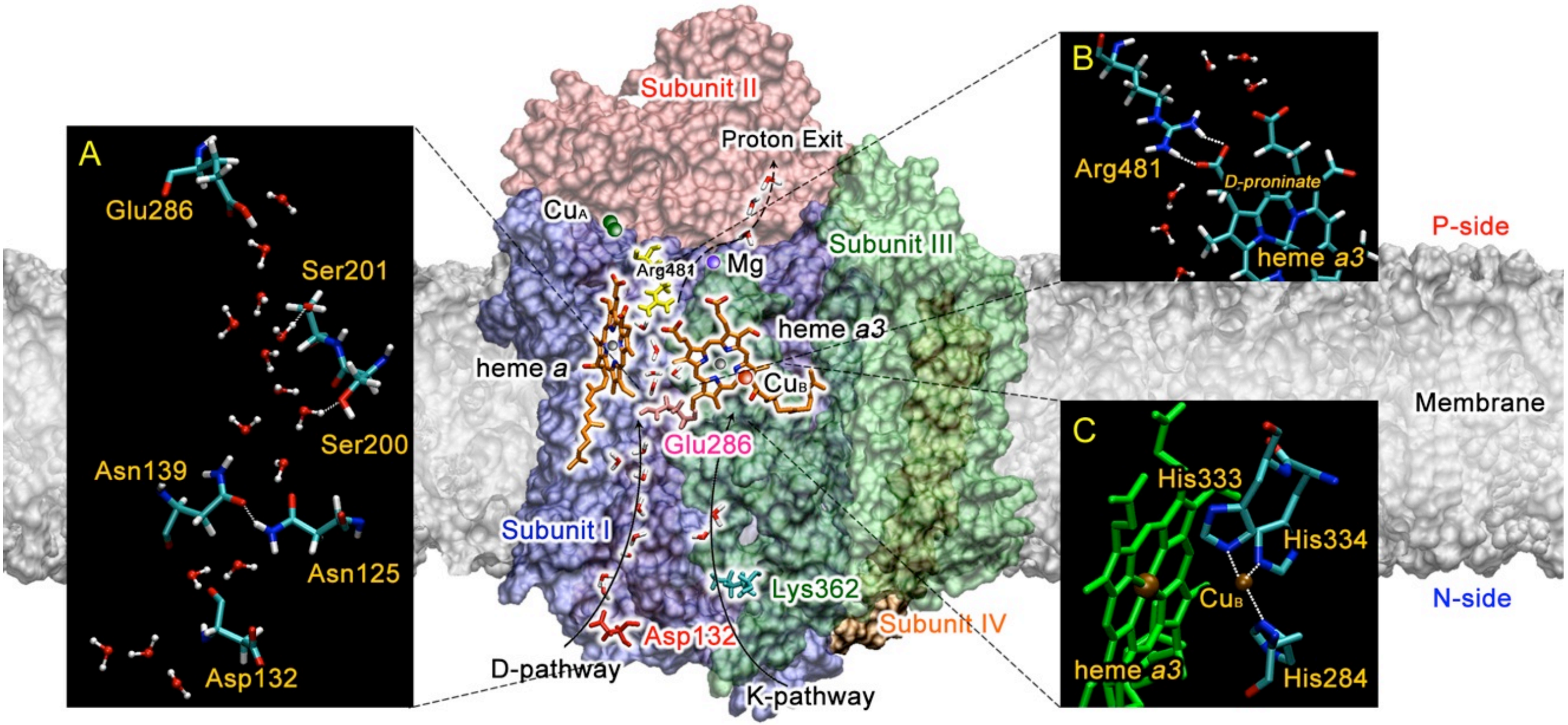
Implementation: RAPTOR in LAMMPS

- Rapid Approach to Proton Transport and Other Reactions (RAPTOR)
- Add-on package to LAMMPS software (C++ and mainly MPI)
- Primary approach to parallelism:
  - 3D domain decomposition with MPI
  - Short-range particle-particle interactions in neighbor list
  - Long-range electrostatics handled with periodic boundary conditions and FFT



Target Application

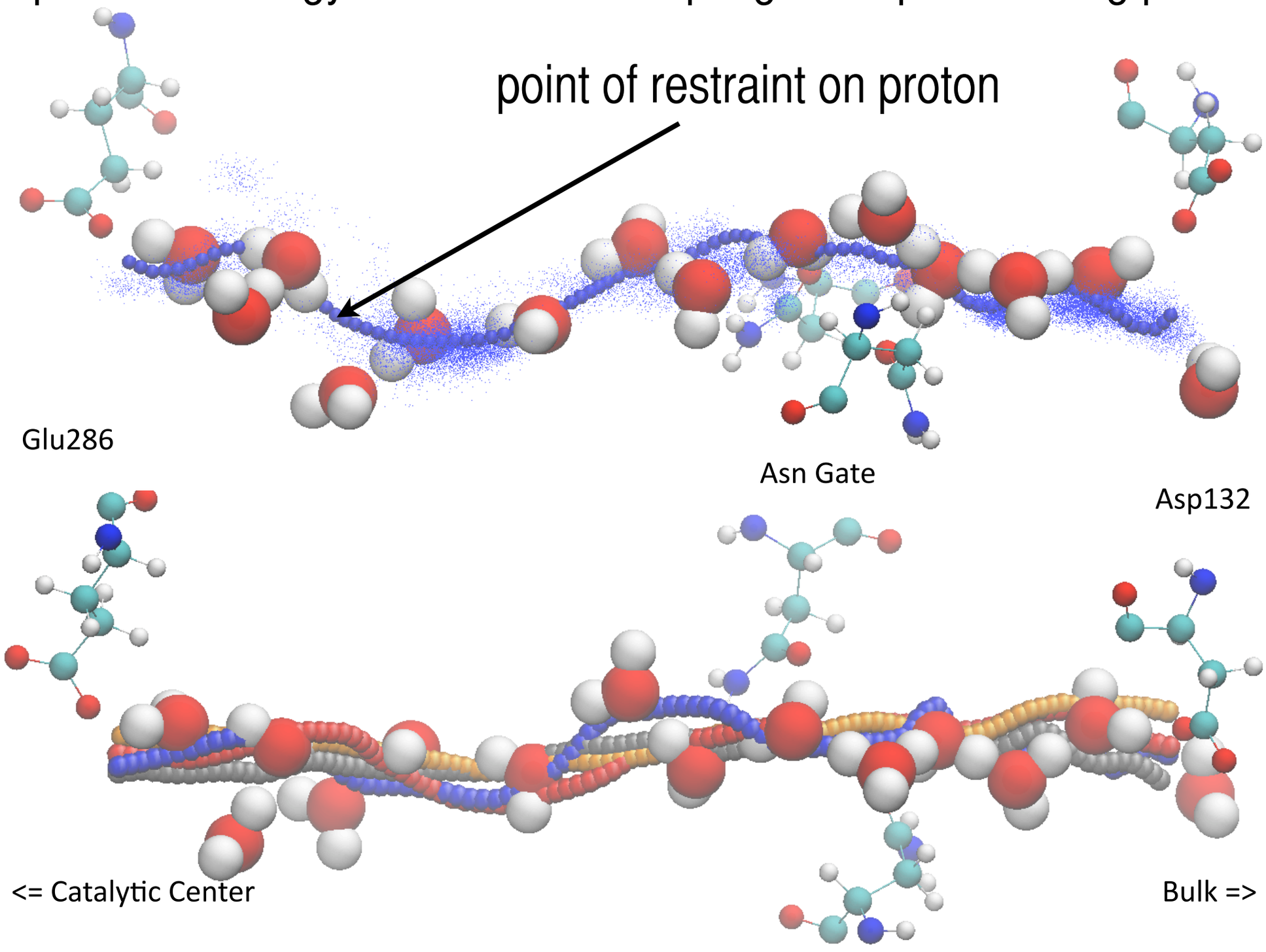
Mapping proton Potential of Mean Force in CcO with MS-EVB
- CcO = 159k atoms, 1 reactive excess proton



Target Application

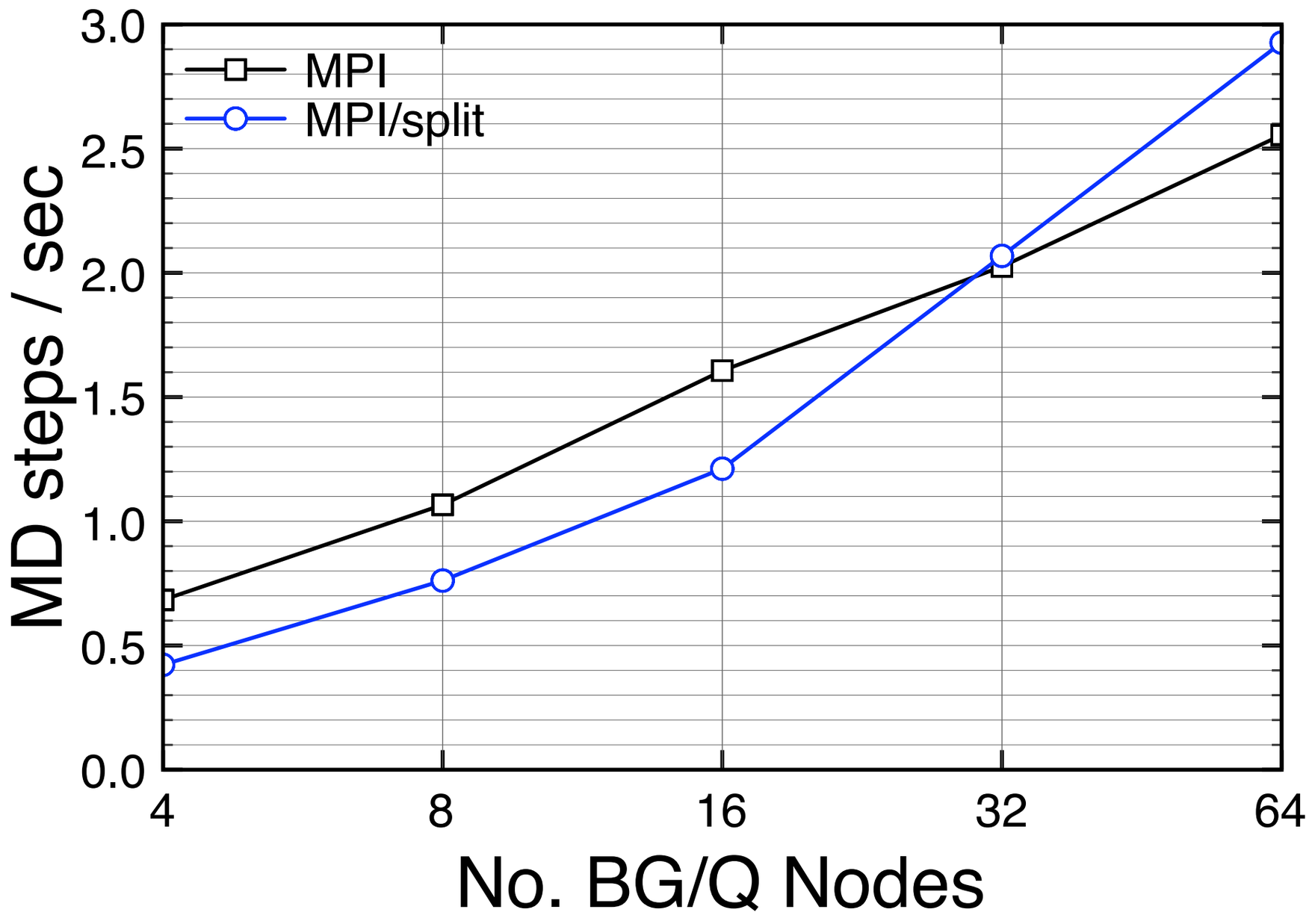
Mapping proton Potential of Mean Force in CcO with MS-EVB

- CcO = 159k atoms, 1 reactive excess proton
- Compute free energy via umbrella sampling at 96 points along path



Scaling as of April 2012

Single window from CcO path
MPI/split:
- FFTs computed concurrently on subpartition
- 3:1 r:k ratio, c16 mode



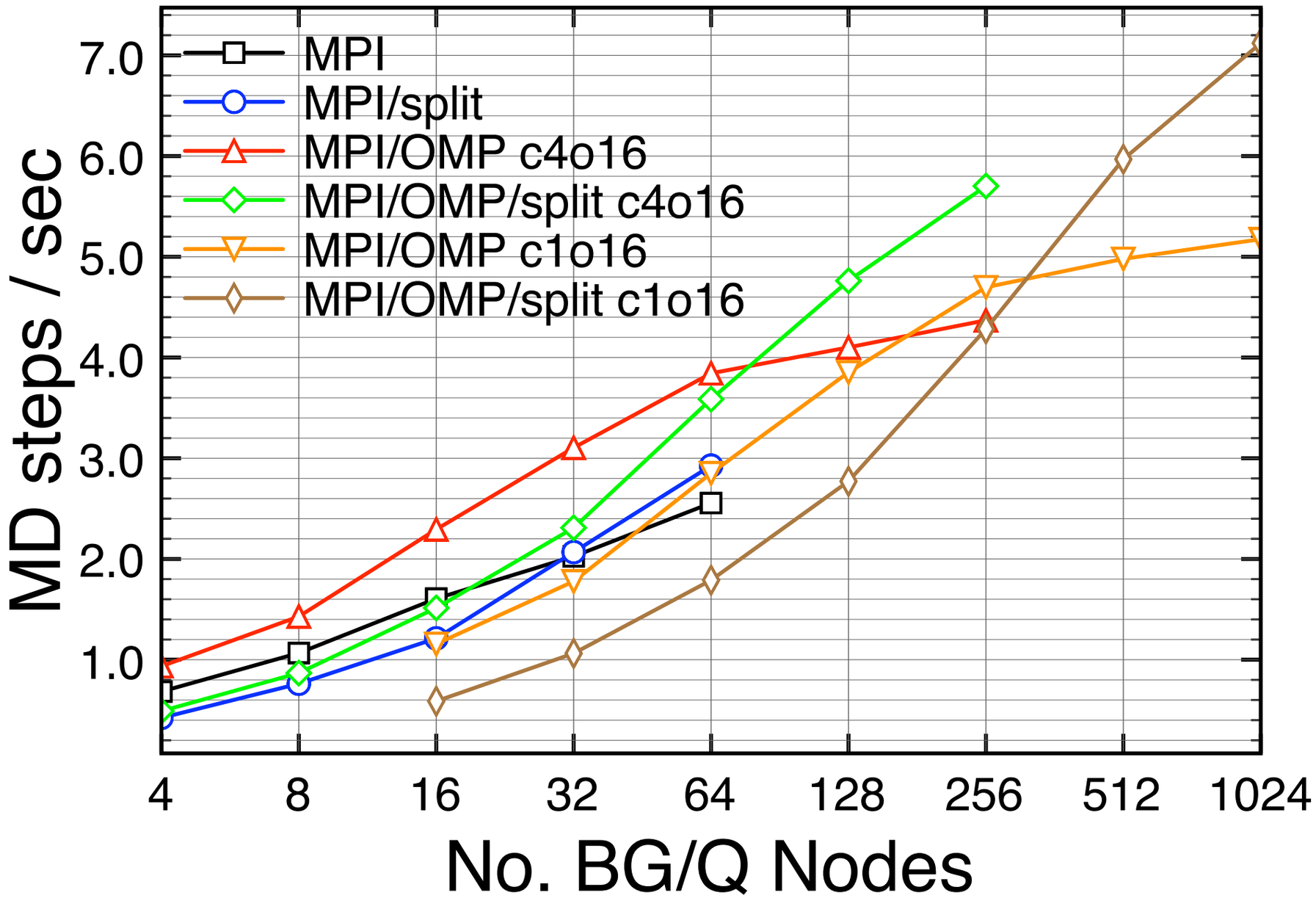
Improving performance

- Profiled code with TAU to identify bottlenecks
- Simple serial optimizations (unrolling, templates, hoisting ifs, etc.)
- QPX-ified handful of compute intensive loops using vector intrinsics
  - Pairwise kernel:
    - If statements/look-up table in loop makes computation somewhat irregular
    - Eliminate look-up table (slower, but now SIMD-izable)
    - Buffer/flush approach:
      1. If compute pair, store data in vector4double buffers, n++
      2. If n==4, flush buffer with vector intrinsic computes, n=0
      - 25% speedup vs. non-SIMD, look-up table approach
      - 50% speedup vs. non-SIMD, no look-up table approach
- Multithreading with OpenMP:
  - Leverage LAMMPS add-on package USER-OMP
  - Threaded several RAPTOR routines
  - Loop level parallelism in force kernels (*i.e.*, force decomposition)
  - Concurrency with some threaded calls to FFT work

Improving performance

Scaling with c1o16 mode (1 MPI rank/node, 16 OMP threads/rank)

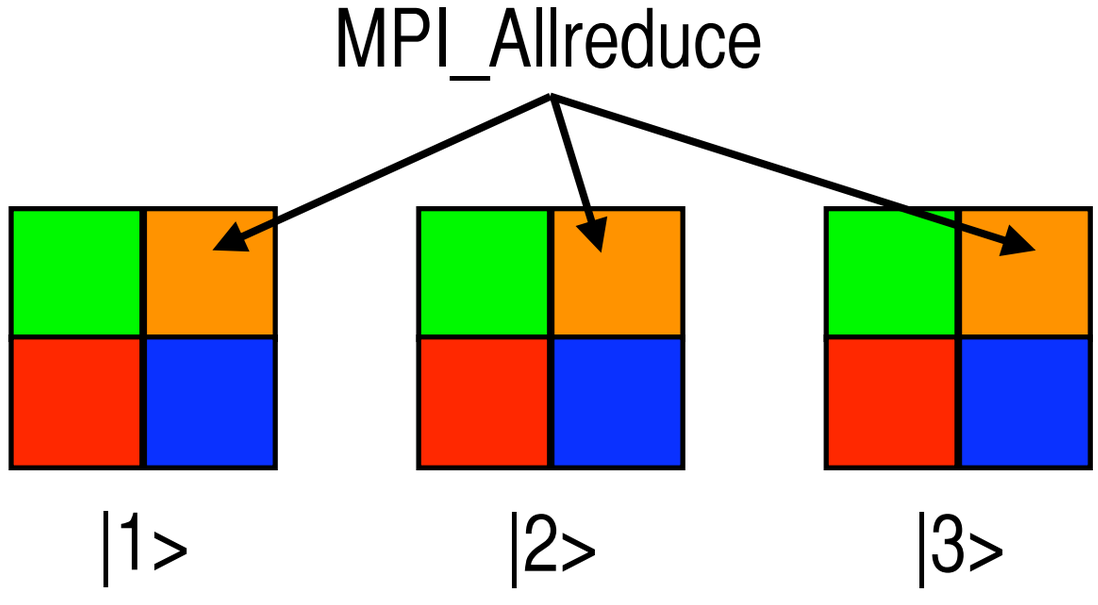
- Not always best to hyperthread the Quad-FPU with threads
- FFTW library benefits from QPX slightly (compiler level)
- Threading staves off the domain decomposition limit to higher node counts



Improving scaling with state decomposition

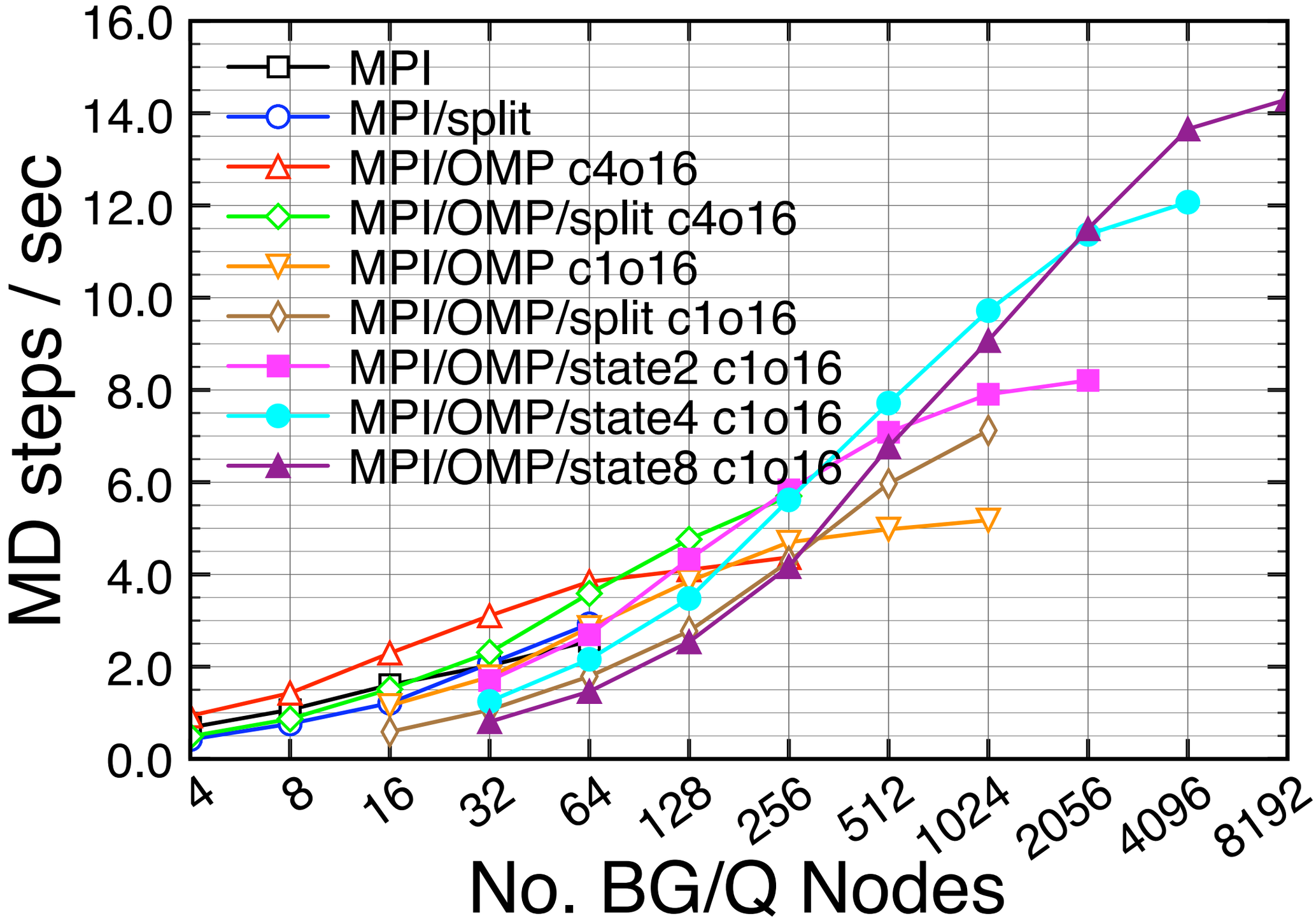
- CcO example has 14–16 EVB states
- Prior examples filling in Hamiltonian matrix elements in serial
- LAMMPS can run in “partitioned” mode using MPI subcommunicators
- State decomposition:
  - Each LAMMPS partition works on different EVB state(s)
  - MPI subcommunicators groups for each spatial domain to take advantage of using MPI collectives during reductions

$$\mathbf{H} = \begin{pmatrix} H & H & H \\ H & H & H \\ H & H & H \end{pmatrix}$$



Improving scaling with state decomposition

Scaling further...

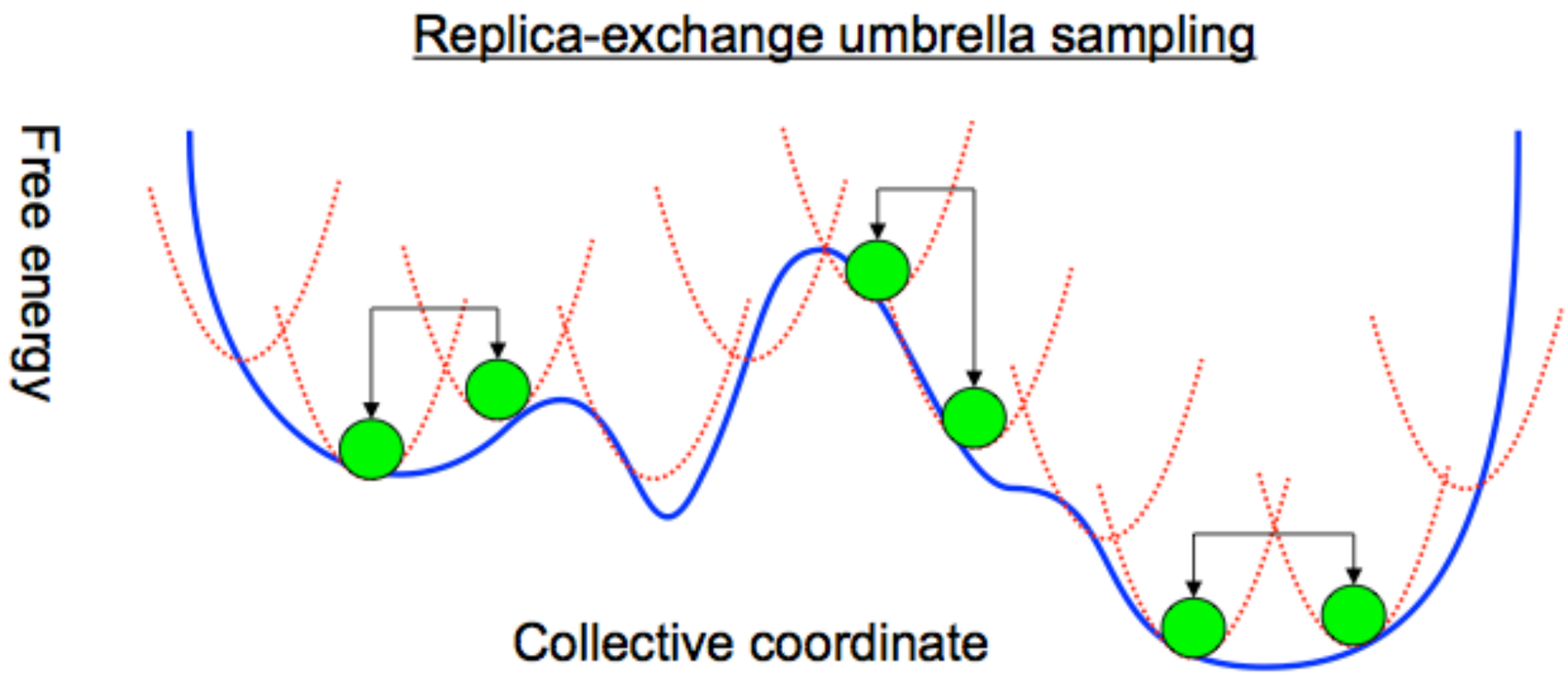


Replica Exchange Umbrella Sampling

- Sampling individual windows of path may converge slowly
- Enhance with REUS:
  - Periodically attempt to swap restraint between neighboring windows
  - CPU time to solution reduced compared to uncoupled

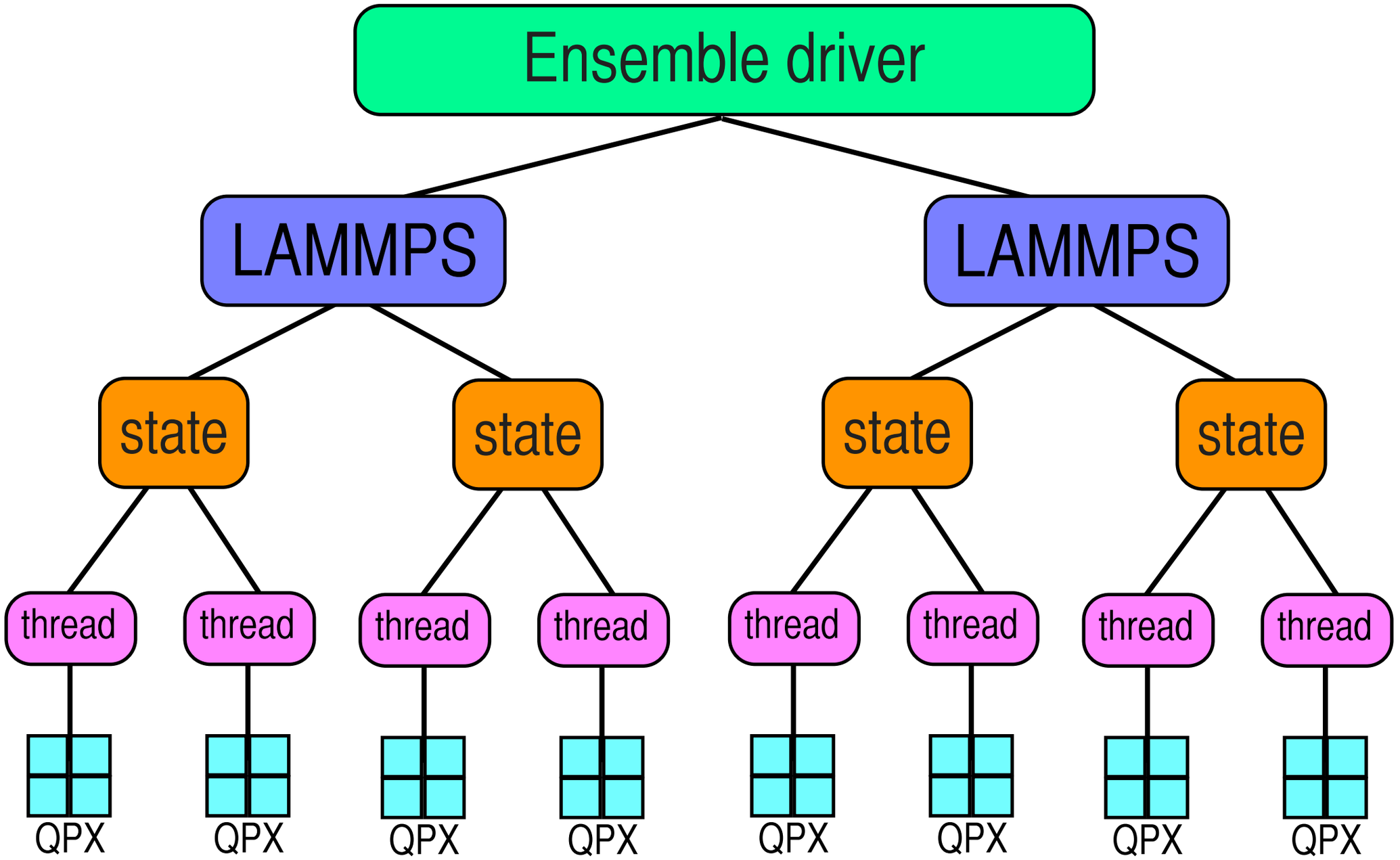
$$P_{IJ} = \min\left[1, \exp(-\beta \Delta E_{IJ})\right]$$

$$\Delta E_{IJ} = (E_I^{kJ} + E_J^{kI}) - (E_I^{kI} + E_J^{kJ})$$



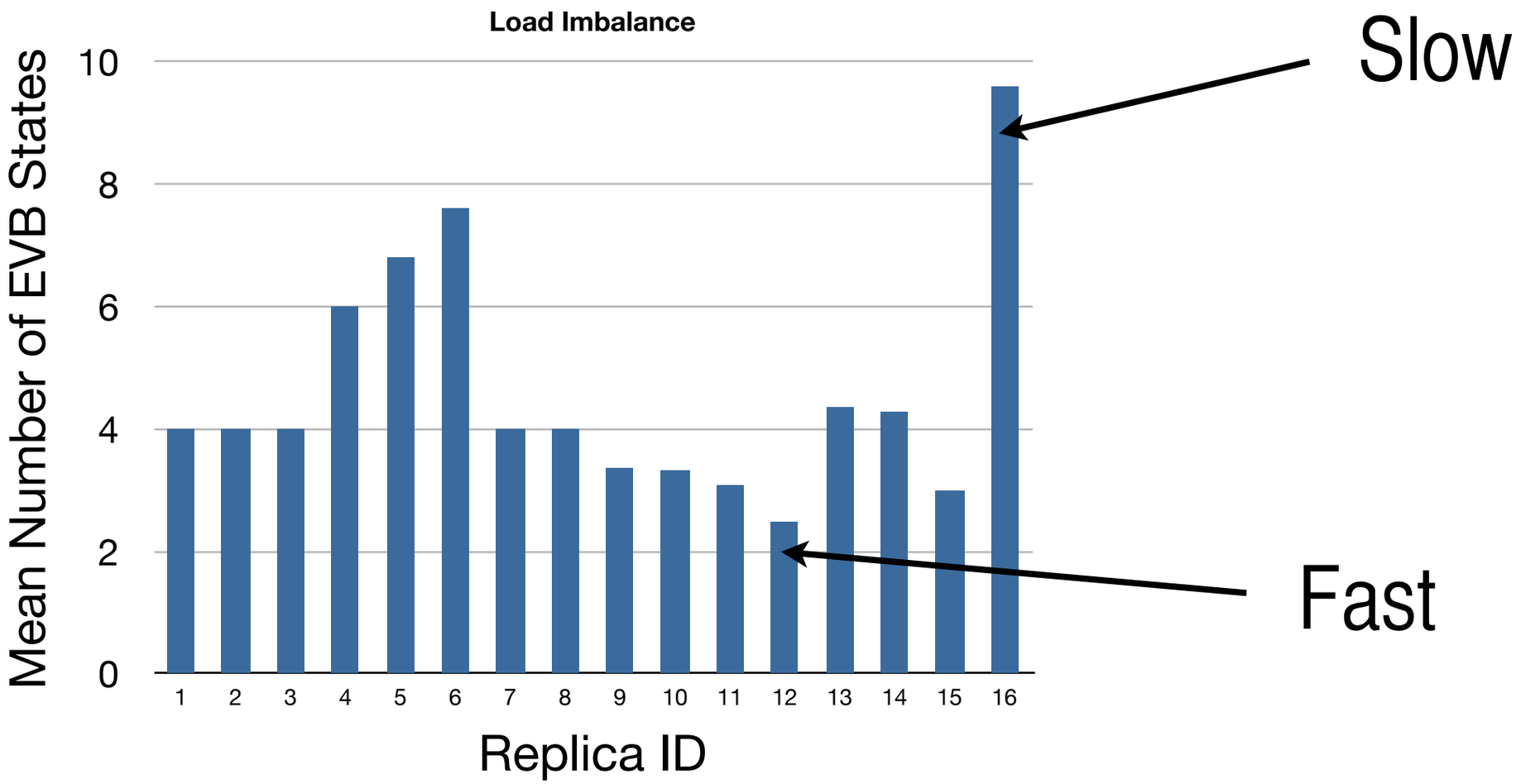
LAMMPS ensembles

- Grown out of collaboration with Jeff Hammond (ALCF)
- Each subcommunicator creates own instance of LAMMPS
- Replica Exchange between subcomms



LAMMPS ensembles: Load balance

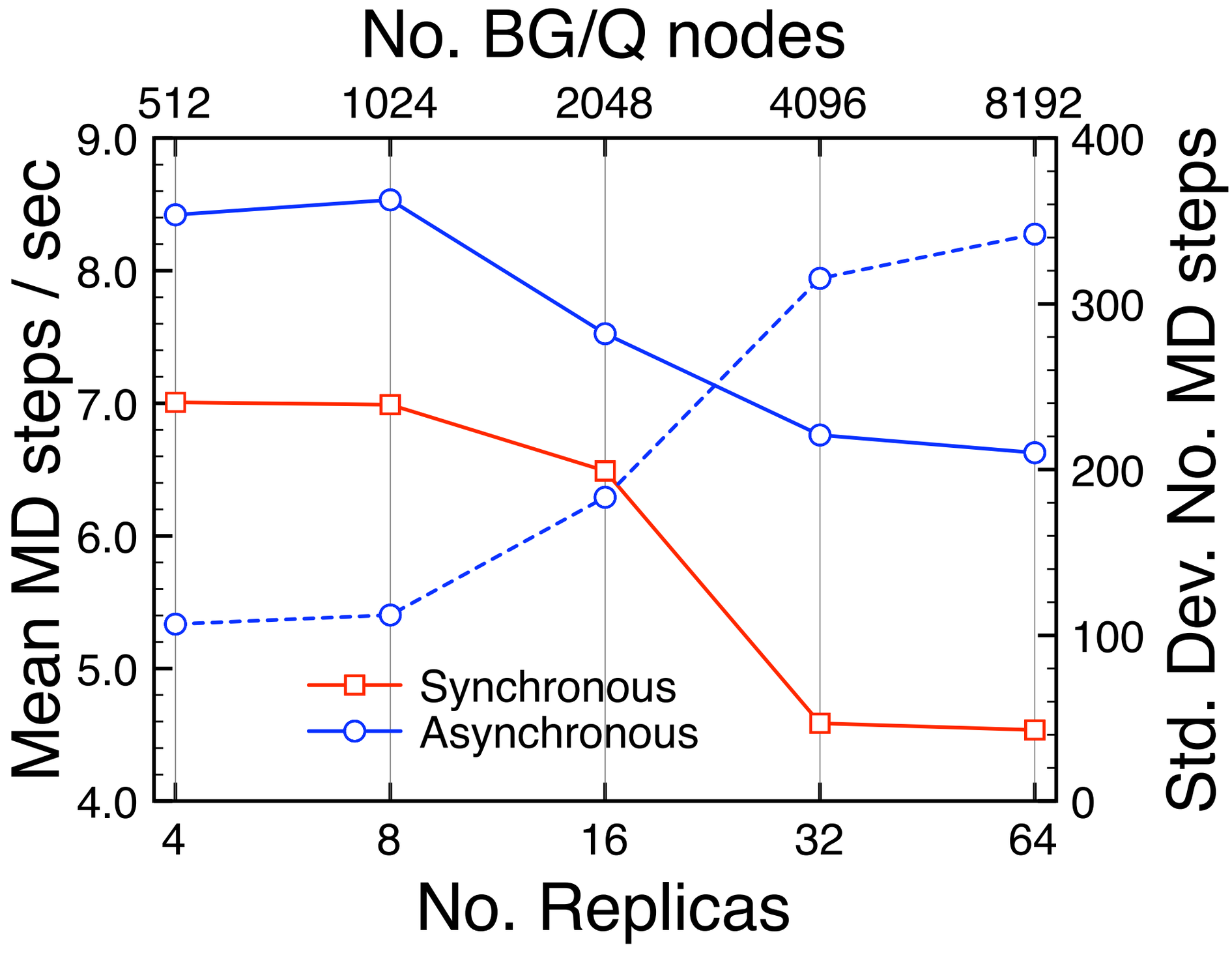
- Replicas must synchronize after N steps for exchange



- Static load balance: assign more procs to slow replicas
  - Number of states change during simulation
- Dynamic load balance: asynchronous MD runs
  - Listener/broadcaster master replica
  - Replicas continue running after N steps until all signaled

REUS example runs: Weak scaling

CcO system, c1o16 mode, 2 state partitions/replica, 128 nodes/replica



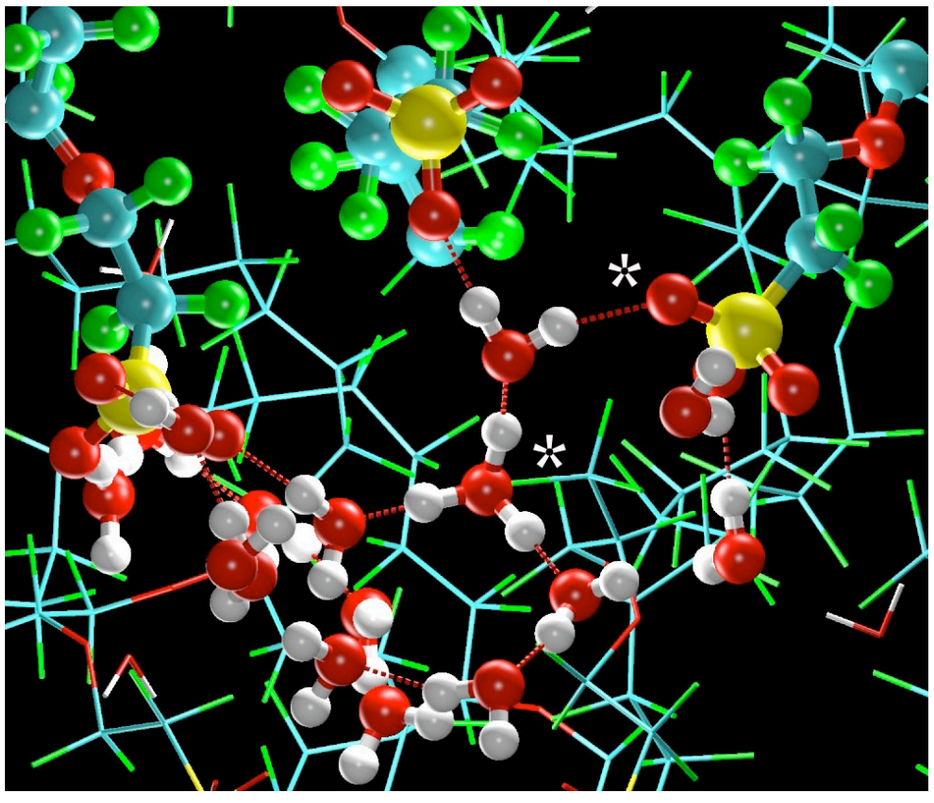
The whole shebang

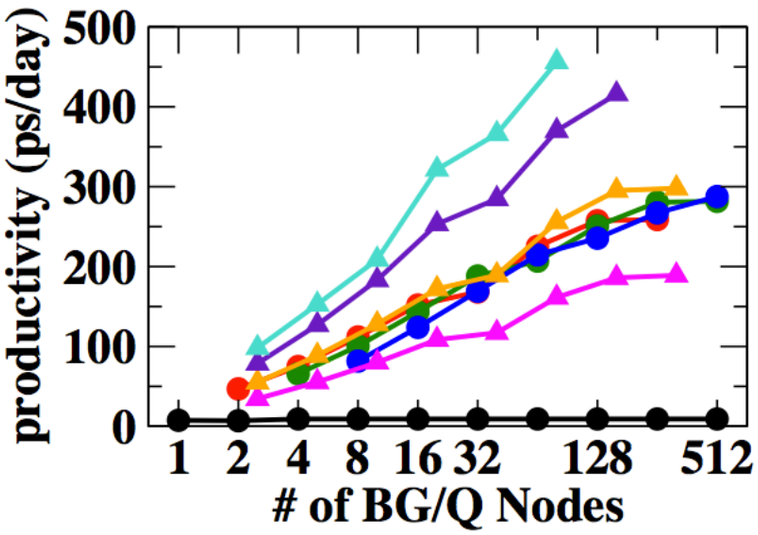
- All 96 umbrella sampling windows
- c1o16 mode, 2 state decomp, asynch dyn load balance

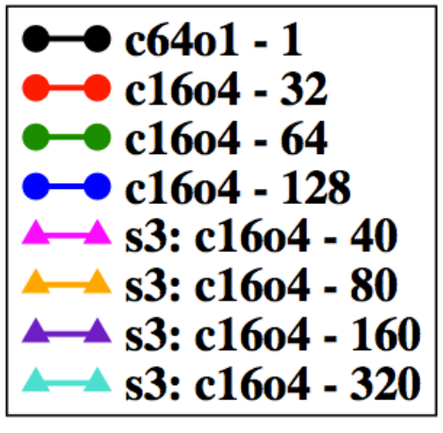
Nodes	Mean steps / sec	Parallel Efficiency	Total steps / sec
3072	2.49	1.00	238.9
6144	4.33	0.85	416.1
12288	7.08	0.67	679.8
24576	10.43	0.50	1001.1

Conclusion and future directions

- Successful parallelization of RAPTOR for Mira
- Enabling interesting simulations (*running right now!*)
- Future directions:
  - Many reactive proton systems (e.g., fuel cell simulations) [Chris Knight]
  - Parallelize across protons







Acknowledgements

<u>Thanks to people involved:</u>
Prof. Gregory Voth (UoC)
Christopher Knight (ANL)
Gard Nelson (UoC)
Yuxing Peng (UoC)
Jeff Hammond (ALCF)
Luke Westby (Notre Dame)

<u>Funding:</u>

U.S. DEPARTMENT OF ENERGY

Argonne NATIONAL LABORATORY

ALCF Early Science Program

The end