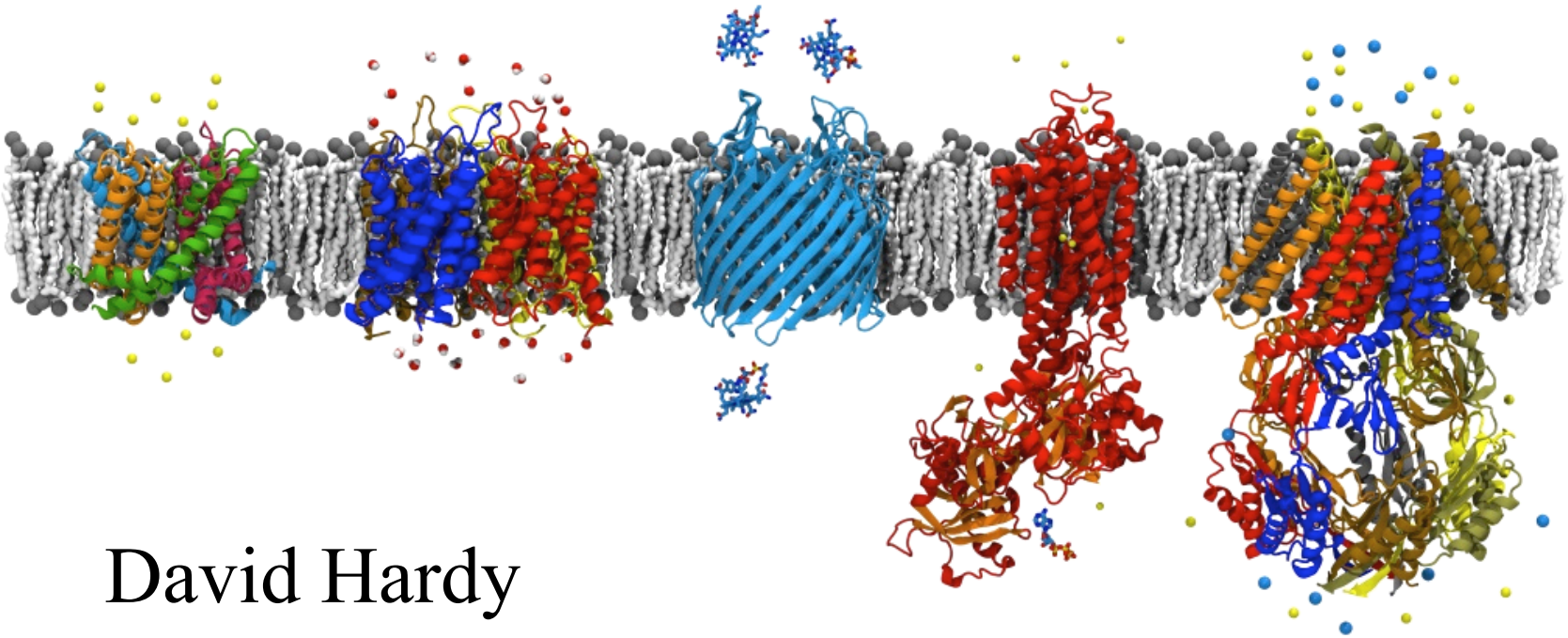
Demonstration: Using NAMD



David Hardy

http://www.ks.uiuc.edu/Research/~dhardy/

NAIS: State-of-the-Art Algorithms for Molecular Dynamics

Obtaining NAMD

- Download pre-built binary (recommended)
- Build it yourself (not too hard, but tedious)
  - Dependencies
    - Charm++ (http://charm.cs.uiuc.edu/software/)
    - FFTW (http://www.fftw.org/download.html)
    - TCL (http://www.tcl.tk/software/tcltk/)
    - Optionally: CUDA Toolkit
  - Recommended: Download source tar ball for that release, has Charm++ version in "charm" directory
  - Windows building requires "cygwin"

Building NAMD

- Recommended: For modern multi-core CPUs, use multicore builds of Charm++ and NAMD
- CAUTION: Charm++ and NAMD do not use identical names to describe the same architecture
  - `./build charm++ multicore-darwin-x86_64`
  - `./config MacOSX-x86_64-g++ \`
    `--with-tcl --tcl-prefix /opt/local \`
    `--with-fftw --fftw-prefix /opt/local \`
    `--with-cuda --cuda-prefix /usr/local/cuda`
- PROBLEM: NAMD CUDA supported on Linux only! It won't build on Windows or Mac.
  - Issue will be addressed after NAMD 2.9 is released

Running NAMD

- Recommended: Read NAMD tutorial (and VMD).
  - URL: http://www.ks.uiuc.edu/Training/Tutorials/
  - "Using VMD" and "NAMD Tutorial"
- Using NAMD CUDA:
  - Some capabilities are missing from CUDA build
    - E.g., alchemical free energy methods, coarse-grained models
  - Most of NAMD's capabilities work with CUDA build
  - Performance concern:
    - Calculating potential energy slows performance
    - Solution: Use "outputEnergies = 100" in configuration file

Executing NAMD

- Multicore workstation:
  - "namd2 +p4 mysim.conf"
- Multicore workstation with 2 GPUs:
  - "namd2 +p4 +devices 0,1 mysim.conf"
- Or use "charmrun namd2 …" (cluster build)
- Or use "mpiexec namd2 …" (MPI build)

NAMD Parallel Performance (1)

- Benchmark performance within 500 steps
  - Grep for "Benchmark time:" lines
  - Use "outputTiming = *nsteps*"
- Multicore builds might need a thread dedicated to communication (> 32 cores):
  - "namd2 +p47 +commthread"
- Maintain standard (UDP), TCP, ibverbs (InfiniBand) for Linux clusters
- Other high-speed network, use MPI
  - SMP builds don't scale as well due to communication thread bottleneck

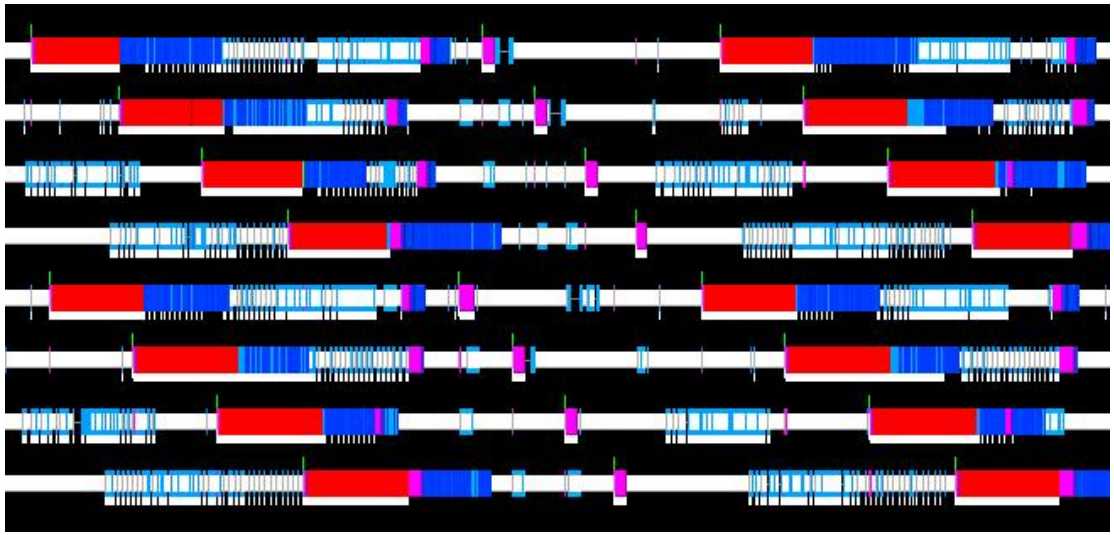
NAMD Parallel Performance (2)

- Short cycle lengths (< 10 steps) hurts scaling since atom migration sends so many messages
  - Keep cycle length at 20 steps
  - Pairlist distance will adjust automatically
  - One pairlist every 10 steps is good ratio
- Good performance when (#patches) ≥ (#PEs)
  - Otherwise increase (#patches) with "twoAwayX yes"
- Additional tuning advice on NAMD wiki:

  http://www.ks.uiuc.edu/Research/namd/wiki/?NamdPerformanceTuning

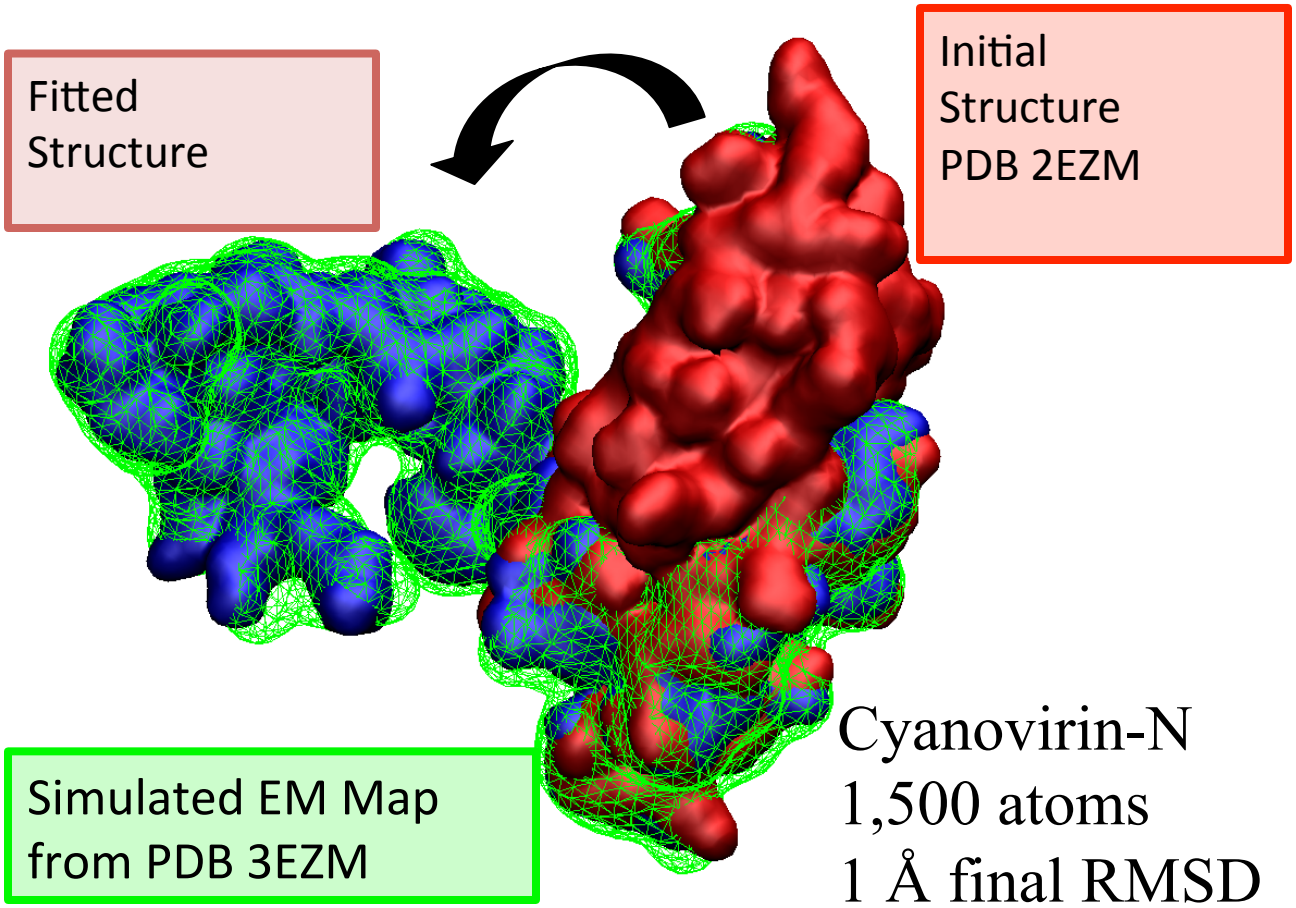
NAMD Profiling

- Use “Projections” Charm++ analysis tools
  - Charm++ runtime automatically records pertinent performance data
  - Rebuild NAMD, rerun benchmark, run projections on results
  - http://www.ks.uiuc.edu/Research/namd/wiki/?NamdPerformanceTuning



NAMD 2.9 Desktop MDFF

with GPU-Accelerated Implicit Solvent and CPU-Optimized Cryo-EM Forces



Fast: 2 ns/day

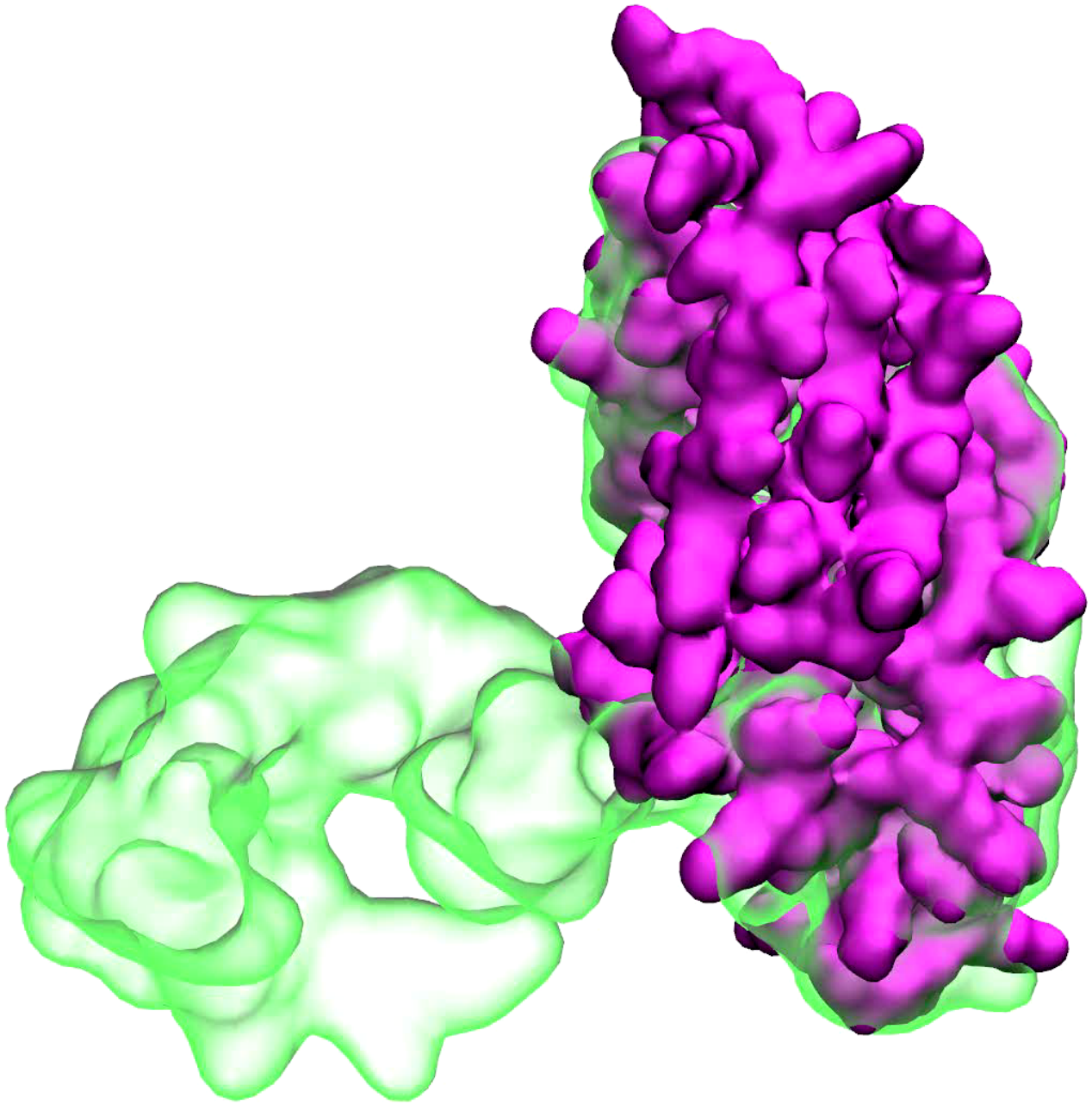
Explicit Solvent
8 cores (1X)

Faster: 12 ns/day

Implicit Solvent
8 cores (6X)

Fastest: 40 ns/day

Implicit Solvent
1 GPU (20X)



Small IMD Demo

- Since HectorGPU configuration does not allow streaming results to the laptop:
  - Demonstration with decalanine