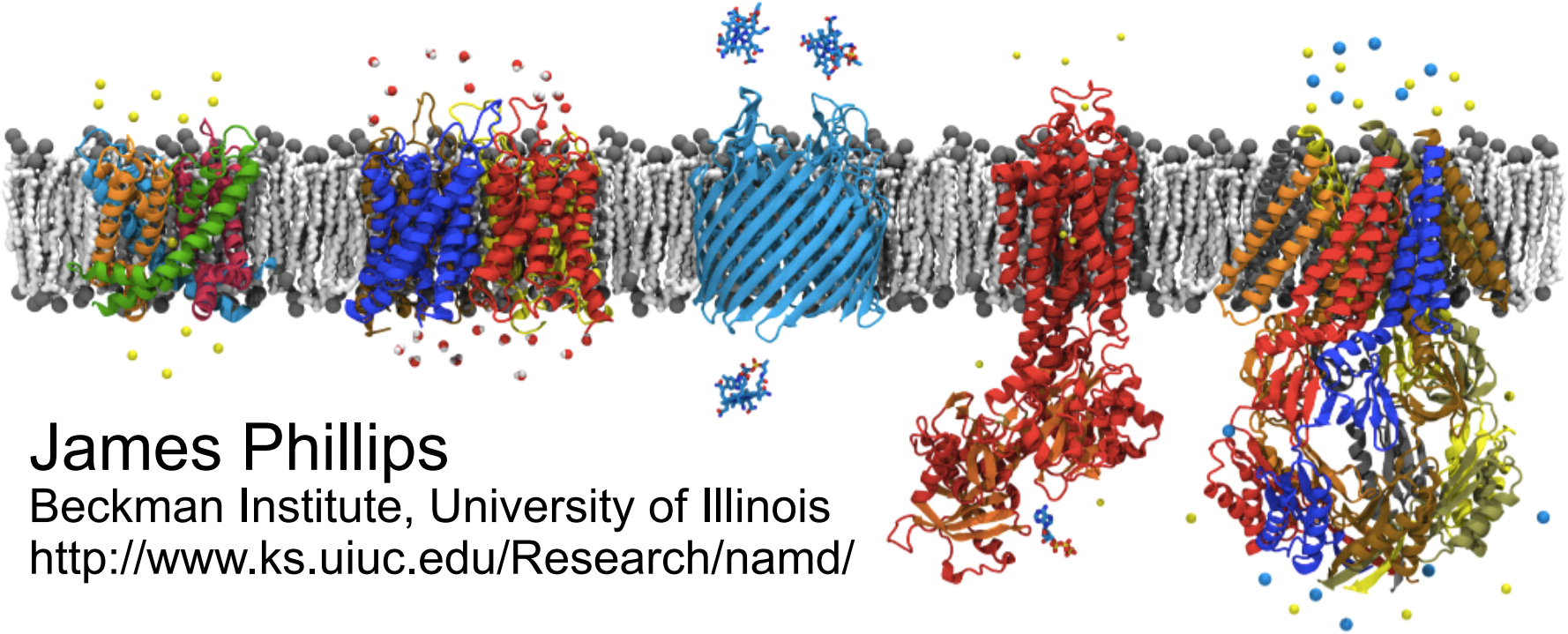
Petascale Charm++ in Practice: Lessons from Scaling NAMD



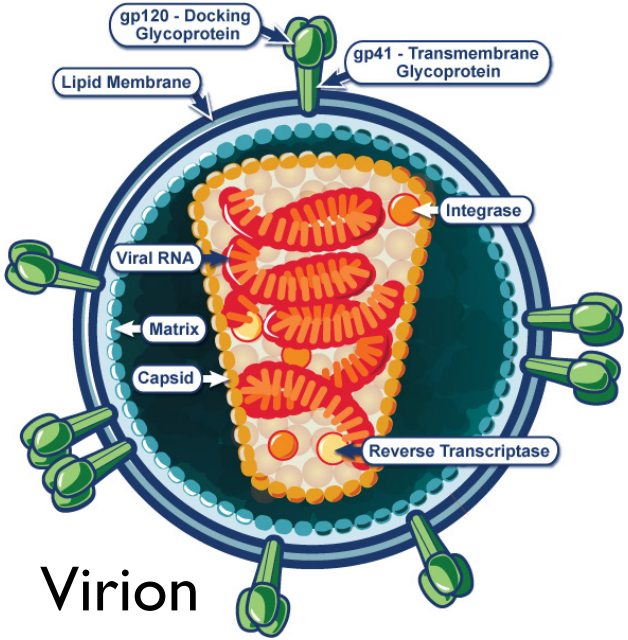
James Phillips

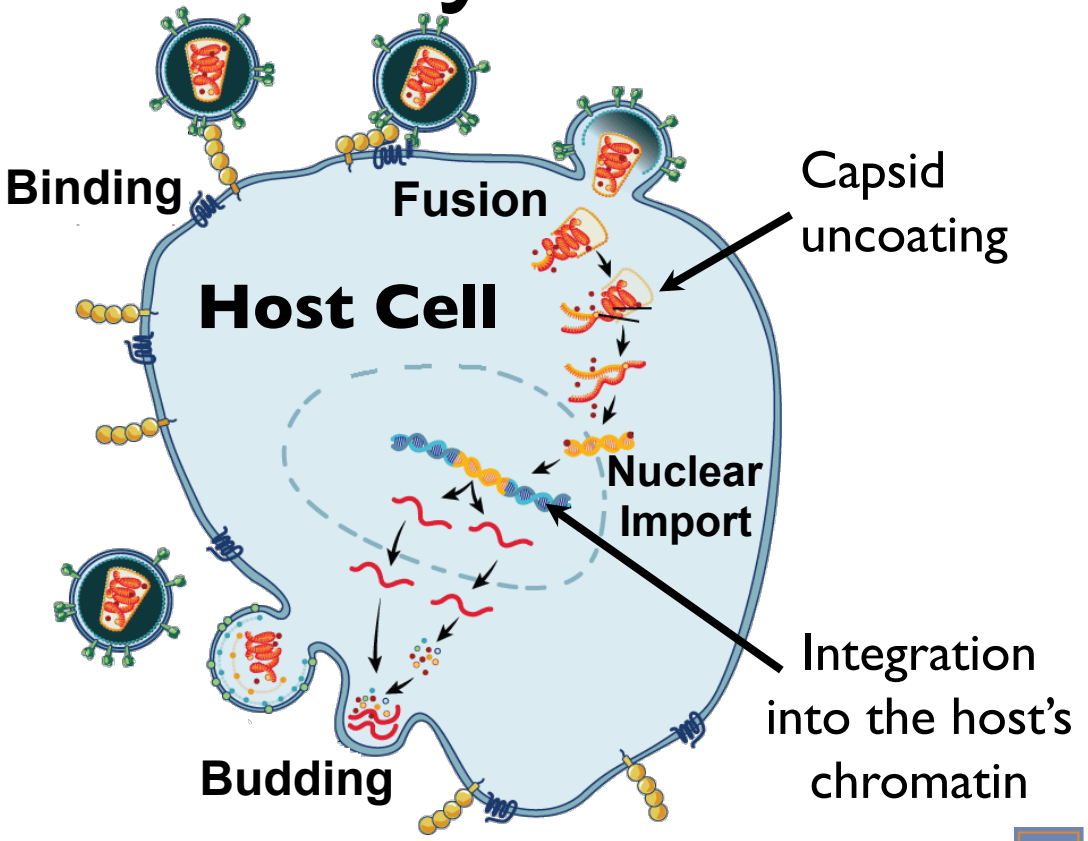
Beckman Institute, University of Illinois

http://www.ks.uiuc.edu/Research/namd/

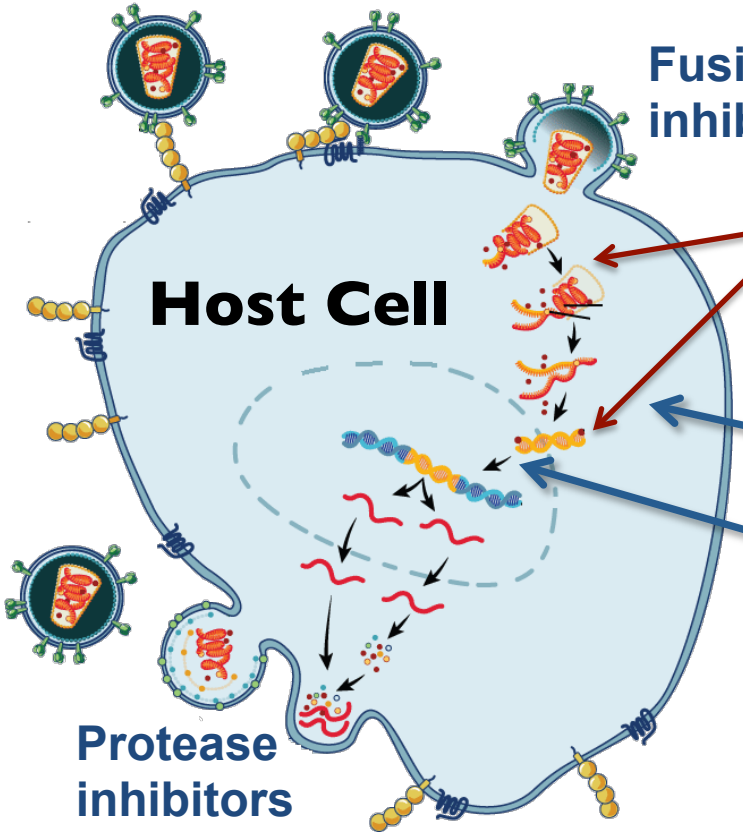
2013 *HPCwire* Editors' Choice Award for Best Use of HPC in Life Sciences

HIV Infective Cycle





HIV Treatment



Fusion/Entry inhibitors

Currently no drug targets *capsid uncoating* or *nuclear import* !

Reverse Transcription (RNA to DNA) inhibitors

Integrase inhibitors

Protease inhibitors

NIH Biomedical Technology Research Center for Macromolecular Modeling and Bioinformatics

Developers of the widely used computational biology software VMD and NAMD

250,000 registered VMD users
61,000 registered NAMD users

Renewed 2012-2017 with 10.0 score (NIH)

research projects include: virus capsids, ribosome, photosynthesis, protein folding, membrane reshaping, animal magnetoreception

600 publications (since 1972)
over 54,000 citations

Achievements Built on People

5 faculty members
8 developers
1 systems administrator
17 postdocs
46 graduate students
3 administrative staff

Tajkorshid, Luthey-Schulten, Stone, Schulten, Phillips, Kale, Mallon

NIH Center Facilities Enable Petascale Biology

External Resources,
90% of our
Computer Power

Over the past five years the Center has assembled all necessary hardware and infrastructure to prepare and analyze petascale molecular dynamics simulations, and ***makes these facilities available to visiting researchers***.

Simulation Output

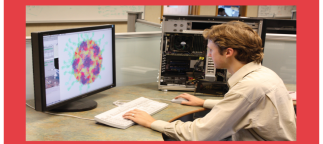
10 Gigabit Network

Petascale Gateway Facility

Storage

Compute

Visualization



High-End Workstations
Accessible to Visitors

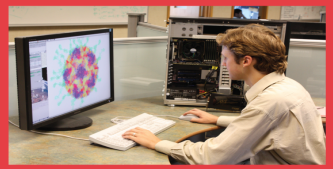
Virtual Facilities Enable Petascale Anywhere

High-end visualization and analysis workstations currently available only in person at the Beckman Institute must be ***virtualized and embedded at supercomputer centers***.

Storage

Compute

Visualization

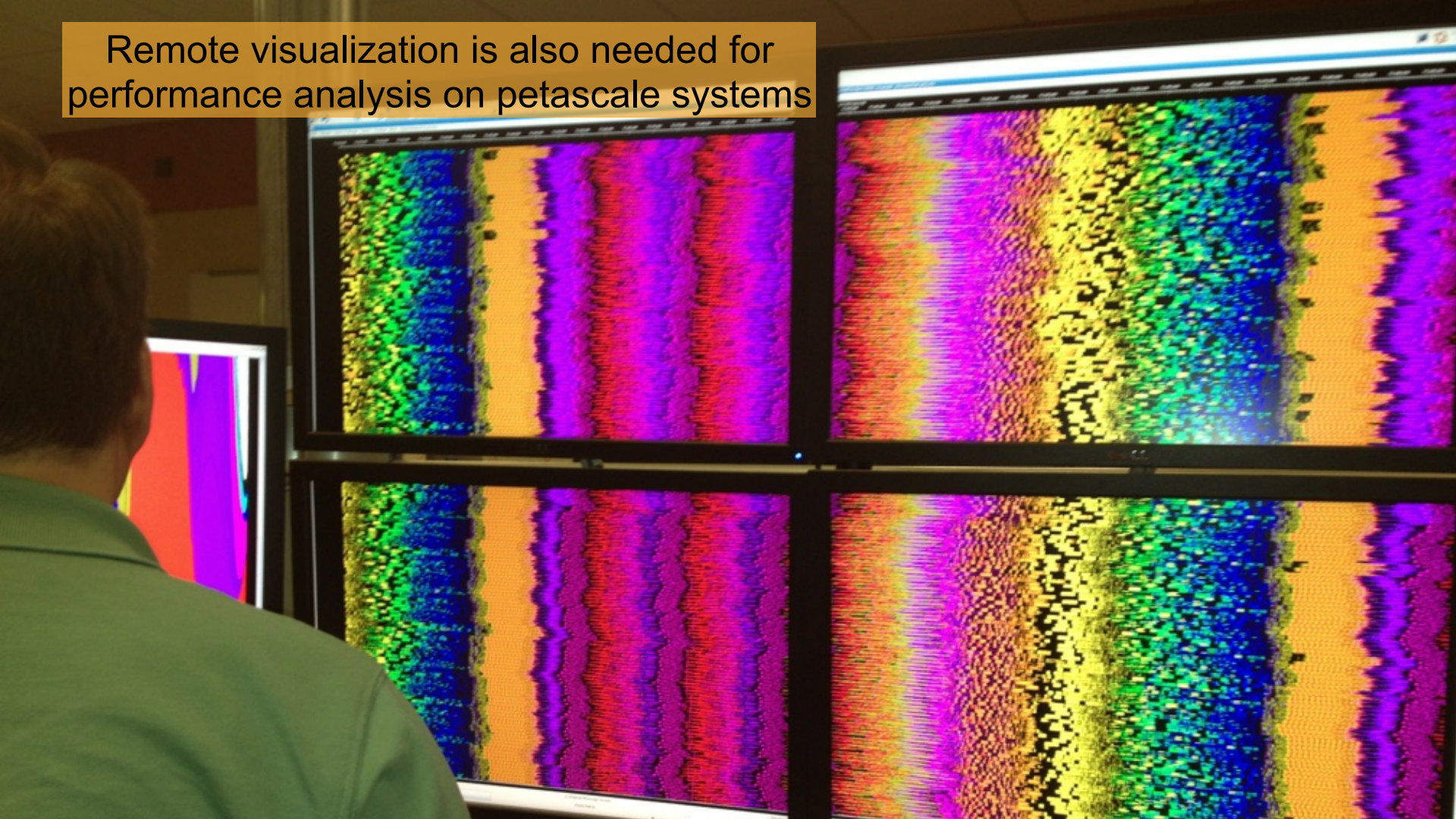


Compressed Video

1 Gigabit Network

Remote Visualization Now

- TACC Stampede supports this today
  - Includes nodes with 1TB memory
  - Not virtualized, allocate full dedicated node
  - New Maverick cluster being added
- Blue Waters – no visualization resource
- Titan – new Rhea “viz” cluster drops GPUs


Remote visualization is also needed for
performance analysis on petascale systems

NAMD Serves NIH Users and Goals

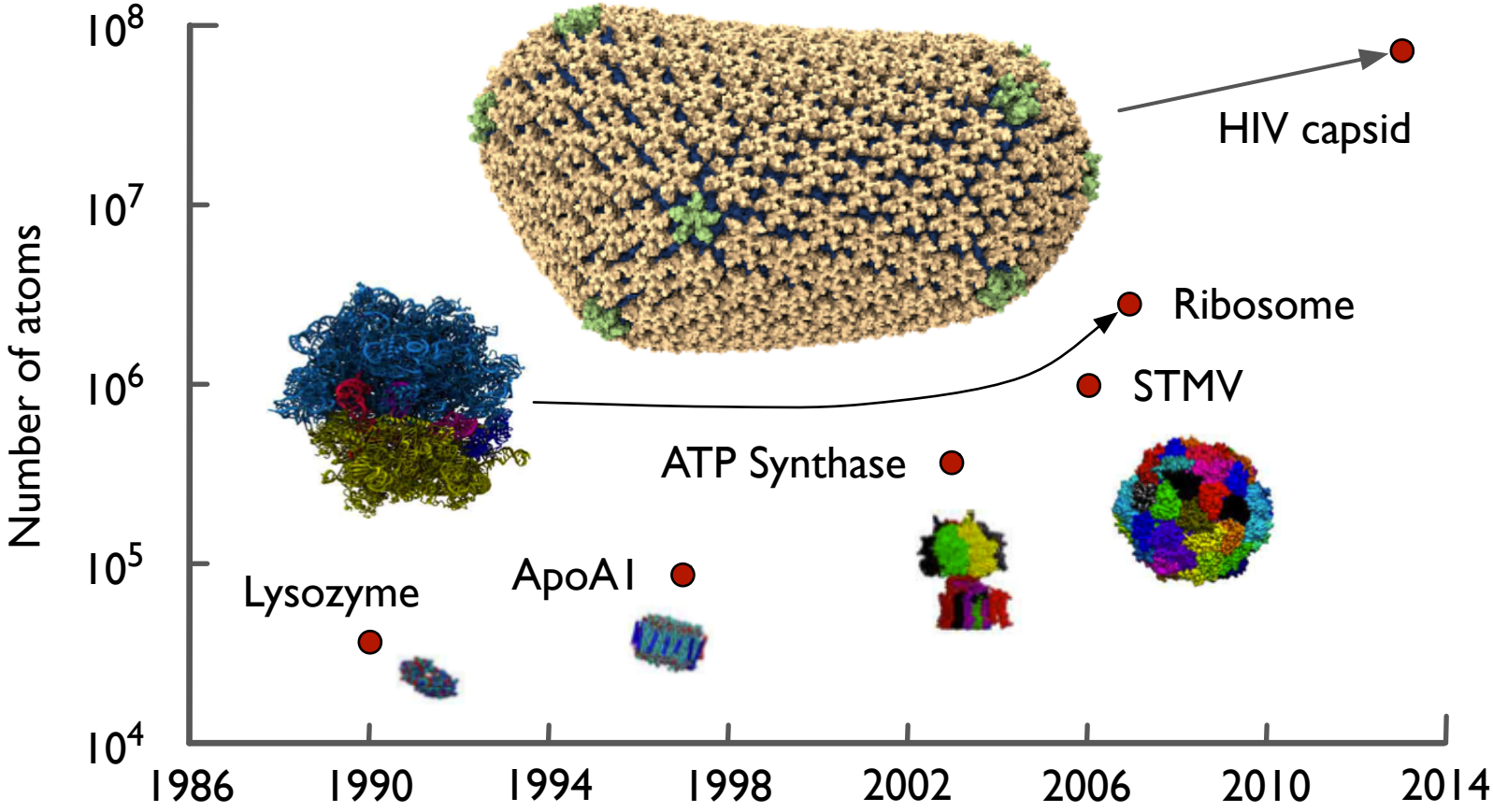
Practical Supercomputing for Biomedical Research

- 60,000 users can't all be computer experts.
  - 18% are NIH-funded; many in other countries.
  - 17,000 have downloaded more than one version.
  - 4000 citations of NAMD reference papers.
- One program available on all platforms.
  - Desktops and laptops – setup and testing
  - Linux clusters – affordable local workhorses
  - Supercomputers – free allocations on XSEDE
  - Blue Waters – sustained petaflop/s performance
  - GPUs - next-generation supercomputing
- User knowledge is preserved across platforms.
  - No change in input or output files.
  - Run any simulation on **any number of cores.**
- Available free of charge to all.

Hands-On Workshops

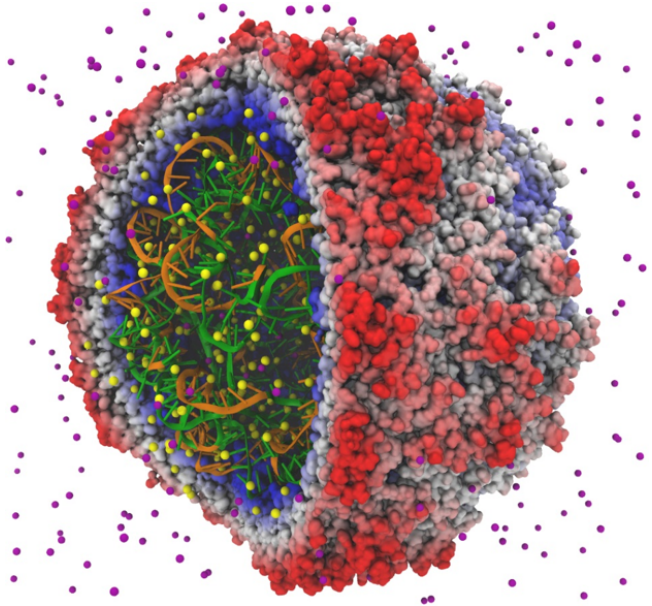
Oak Ridge TITAN

Structural data drives simulations



First Simulation of a Virus Capsid (2006)

Satellite Tobacco Mosaic Virus (STMV)



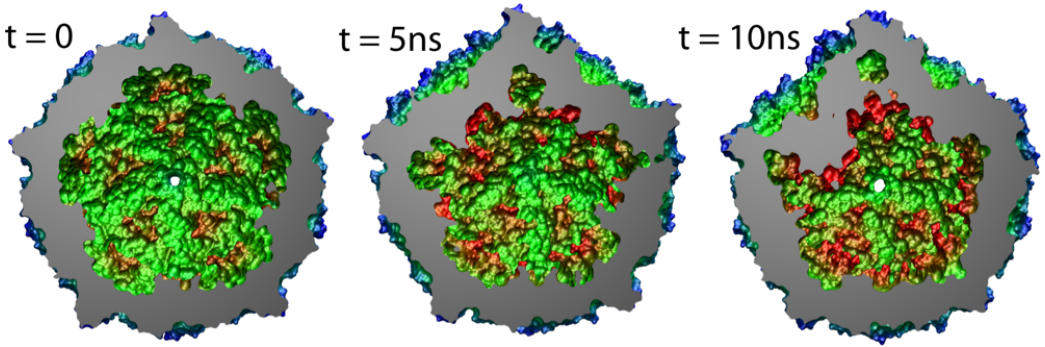
1 million atoms

huge system for 2006

First MD simulation of a complete virus capsid

STMV smallest available capsid structure

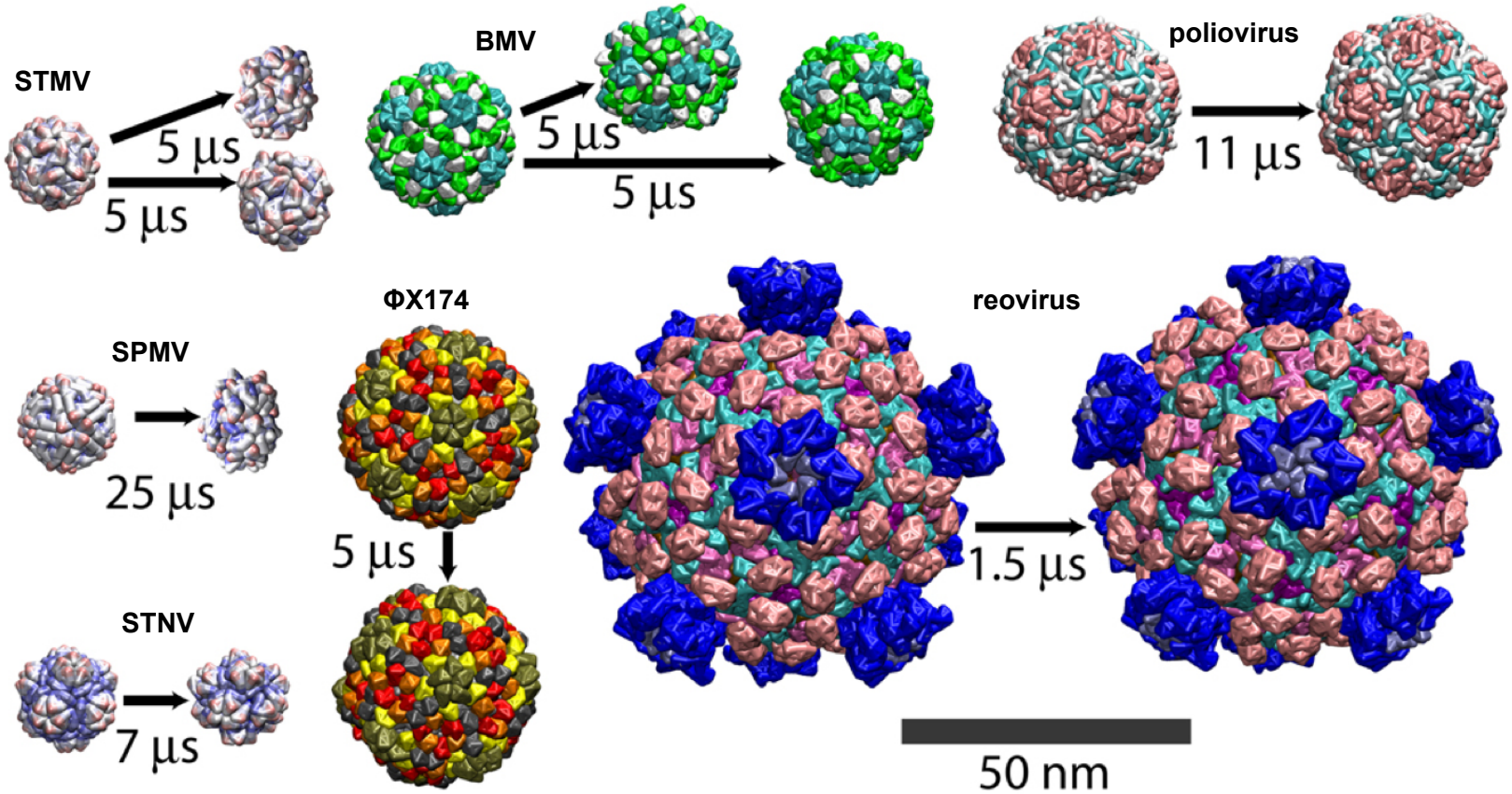
MD showed that STMV capsid collapses without its RNA core



Freddolino et al., ***Structure***, **14**:437 (2006)

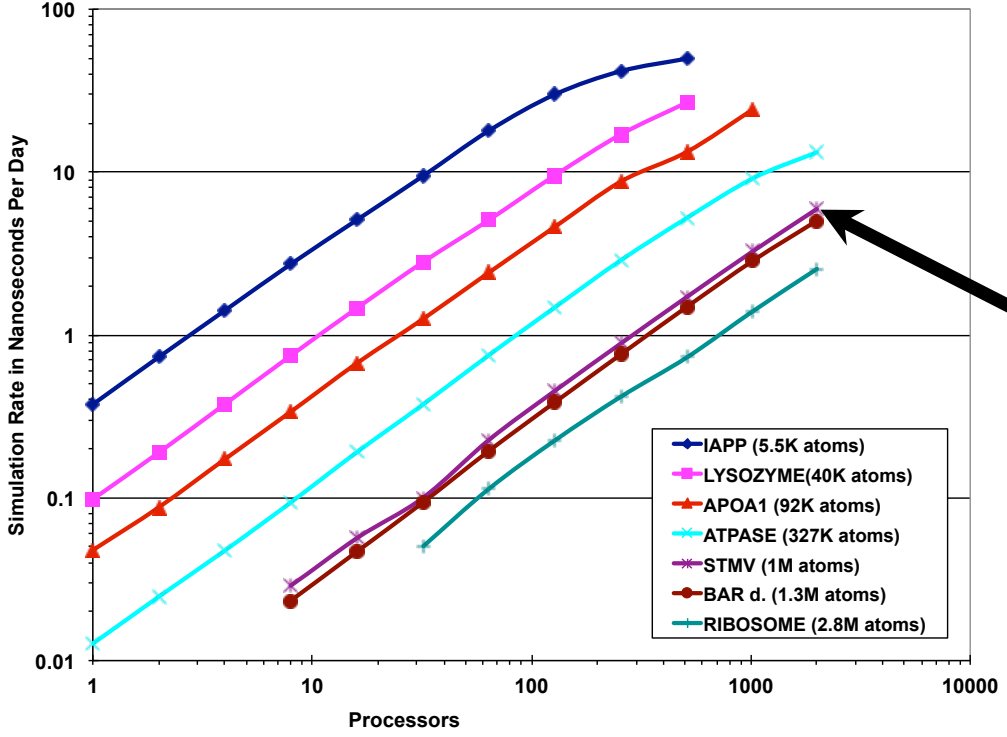
Coarse-Grained Simulation of Viral Capsids (2006)

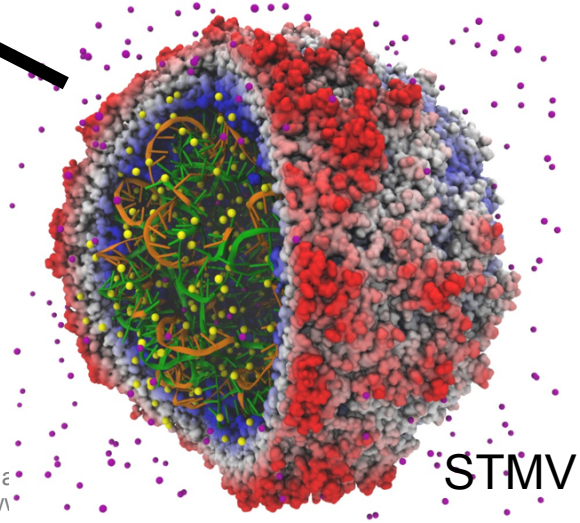
Coarse-graining permits long simulations, but has many limitations...



Arkhipov, Freddolino, and Schulten, ***Structure***, **14**:1767 (2006)

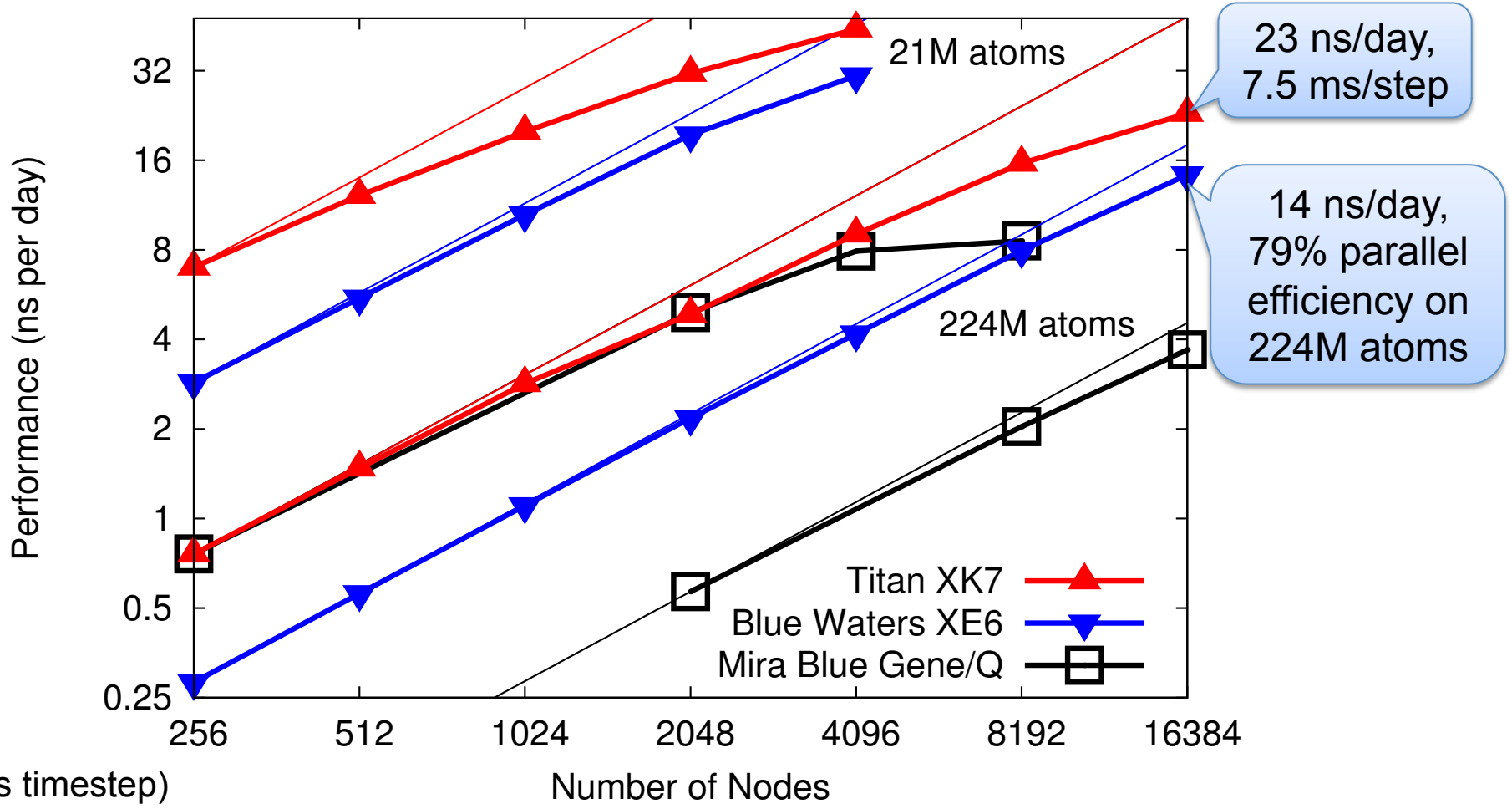
2007 Performance on PSC Cray XT3





STMV

NAMD on Petascale Platforms



(2fs timestep)

NAMD Benefits from Charm++ Collaboration

- Illinois Parallel Programming Lab
  - Prof. Laxmikant Kale
  - charm.cs.illinois.edu

- Long standing collaboration
  - Since start of Center in 1992
  - Gordon Bell award at SC2002
  - Joint Fernbach award at SC12

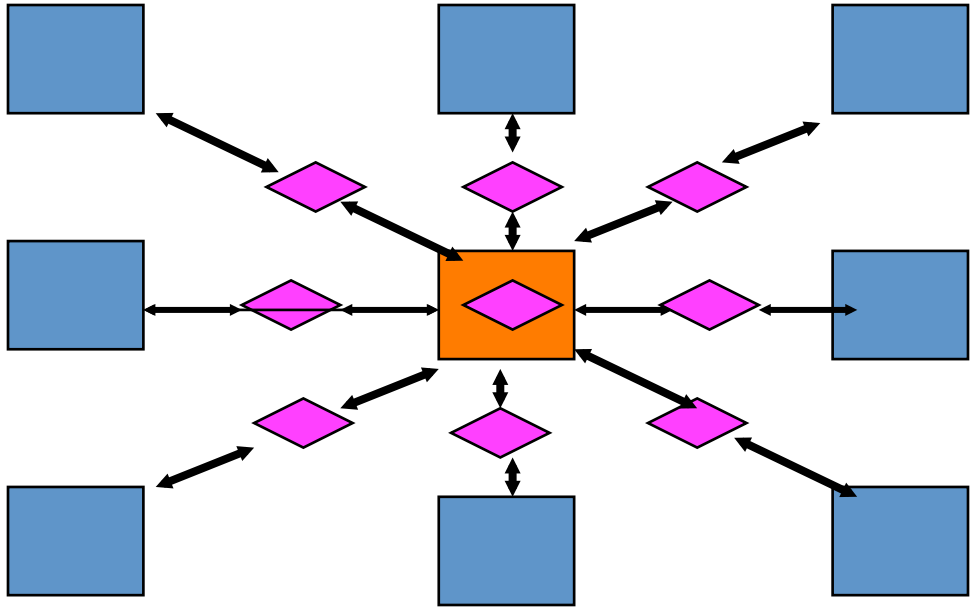
- Synergistic research
  - NAMD requirements drive and validate CS work
  - Charm++ software provides unique capabilities
  - Enhances NAMD performance in many ways

Charm++ Used by NAMD

- Parallel C++ with *data driven* objects.
- Asynchronous method invocation.
- Prioritized scheduling of messages/execution.
- Measurement-based load balancing.
- Portable messaging layer.

NAMD Hybrid Decomposition

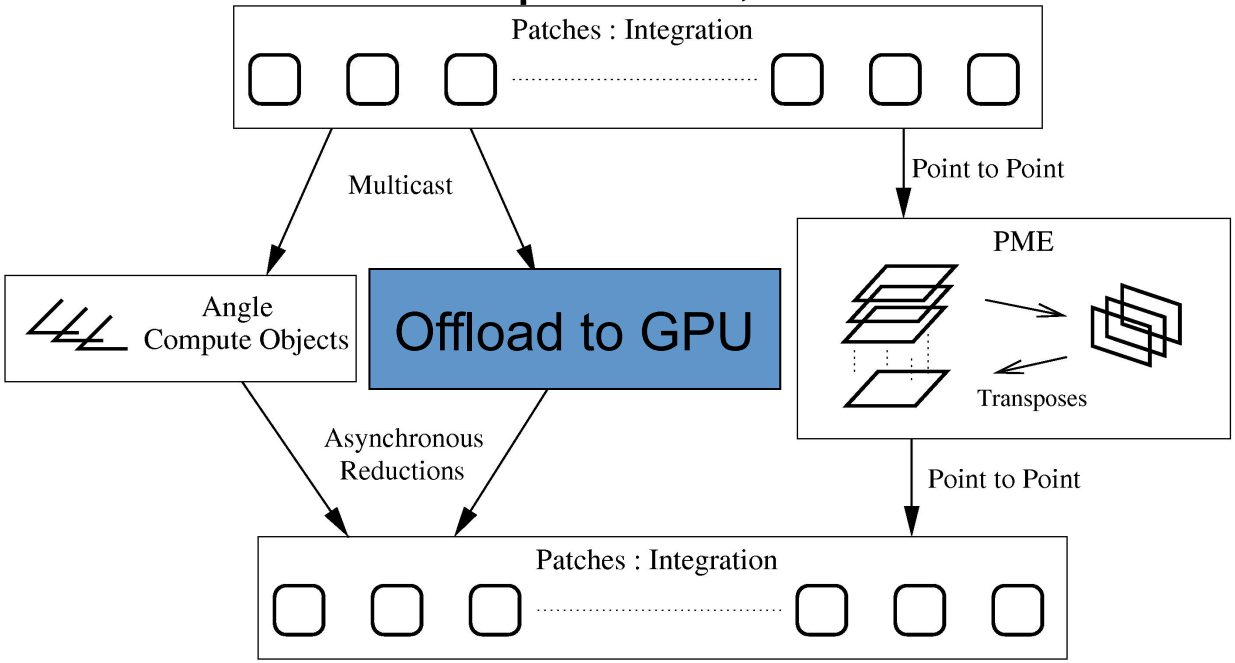
Kale *et al., J. Comp. Phys.* 151:283-312, 1999.



- Spatially decompose data and communication.
- Separate but related work decomposition.
- "Compute objects" facilitate iterative, measurement-based load balancing system.

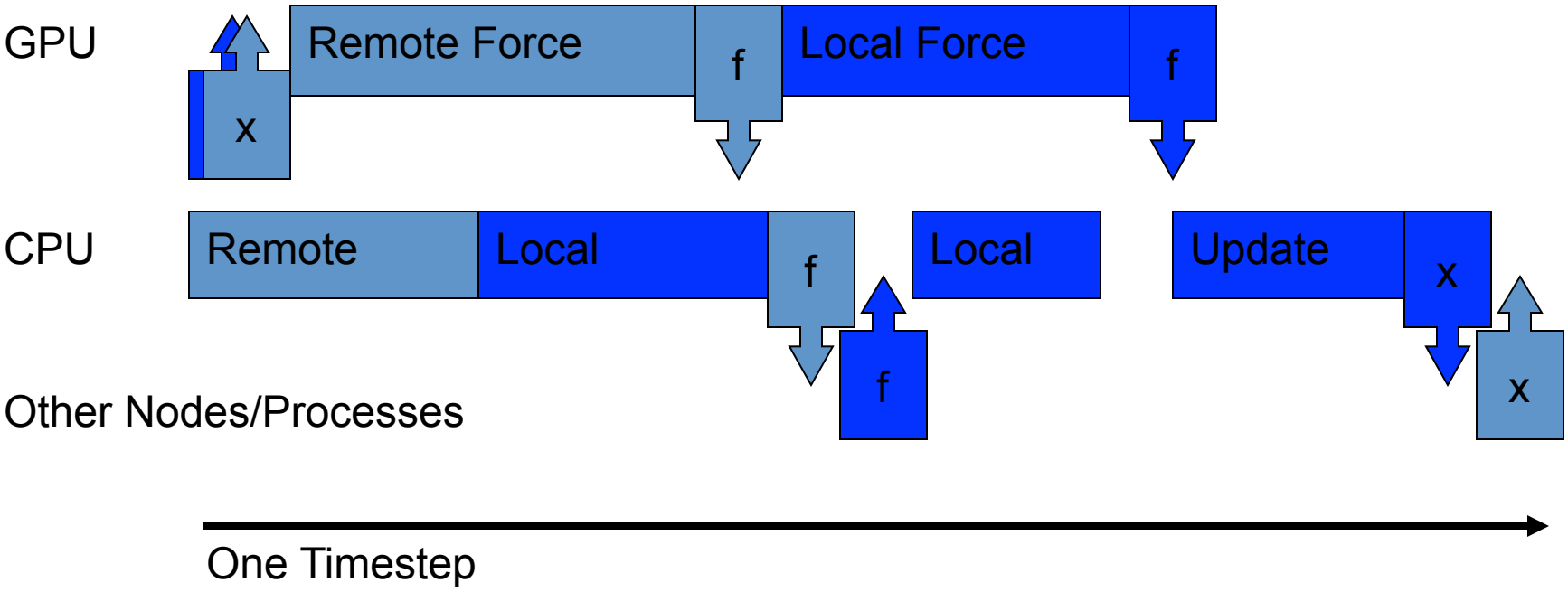
NAMD Overlapping Execution

Phillips *et al., SC2002*.



Objects are assigned to processors and queued as data arrives.

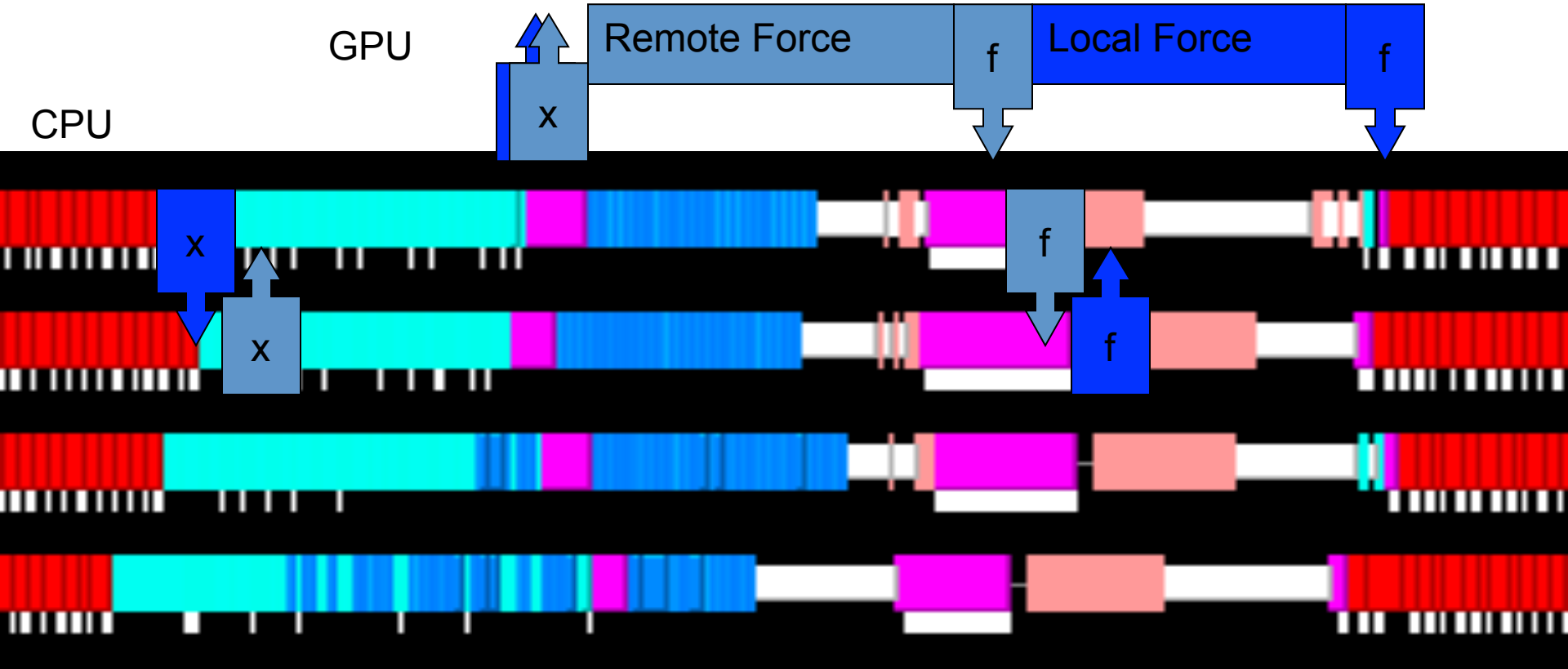
Overlapping GPU and CPU with Communication



Phillips *et al.*, SC2008

Actual Timelines from NAMD

Generated using Charm++ tool “Projections” http://charm.cs.uiuc.edu/



Blue Waters Posed Many Challenges

- Scale NAMD to 100M atoms
  - Read new .js file format
  - Distribute or compress static molecular structure data
  - Parallel atomic data input
  - Use shared memory in a node
  - Parallel load balancing
  - Parallel, asynchronous trajectory and restart file output
  - 2D decomposition of 3D FFT
  - Limit steering force messages
  - Fix minimizer stability issues
- Also build benchmarks…

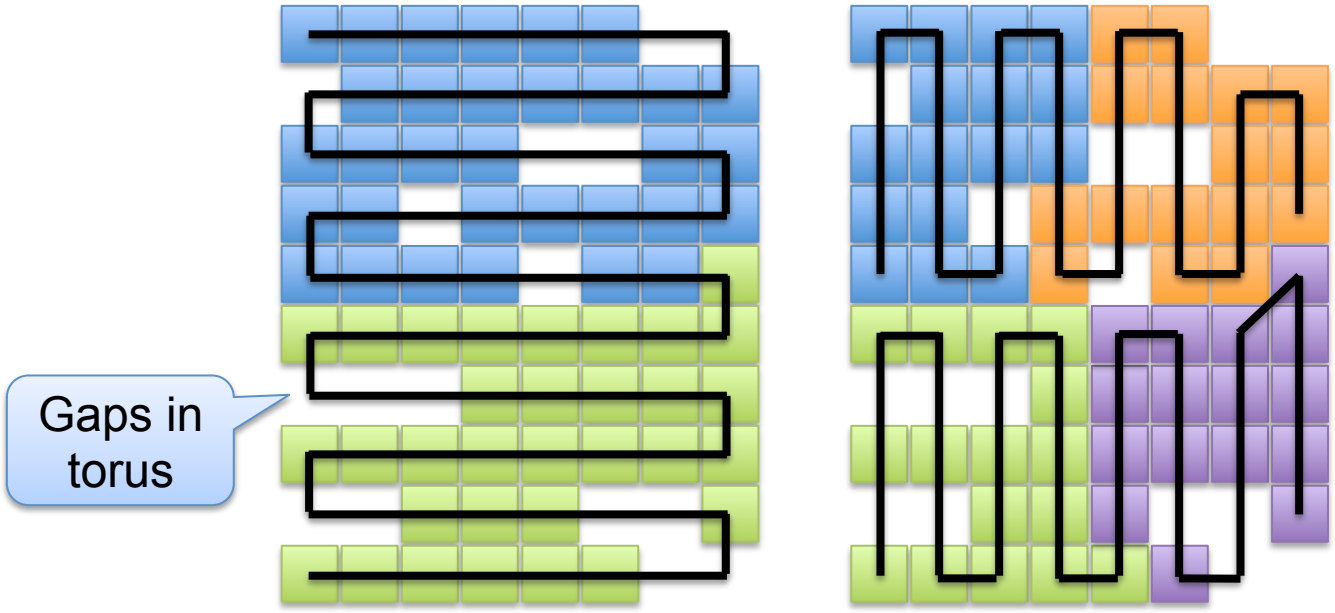
- Scale NAMD to 300K cores
  - Charm++ shared memory tuning
  - IBM Power7 network layer
  - IBM BlueGene/Q network layer
  - Cray Gemini network layer
  - Cray torus topology information
  - Charm++ replica layers
  - Optimize for physical nodes
  - Adapt trees to avoid throttling
  - Optimize for torus topology
  - Optimize for parallel filesystem
- Also optimize for GPUs…

Torus Topology Adaptation

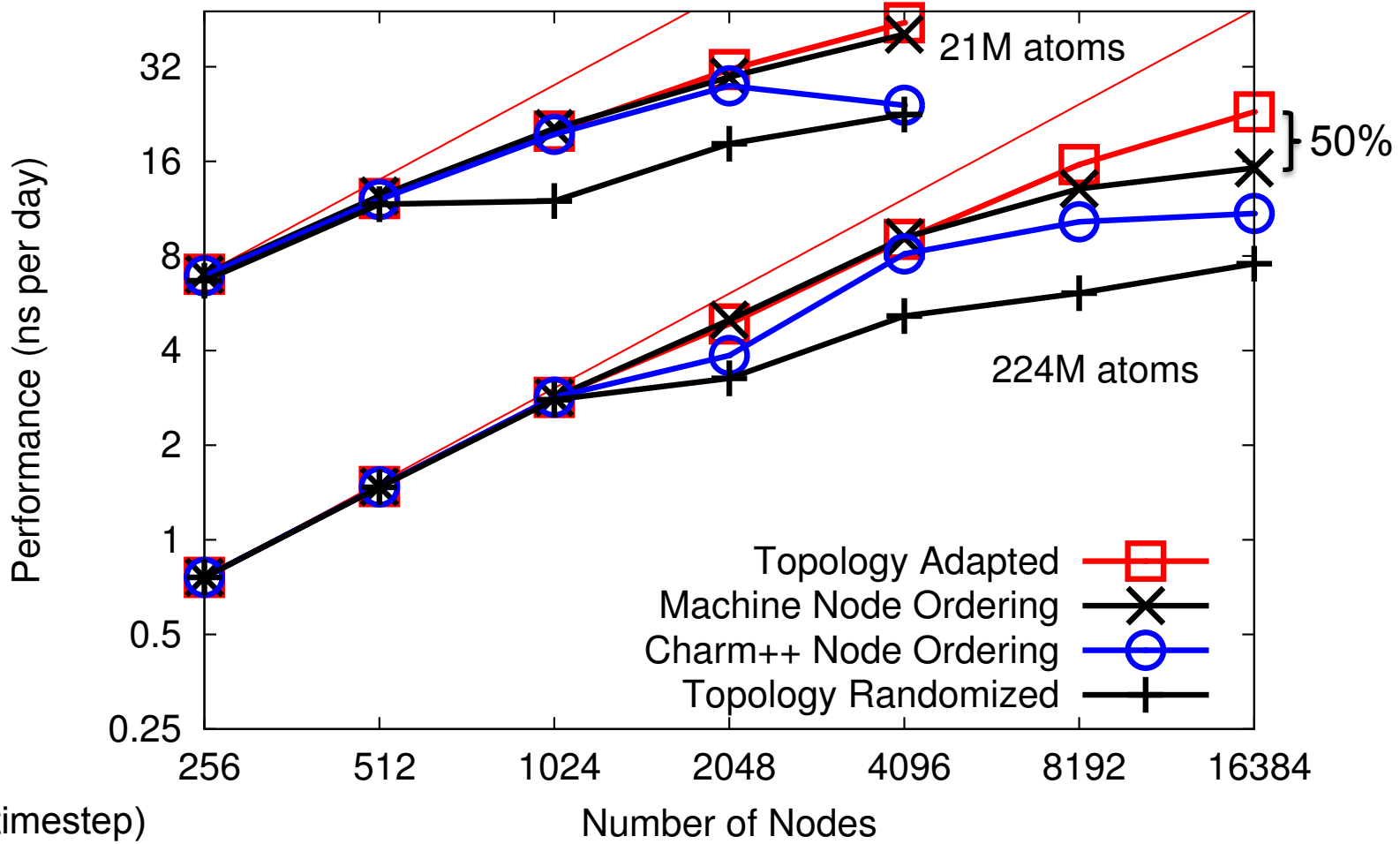
- Re-wrap network torus to minimal grid
- Simultaneous recursive bisection to align patch grid and network torus
- Simultaneous recursive bisection to align PME Z-pencil and host PE average patch coordinates

- Recursive bisection also used for partitioning

Recursive Bisection Example



NAMD Topology Mapping on Titan Cray XK7

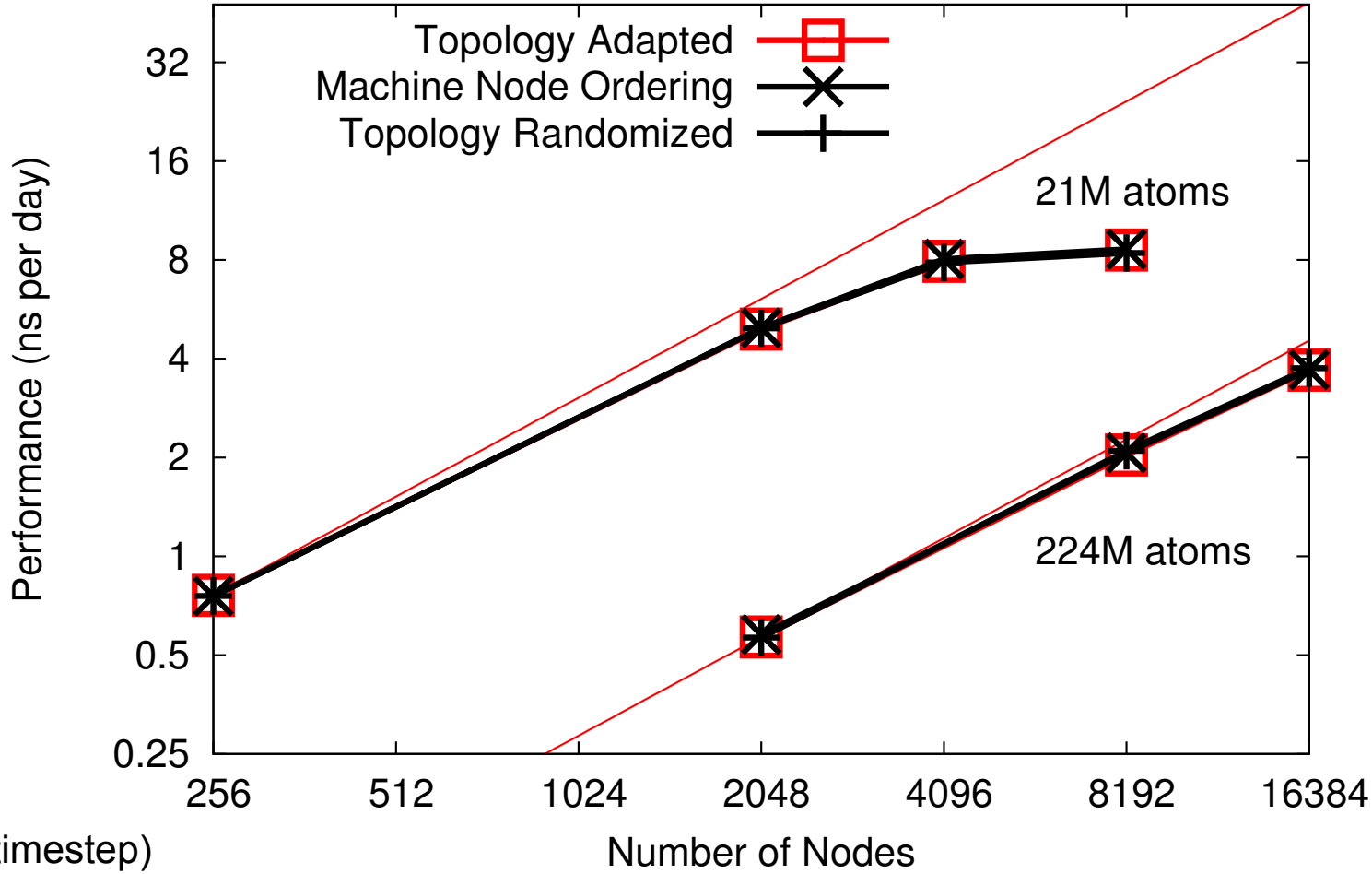


(2fs timestep)

Benchmarking Caution

- Cray XE/XK performance varies due to:
  - Compactness of nodes assigned to job
  - Other jobs running on machine (cross-traffic)
  - I/O activity (more Blue Waters than Titan)
- To test performance impact of changes, run old and new back-to-back in *same job*.

NAMD Topology Mapping on Mira Blue Gene/Q



(2fs timestep)

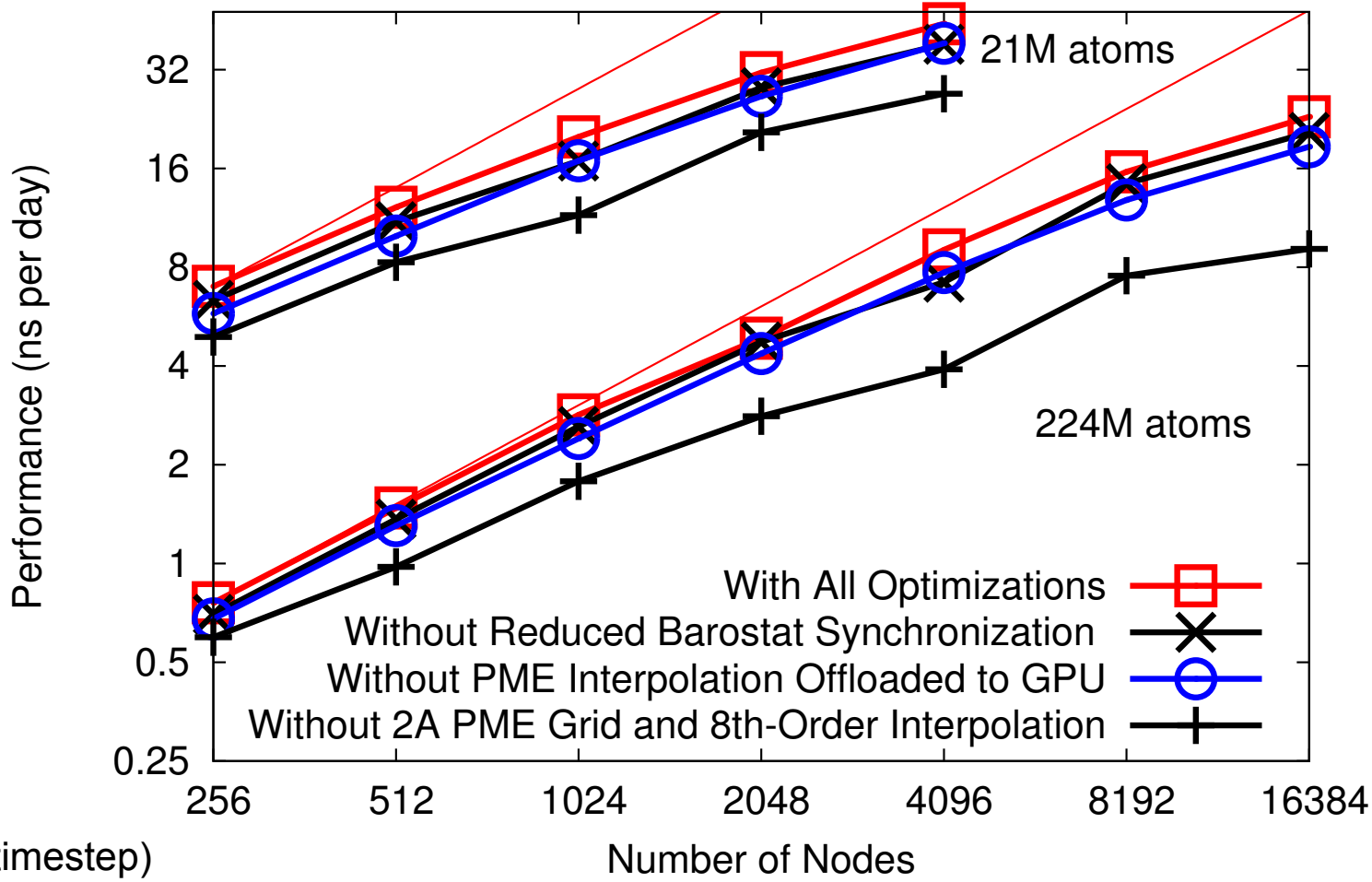
NAMD PME CUDA Kernel

- Bottleneck for 100M atoms is PME FFT communication
  - Switch from 4th-order to 8th-order interpolation on coarser grid
- Doing 8th-order PME on GPU improves critical path
- CPU may be bottleneck for 8th-order PME
  - Especially as GPU non-bonded gets faster…
- Simplest design that might possibly work:
  - One stream per host PE (preserve control flow)
  - One atom per warp with warp-synchronous programming
  - Atomics to accumulate charge grid in global memory
    - One per thread so accesses coalesce
    - Also build “used” flags arrays for x-y pencils and z plane

PME Kernel Aggregation

- Initial version slower than PME on CPU
- First, one launch per PE, not per patch
- Second, one charge array per node
  - First version to beat PME on CPU
  - Node-level coordination a challenge in Charm++
  - Reduces number of messages sent per node!
    - Need to backport to PME on CPU version
    - May help CPU-only version, but not as much

Other NAMD Optimizations on Titan Cray XK7



(2fs timestep)

Lessons Learned

- Optimizing communication is job one:
  - **Eliminate PE *n* to PEs 2*n*+1, 2*n*+2 trees**
  - Avoid messages between physical nodes
  - Within physical node not as bad
  - Within node (process) is virtually free
  - Trading 8th order PME computation for reduced communication a big win!

Lessons Learned

- Incremental development is possible
  - Go from working code to working code
  - Fix memory issues as they arise
  - Use profiling to diagnose performance issues
    - But solutions may require deeper insight
  - Major rewrites require major effort
    - Make sure you understand the issues first!

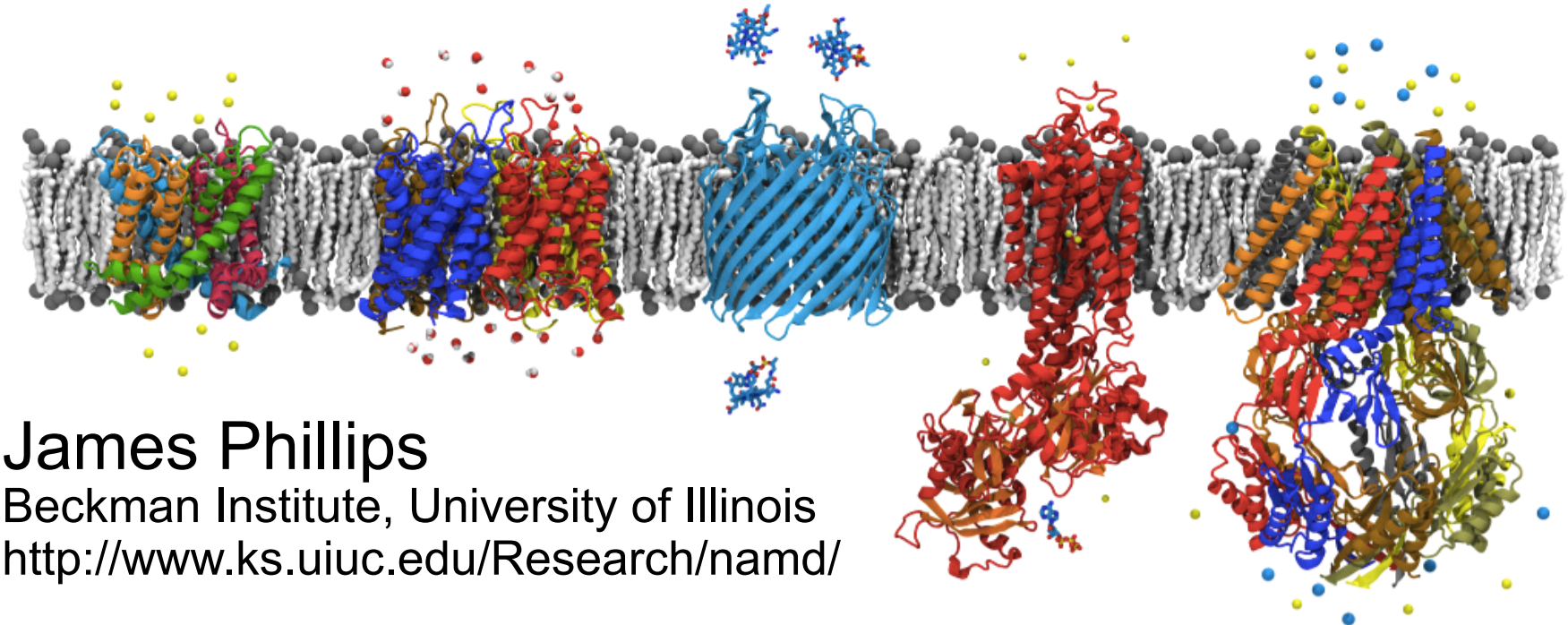
Lessons for Charm++

- Need to shift programming orientation from PE (thread) to node (process):
  - Shared data and parallel I/O
  - Offload processors (GPU/MIC)
  - Dynamic load balancing within node
  - Array elements execute on any PE in node
    - I.e., *nodearray* analogous to *nodegroup*

Lessons for Charm++

- Need better support for execution fences
  - Don’t start X on PE/node until all Y starts
  - Avoids delay of highest-priority work
  - Recurring idiom:
    - Hold computes until all patches/proxies ready
    - Hold GPU nonbonded until GPU PME
    - Used for performance, not correctness
  - Enables MPI-style sequential programming

Thanks to: NIH, NSF, DOE, NCSA,
NVIDIA (**Sarah Tariq**, Patric Zhao, Sky Wu, Justin Luitjens, Nikolai Sakharnykh),
Cray (Sarah Anderson, Ryan Olson), NCSA (Robert Brunner),
PPL (Eric Bohm, Yanhua Sun, Gengbin Zheng, Nikhil Jain)
and 19 years of NAMD and Charm++ developers and users.



James Phillips
Beckman Institute, University of Illinois
http://www.ks.uiuc.edu/Research/namd/