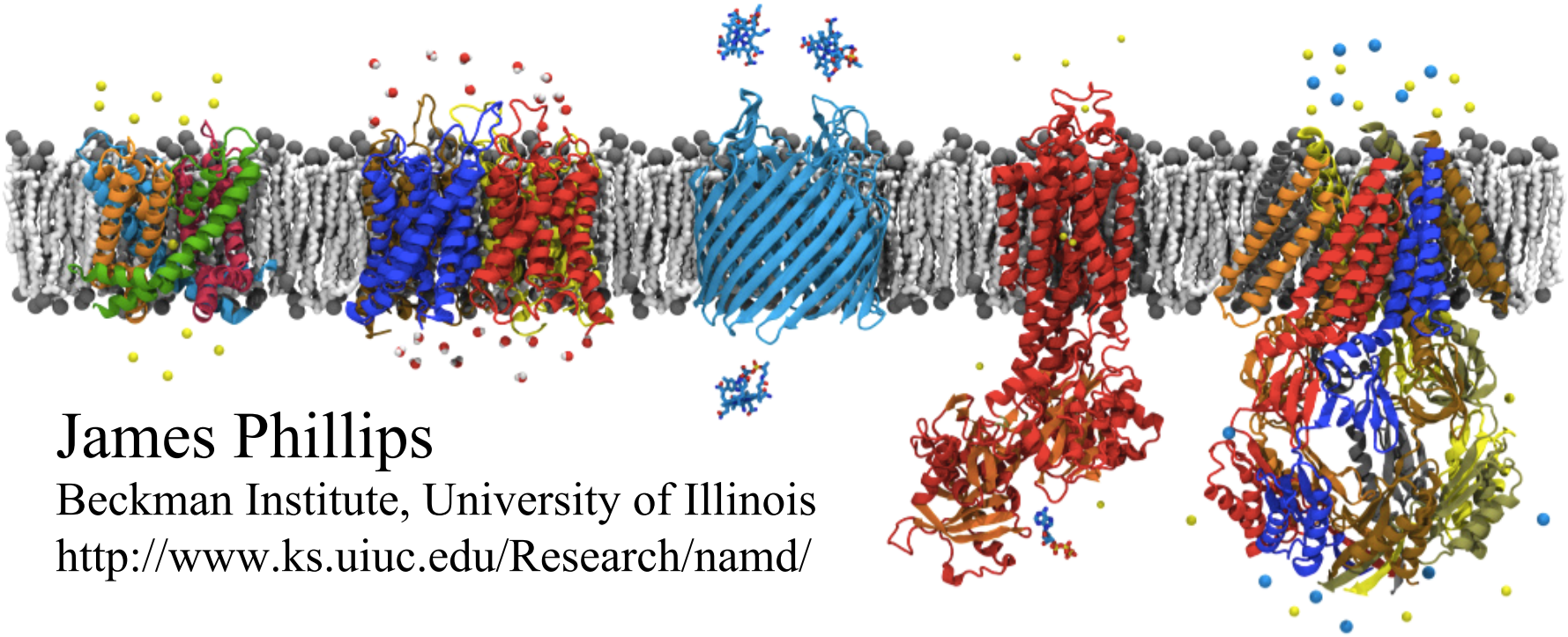
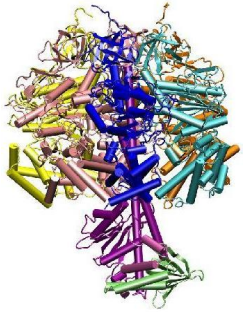
Petascale Molecular Dynamics Simulations on Titan and Blue Waters



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

NAMD: Scalable Molecular Dynamics

2002 Gordon Bell Award



ATP synthase

PSC Lemieux

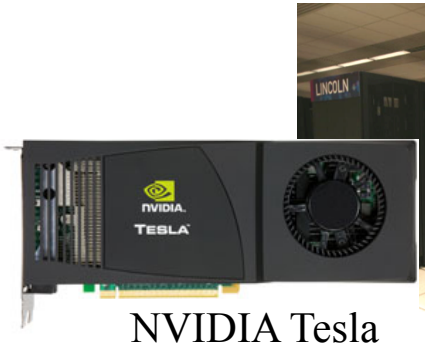
57,000 Users, 2900 Citations

Computational Biophysics Summer School

Blue Waters Target Application

Illinois Petascale Computing Facility

GPU Acceleration



NVIDIA Tesla

NCSA Lincoln

Biomedical Technology Research Center for Macromolecular Modeling and Bioinformatics
http://www.ks.uiuc.edu/

Beckman Institute, UIUC

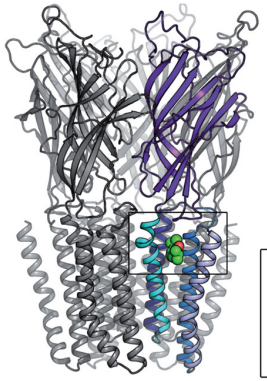
NAMD impact is broad and deep

- Comprehensive, industrial-quality software
  - Integrated with VMD for simulation setup and analysis
  - Portable extensibility through Tcl scripts (also used in VMD)
  - Consistent user experience from laptop to supercomputer
- Large user base – 57,000 users
  - 10,300 (18%) are NIH-funded; many in other countries
  - 16,600 have downloaded more than one version
- Leading-edge simulations
  - "most-used software" on NICS Cray XT5 (largest NSF machine)
  - "by far the most used MD package" at TACC (2nd and 3rd largest)
  - NCSA Blue Waters early science projects and acceptance test
  - Argonne Blue Gene/Q early science project

Outside researchers choose NAMD and succeed

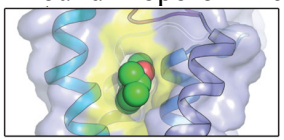
2900 external citations since 2007

Corringer, et al., *Nature*, 2011

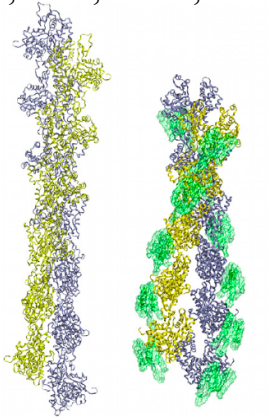


180K-atom 30 ns study of anesthetic binding to bacterial ligand-gated ion channel provided "complementary interpretations…that could not have been deduced from the static structure alone."

Bound Propofol Anesthetic



Voth, et al., *PNAS*, 2010



500K-atom 500 ns investigation of effect of actin depolymerization factor/cofilin on mechanical properties and conformational dynamics of actin filament.

Bare actin Cofilactin

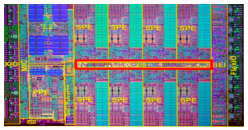
Recent NAMD Simulations in *Nature*

- **M. Koeksal, et al.,** *Taxadiene synthase structure and evolution of modular architecture in terpene biosynthesis*. (2011)
- **C.-C. Su, et al.**, *Crystal structure of the CusBA heavy-metal efflux complex of Escherichia coli.* (2011)
- **D. Slade, et al.**, *The structure and catalytic mechanism of a poly(ADP-ribose) glycohydrolase*. (2011)
- **F. Rose, et al.,** *Mechanism of copper(II)-induced misfolding of Parkinson's disease protein*. (2011)
- **L. G. Cuello, et al.**, *Structural basis for the coupling between activation and inactivation gates in K(+) channels*. (2010)
- **S. Dang, et al.,**, *Structure of a fucose transporter in an outward-open conformation.* (2010)
- **F. Long, et al.**, *Crystal structures of the CusA efflux pump suggest methionine-mediated metal transport*. (2010)
- **R. H. P. Law, et al.**, *The structural basis for membrane binding and pore formation by lymphocyte perforin.* (2010)
- **P. Dalhaimer and T. D. Pollard**, *Molecular Dynamics Simulations of Arp2/3 Complex Activation.* (2010)
- **J. A. Tainer, et al.**, *Recognition of the Ring-Opened State of Proliferating Cell Nuclear Antigen by Replication Factor C Promotes Eukaryotic Clamp-Loading*. (2010)

Early Acceleration Options

- Outlook in 2005-2006:
  - FPGA reconfigurable computing (with NCSA)
    - Difficult to program, slow floating point, expensive

  - Cell processor (NCSA hardware)
    - Relatively easy to program, expensive



  - ClearSpeed (direct contact with company)
    - Limited memory and memory bandwidth, expensive

  - MDGRAPE
    - Inflexible and expensive

  - Graphics processor (GPU)
    - Program must be expressed as graphics operations

CUDA: Practical Performance

November 2006: NVIDIA announces CUDA for G80 GPU.

- CUDA makes GPU acceleration usable:
  - Developed and supported by NVIDIA.
  - No masquerading as graphics rendering.
  - New shared memory and synchronization.
  - No OpenGL or display device hassles.
  - Multiple processes per card (or vice versa).
- BTRC and collaborators make it useful:
  - Experience from VMD development
  - David Kirk (Chief Scientist, NVIDIA)
  - Wen-mei Hwu (ECE Professor, UIUC)

Stone *et al.*, *J. Comp. Chem.* **28**:2618-2640, 2007.

Know Your Supercomputers

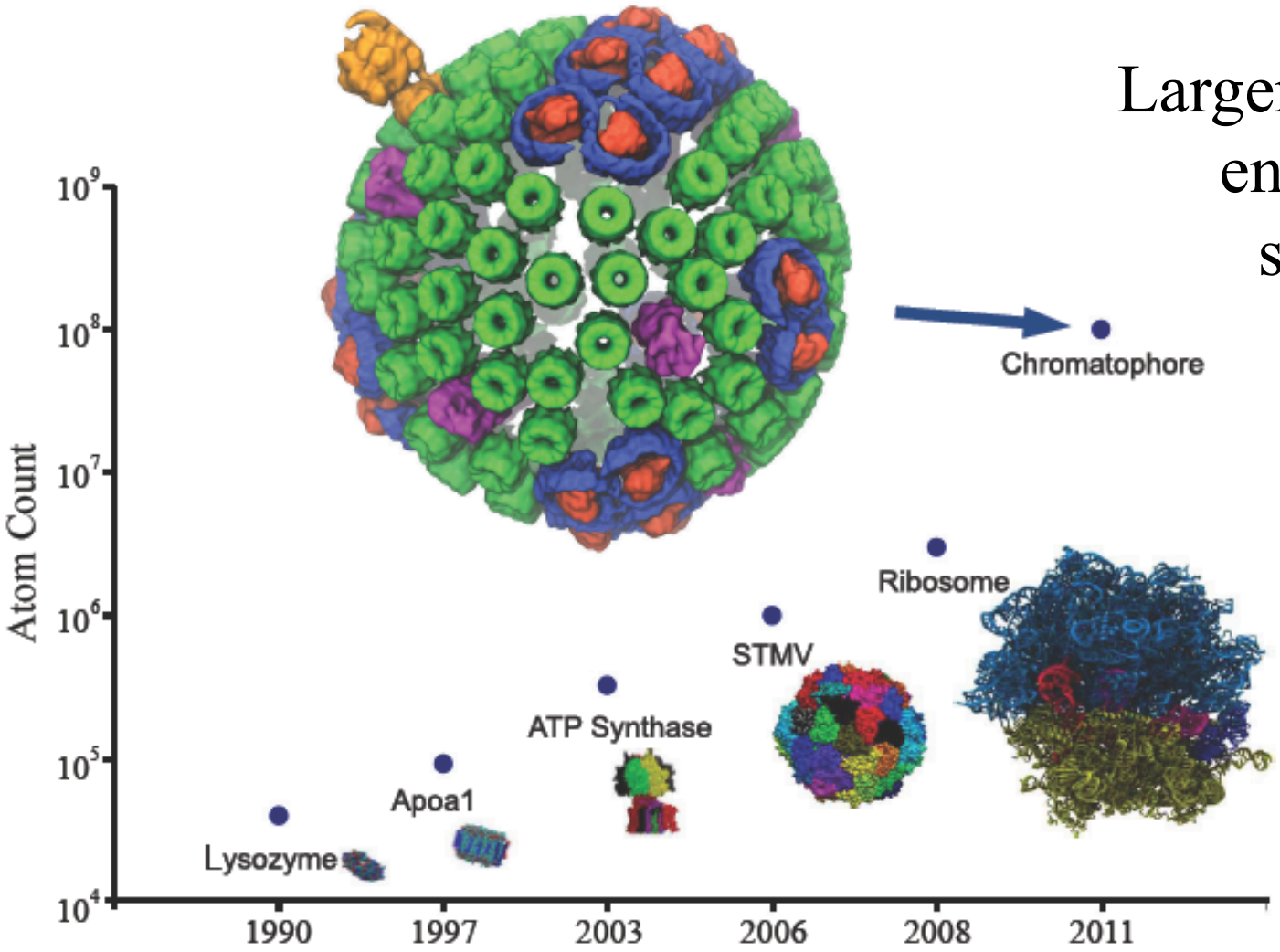
Titan

- Funded by DOE
- Allocated by INCITE, etc.
- NCCS (Oak Ridge)
- 18,688 XK7 compute nodes
- 8,972 GPUs as of last week, other half of machine down

Blue Waters

- Funded by NSF
- Allocated by PRAC
- NCSA (U. Illinois)
- 22,000 XK6 compute nodes + 3,000 XK7 compute nodes
- Available to “friendly users”

Larger machines enable larger simulations



Beckman Institute, UIUC

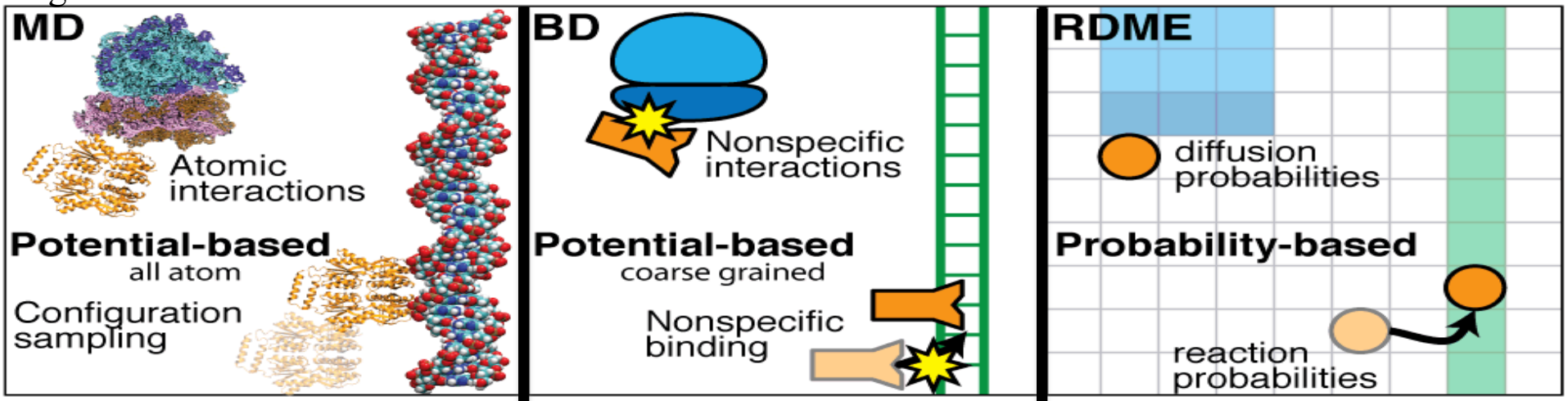

NIH BTRC for Macromolecular Modeling and Bioinformatics
1990-2017
Beckman Institute
University of Illinois at
Urbana-Champaign

Physics of in vivo Molecular Systems

Biomolecular interactions span many orders of magnitude in space and time.

femtoseconds Center software provides multi-scale computational modeling. hours

Ångstrom microns



NAMD Scalable Molecular Dynamics

MDFF Molecular Dynamics Flexible Fitting

HMMM Highly Mobile Membrane Mimetic

VMD Visual Molecular Dynamics

BrownianMover Brownian Dynamics

VMD Visual Molecular Dynamics

NAMD Scalable Molecular Dynamics

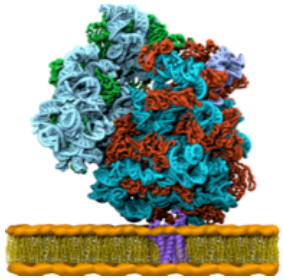
BD: Brownian Dynamics

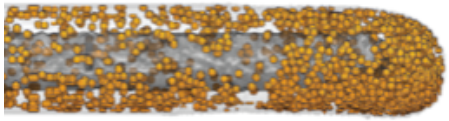
LatticeMicrobes Whole Cell Simulations

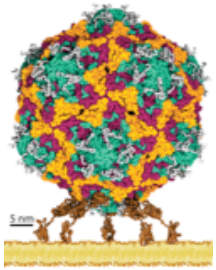
VMD Visual Molecular Dynamics

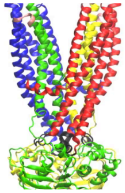
RDME: Reaction-diffusion master equation

Collaborative Driving Projects

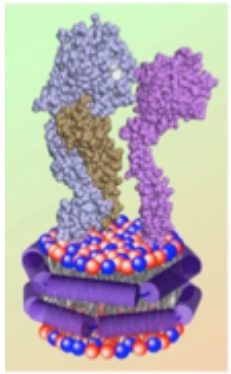


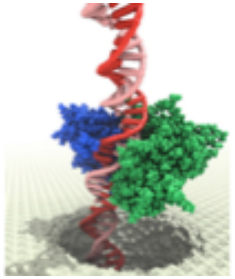


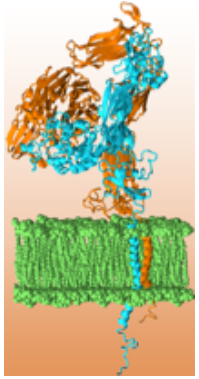




1. Ribosome	R. Beckmann (U. Munich) J. Frank (Columbia U.) T. Ha(UIUC) K. Fredrick (Ohio state U.) R. Gonzalez (Columbia U.)
2. Blood Coagulation Factors	J. Morrissey (UIUC) S. Sligar (UIUC) C. Rienstra (UIUC) G. Gilbert (Harvard)
3. Whole Cell Behavior	W. Baumeister (MPI Biochem.) J. Xiao (Johns Hopkins U.) C.N. Hunter (U. Sheffield) N. Price (U. Washington)
4. Biosensors	R. Bashir (UIUC) J. Gundlach (U. Washington) G. Timp (U. Notre Dame) M. Wanunu (Northeastern U.) L. Liu (UIUC)
5. Viral Infection Process	J. Hogle (Harvard U.) P. Ortoleva (Indiana U.) A. Gronenborn (U. Pittsburgh)
6. Integrin	T. Ha (UIUC) T. Springer (Harvard U.)
7. Membrane Transporters	H. Mchaourab (Vanderbilt U.) R. Nakamoto (U. Virginia) D.-N. Wang (New York U.) H. Weinstein (Cornell U.)

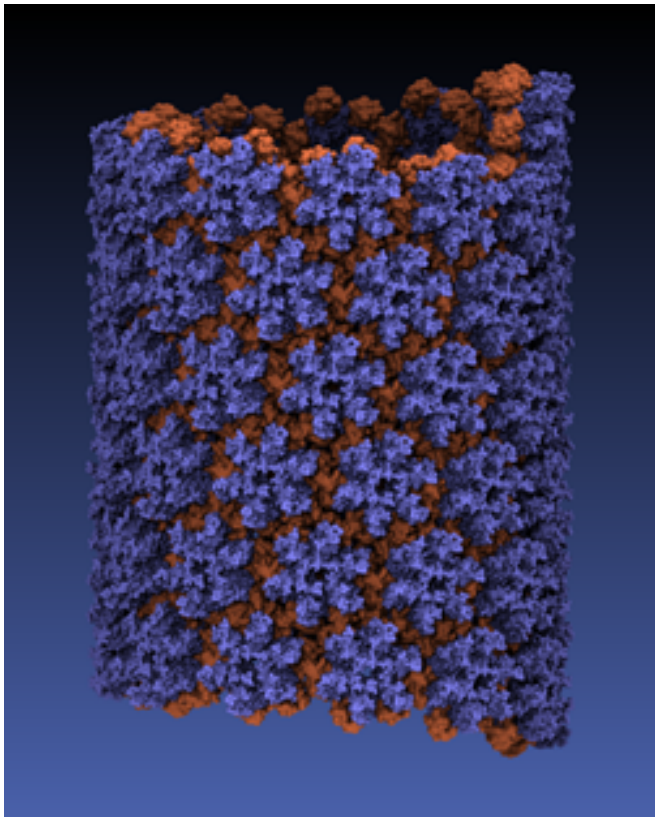




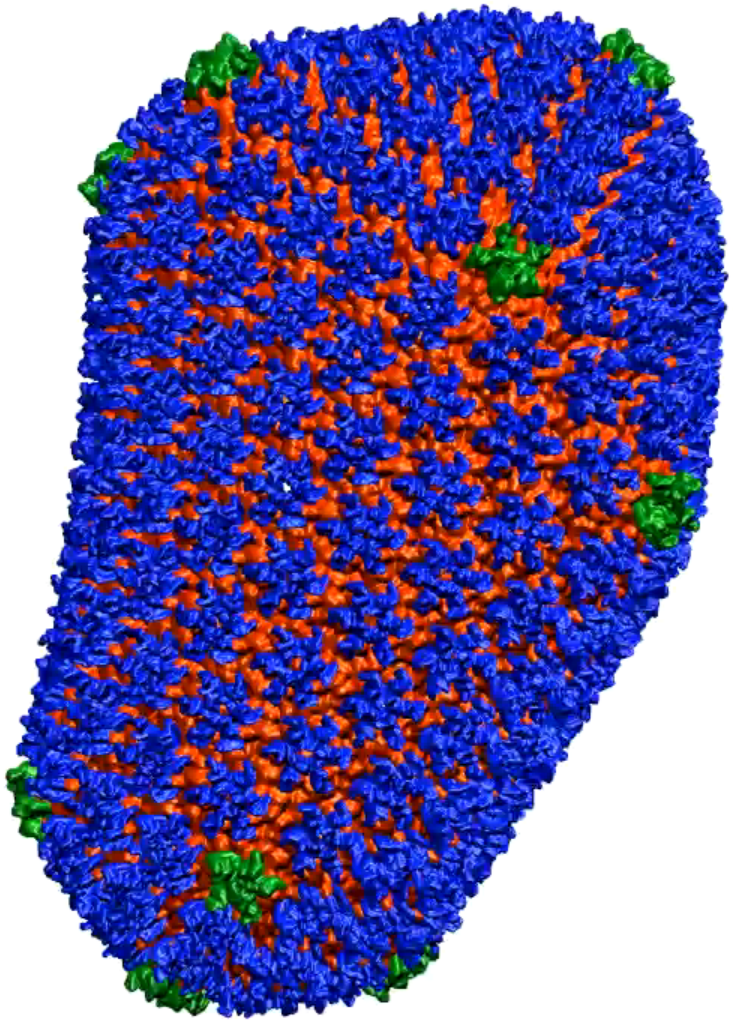


2012: Blue Waters Early Science Project

"The first all-atom structure of an **HIV virus capsid** in its tubular form, courtesy Klaus Schulten, University of Illinois at Urbana-Champaign Theoretical and Computational Biophysics Group/ Beckman Institute; Angela Gronenborn and Peijun Zhang, University of Pittsburgh School of Medicine Center for HIV Protein Interactions/Department of Structural Biology."



2013:



GRID VCA: Take our money!

- Our group has spent several years assembling storage, analysis, and visualization hardware to prepare for Blue Waters.
  - Total of 6 high-end public desktops with 72GB plus Fermi Quadroplex.
  - Still upgrading building network to 10Gbit/s.
- Just last week, VMD user needed to render an image too large for visualization host, needed hours to write scene file to NFS, an hour to read scene, five minutes to actually render scene on 256GB server.
- One GRID VCA could provide higher peak memory to any office.
  - In the same rack as storage server on a local switch.
  - Even better, directly attached to the Blue Waters Lustre storage system.
- Highly useful capability for any supercomputing center.
  - This is not an endorsement. I only heard about it this morning.

Parallel Programming Lab

University of Illinois at Urbana-Champaign

Siebel Center for Computer Science

http://charm.cs.illinois.edu/

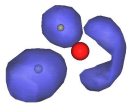
GTC 2013

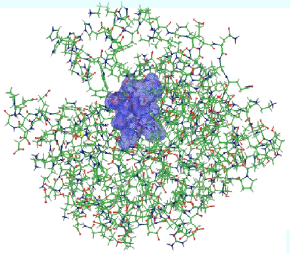
Biomedical Technology Research Center for Macromolecular Modeling and Bioinformatics
http://www.ks.uiuc.edu/

Beckman Institute, UIUC

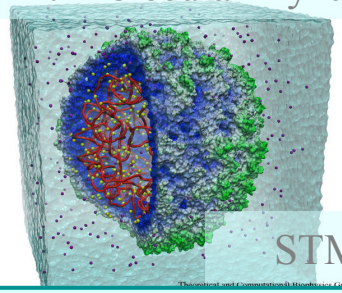
Develop abstractions in context of full-scale applications

Quantum Chemistry (QM/MM)

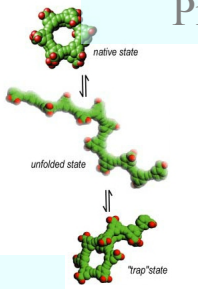




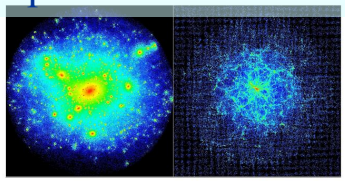
NAMD: Molecular Dynamics



Protein Folding



Computational Cosmology

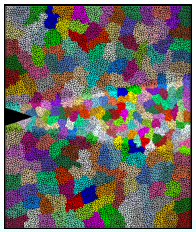


STM virus simulation

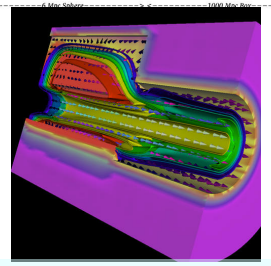
Parallel Objects,

Adaptive Runtime System

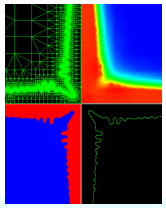
Libraries and Tools



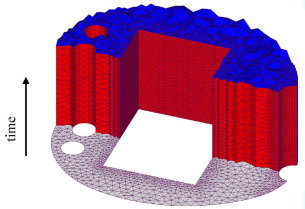
Crack Propagation



Rocket Simulation



Dendritic Growth



Space-time meshes

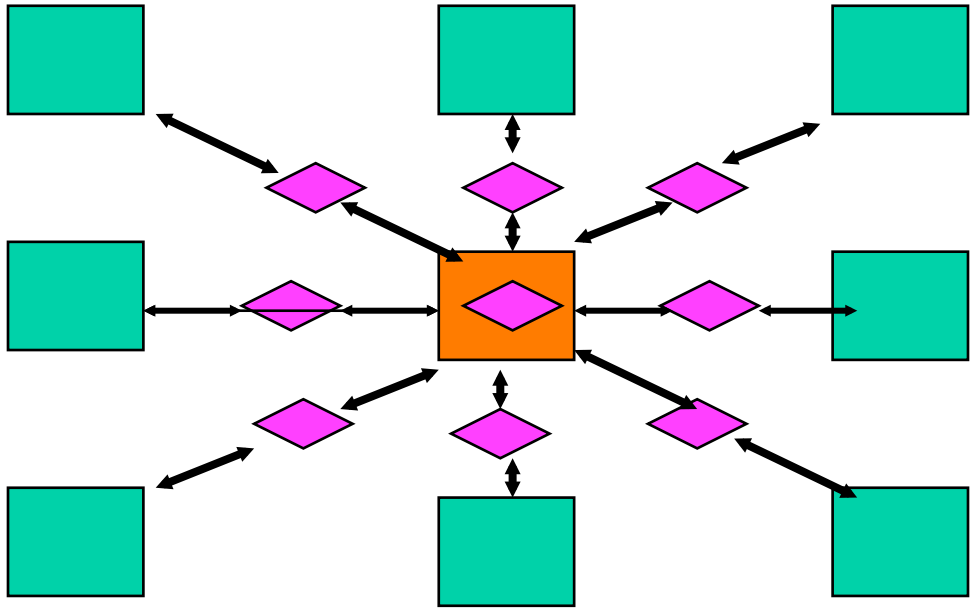
The enabling CS technology of parallel objects and intelligent Runtime systems has led to several collaborative applications in CSE

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

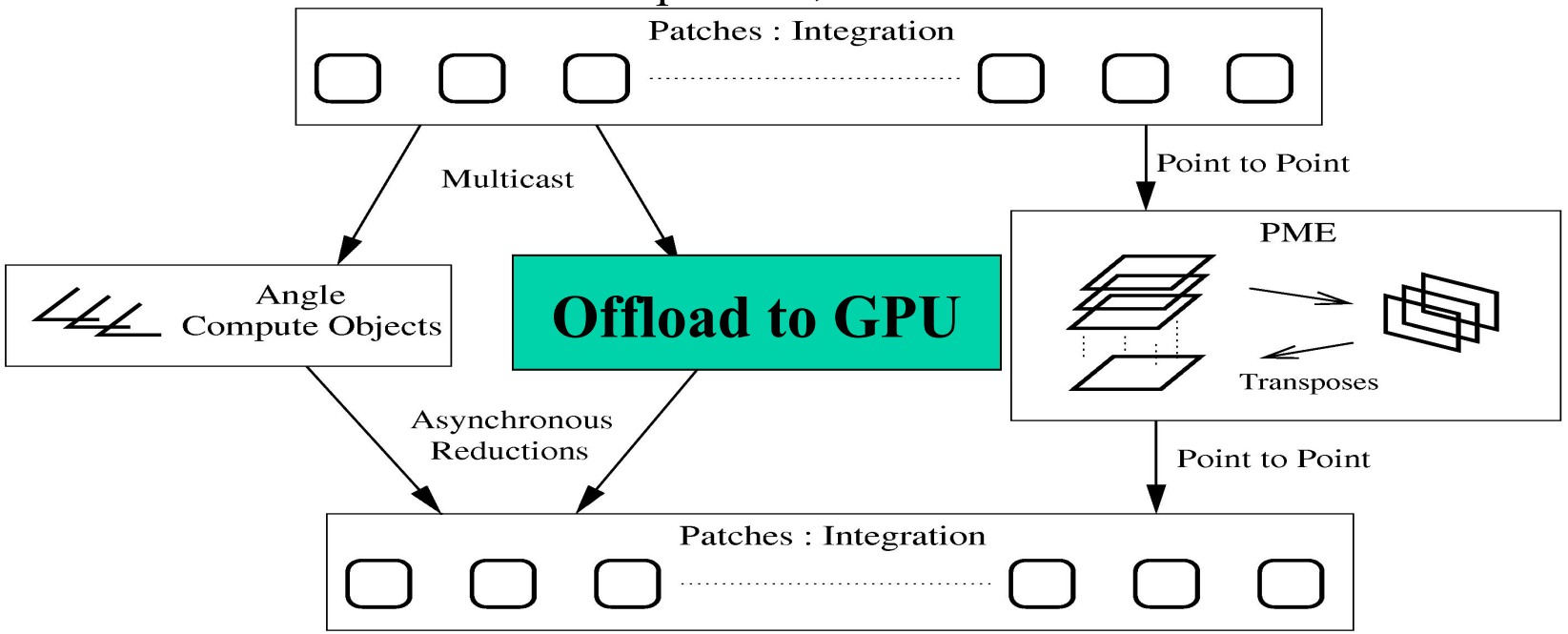
Kale *et al., J. Comp. Phys.* **151**:283-312, 1999.



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

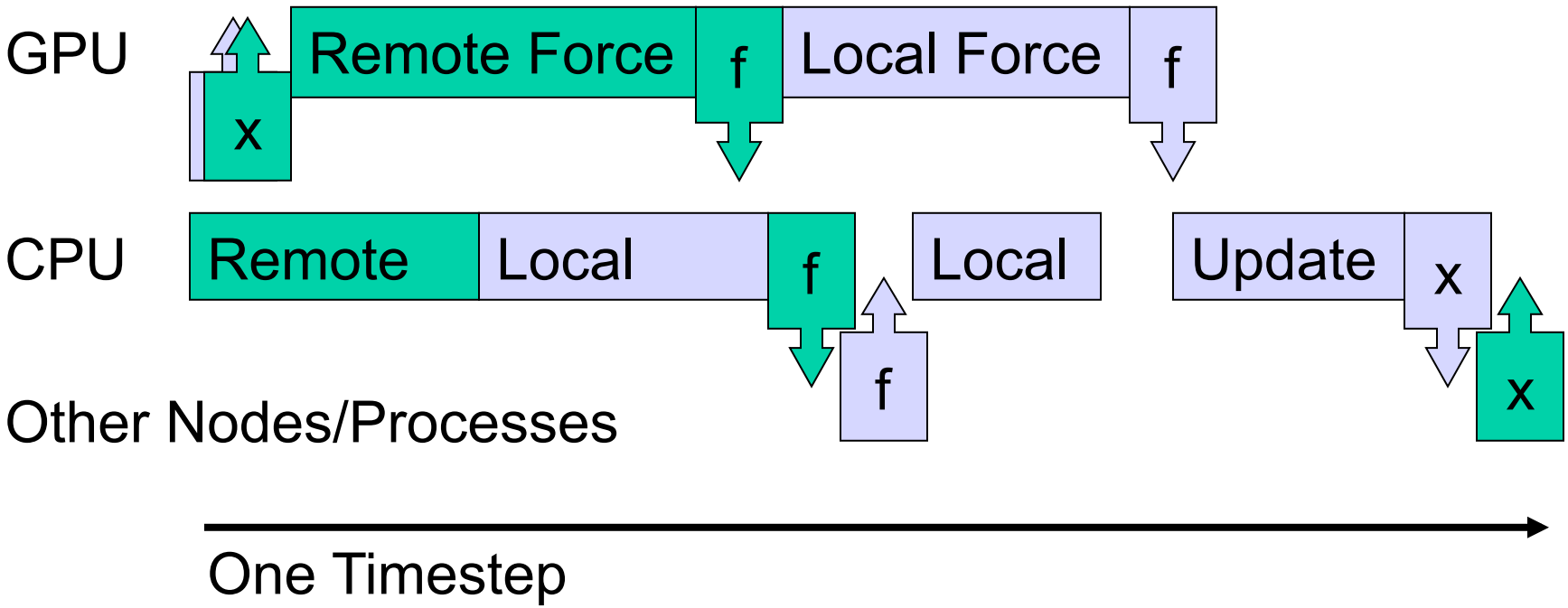
NAMD Overlapping Execution

Phillips *et al., SC2002.*



Objects are assigned to processors and queued as data arrives.

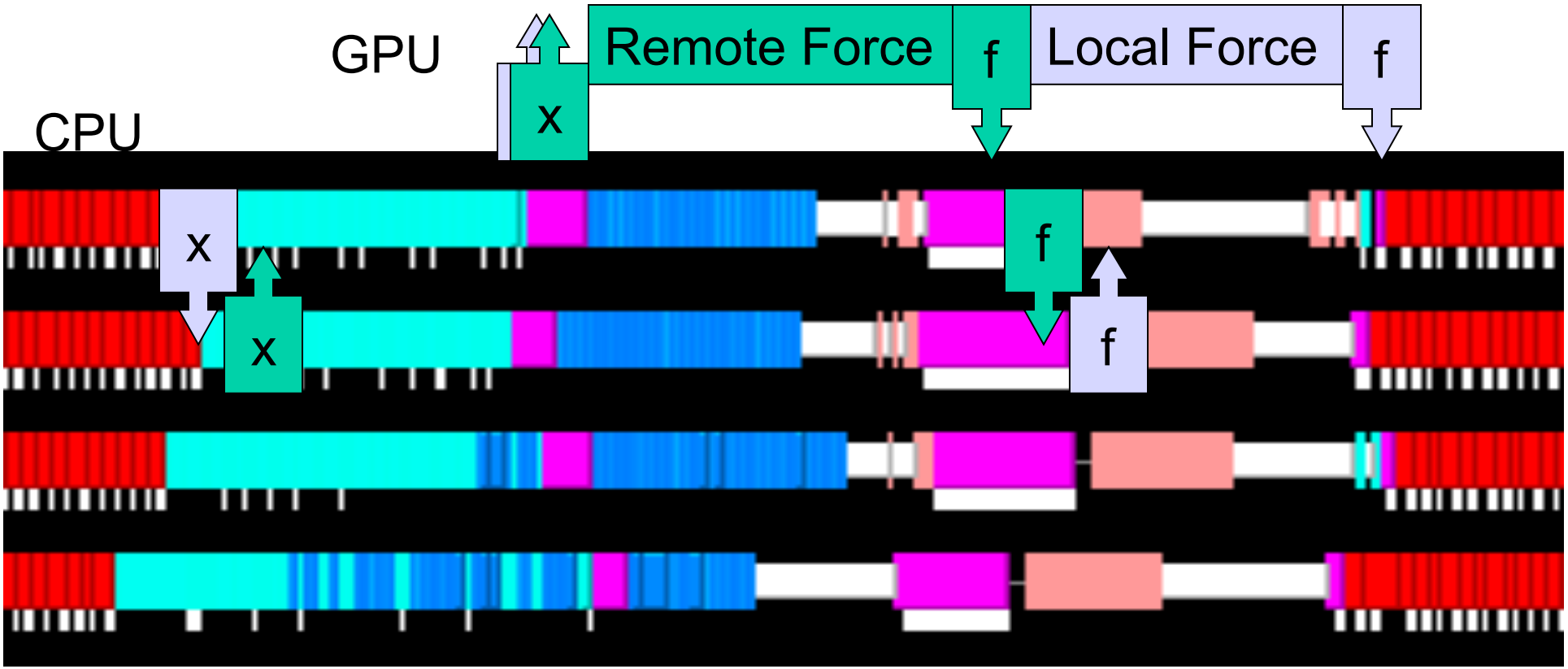
Overlapping GPU and CPU with Communication



Phillips *et al.*, SC2008

Actual Timelines from NAMD

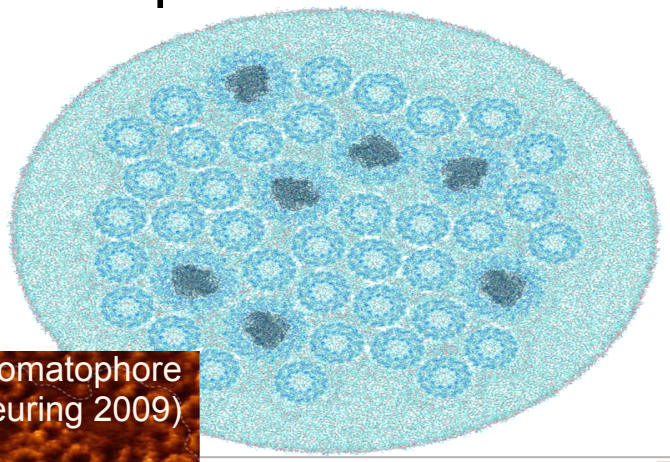
Generated using Charm++ tool “Projections” http://charm.cs.uiuc.edu/

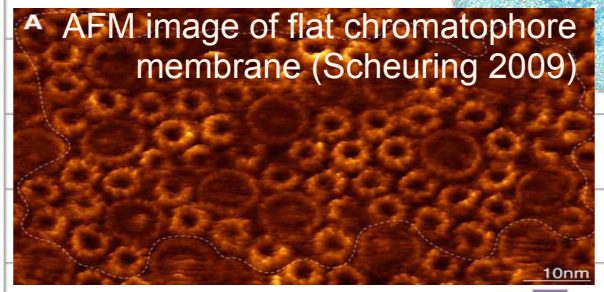


Tsubame (Tokyo) Application of GPU Accelerated NAMD (fall 2011)

20 million atom proteins + membrane

8400 cores

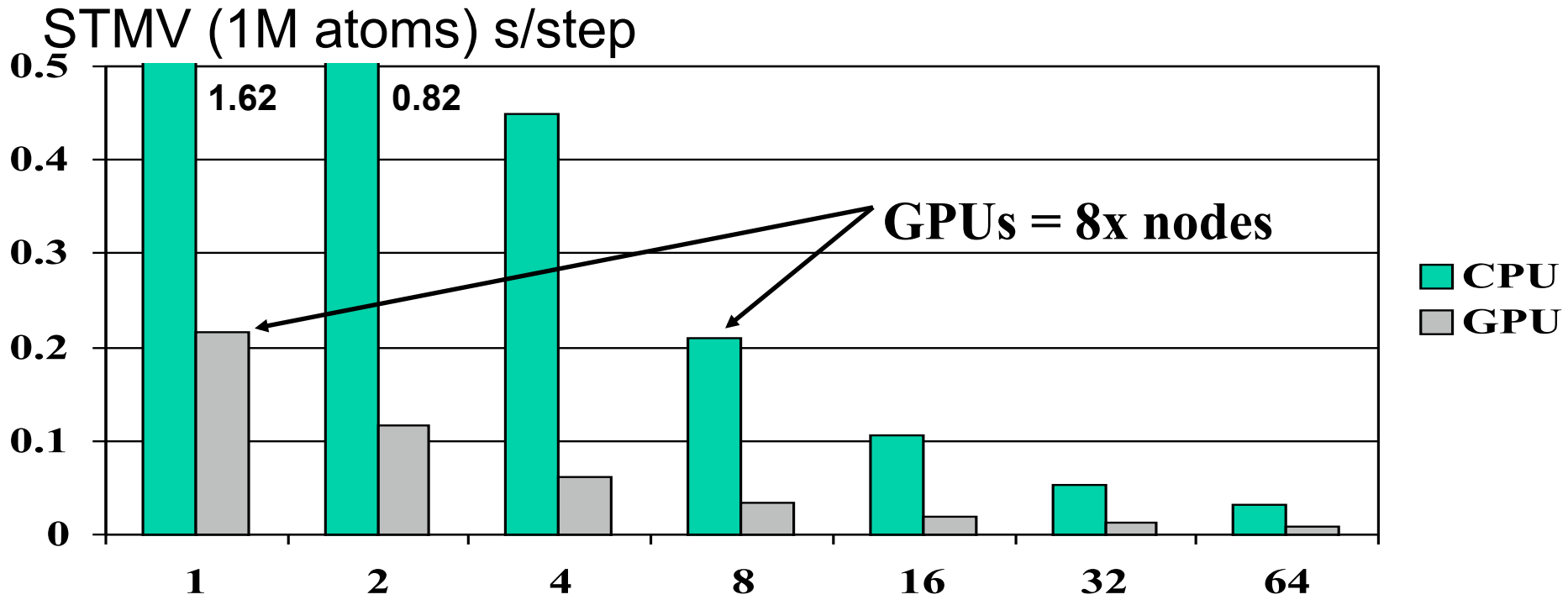




	4nodes	8nodes	32nodes	64nodes	128nodes	256nodes	512nodes	700nodes
CPU 12 cores	0.02	0.04	0.14	0.23	0.41	0.92	1.69	2.23
CPU 12 cores + 1 GPU	N/A	N/A	0.42	0.80	1.50	2.93	5.01	5.98
CPU 12 cores + 2 GPUs	N/A	N/A	0.62	1.17	2.00	3.68	6.18	7.38
CPU 12 cores + 3 GPUs	0.10	0.19	0.65	1.25	2.11	3.93	6.42	8.00

NAMD 2.9 on Keeneland ID

(12 Intel cores and 3 Telsa M2070 GPUs per node)



Biomedical Technology Research Center for Macromolecular Modeling and Bioinformatics
http://www.ks.uiuc.edu/

Beckman Institute, UIUC

Trends Affecting Performance

- GPU performance increasing
  - Performance limit will be code on CPU
  - Most highly tuned CPU code moved to GPU
  - Remaining CPU code is also less efficient
  - Therefore CPU must run serial code well
- CPU serial performance static
- CPU core counts increasing

Suggested Strategy

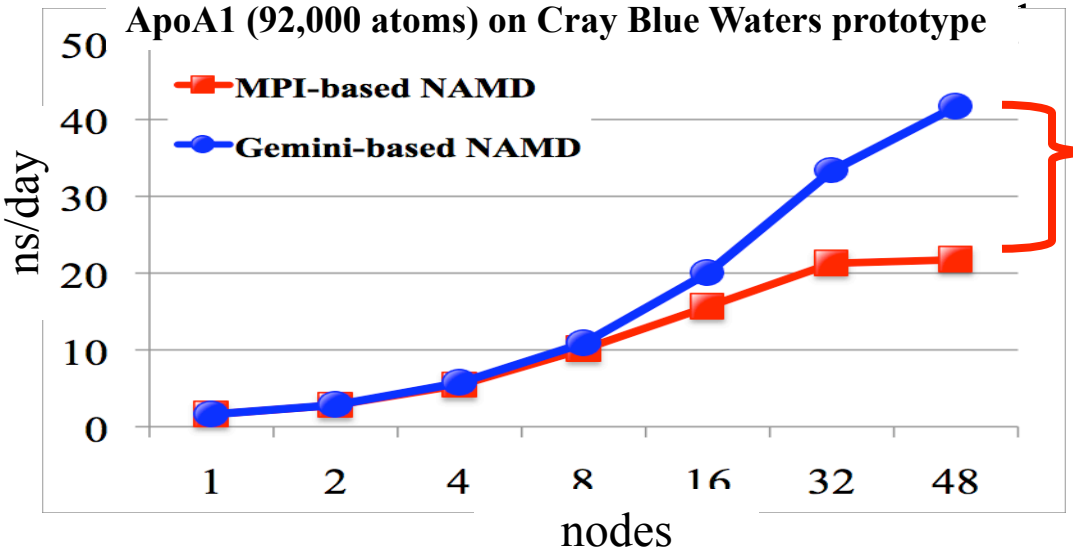
- Focus on CPU-side code
  - Port to GPU or optimize/paralellize on CPU
  - Stream results off GPU to increase overlap
  - Use CPUs with best single-thread performance
- Focus on communication
  - Reduce communication overhead on CPU
  - Deal with multithreaded MPI issues
  - General parallel scalablity improvements

Streaming GPU Results to CPU

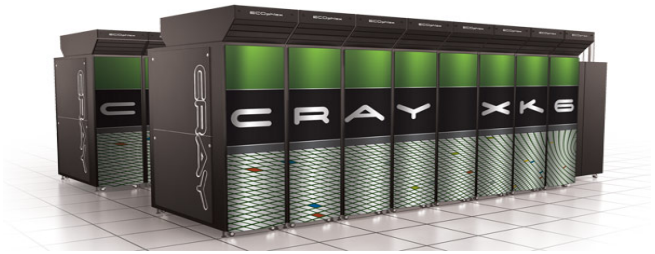
- Allows incremental results from a single grid to be processed on CPU before grid finishes on GPU
- GPU side:
  - Write results to host-mapped memory
  - __threadfence_system() and __syncthreads()
  - Atomic increment for next output queue location
  - Write result index to output queue
- CPU side:
  - Poll end of output queue (int array) in host memory

Cray Gemini Optimization

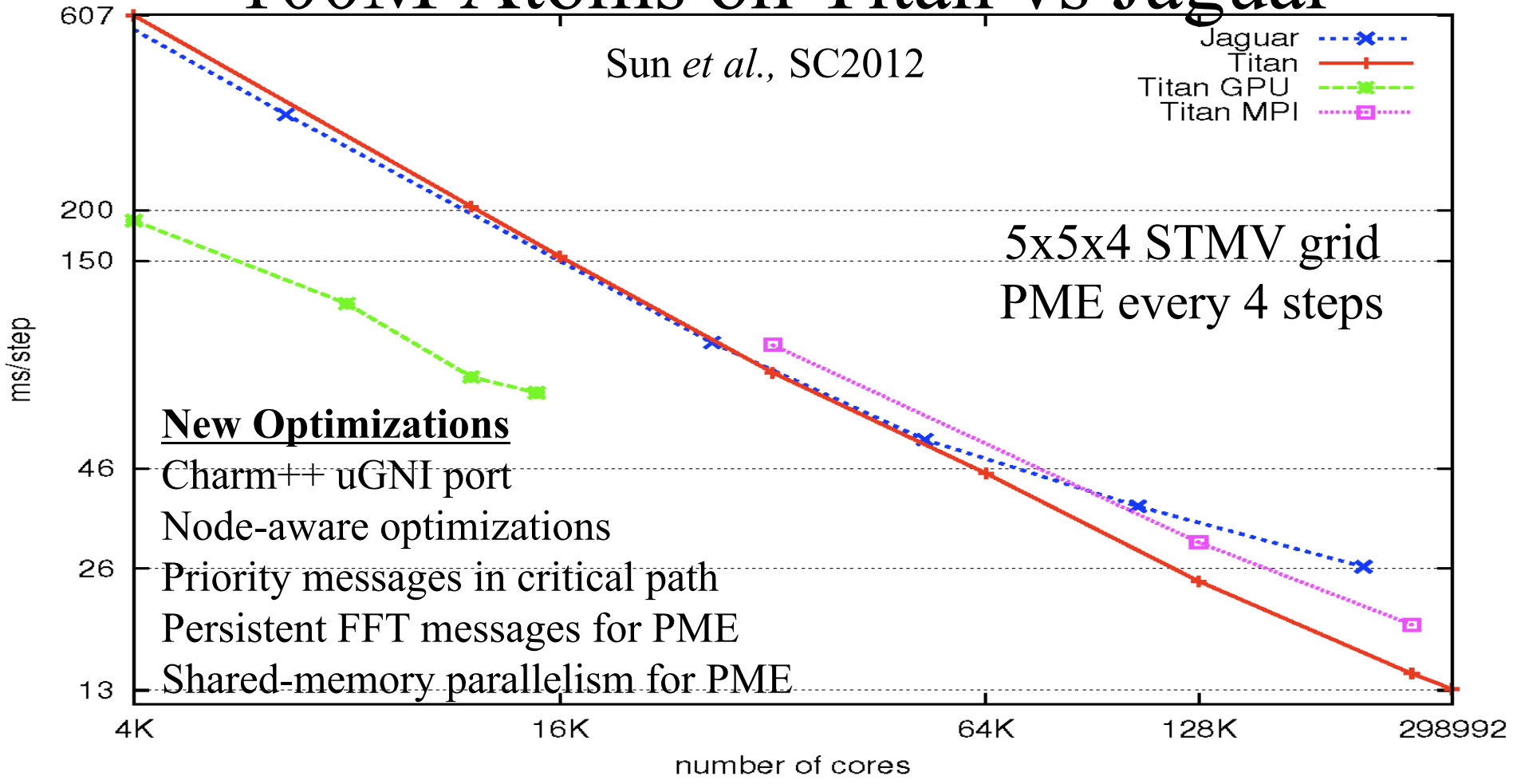
- The new Cray machine has a better network (called Gemini)
- MPI-based NAMD scaled poorly
- BTRC implemented direct port of Charm++ to Cray
  - *uGNI* is the lowest level interface for the Cray Gemini network



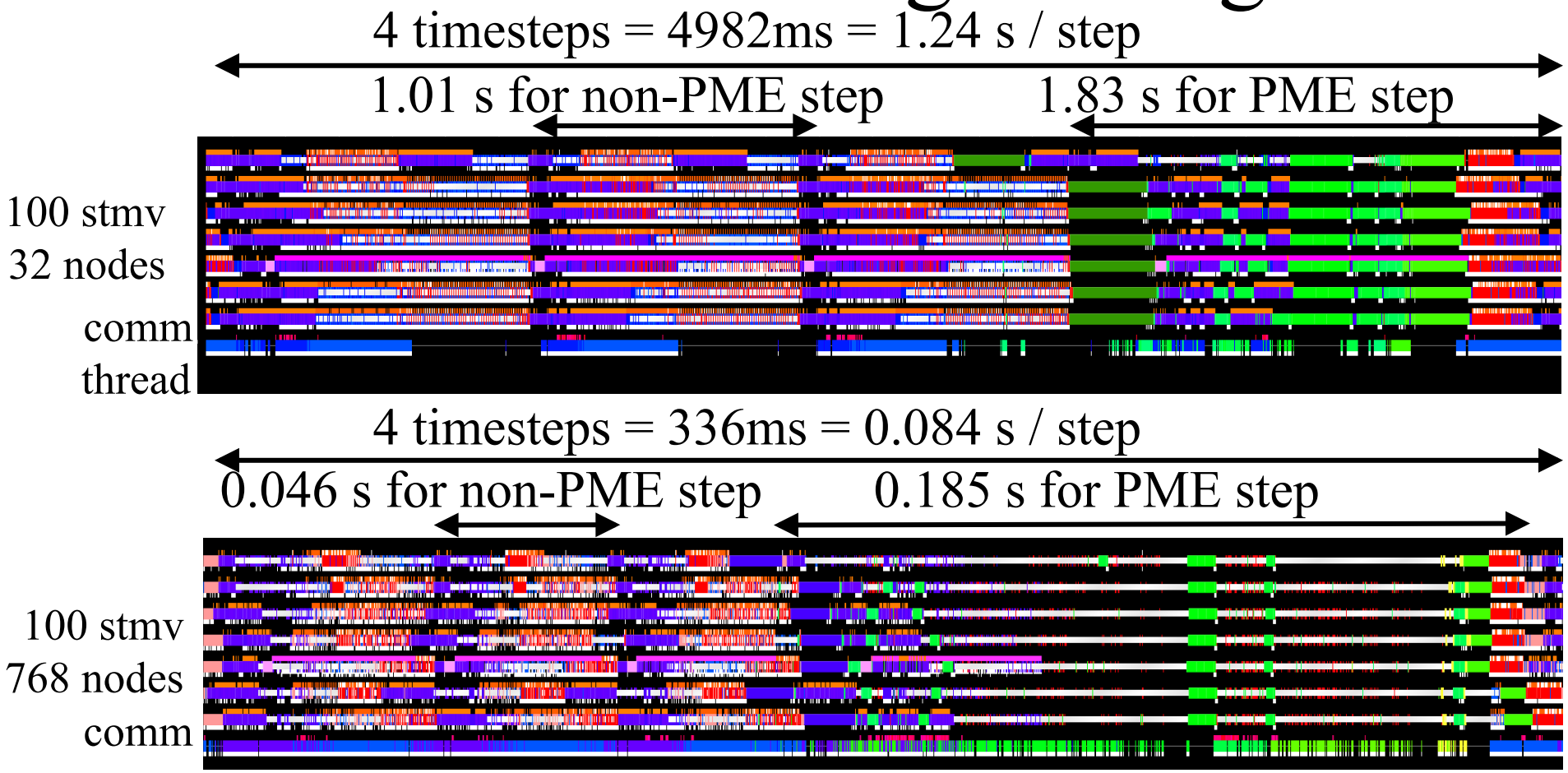
Gemini provides at least 2x increase in usable nodes for strong scaling



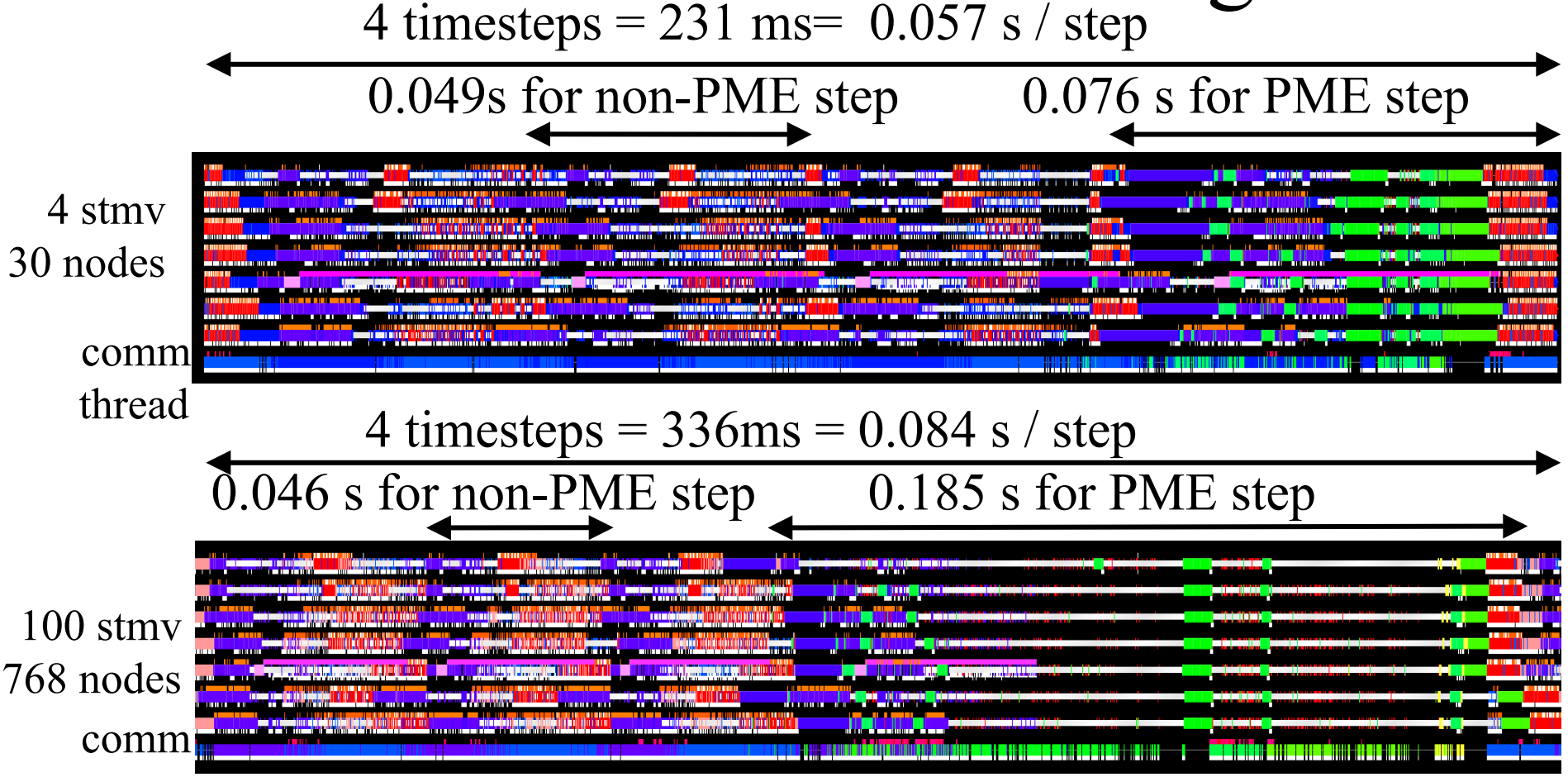
100M Atoms on Titan vs Jaguar



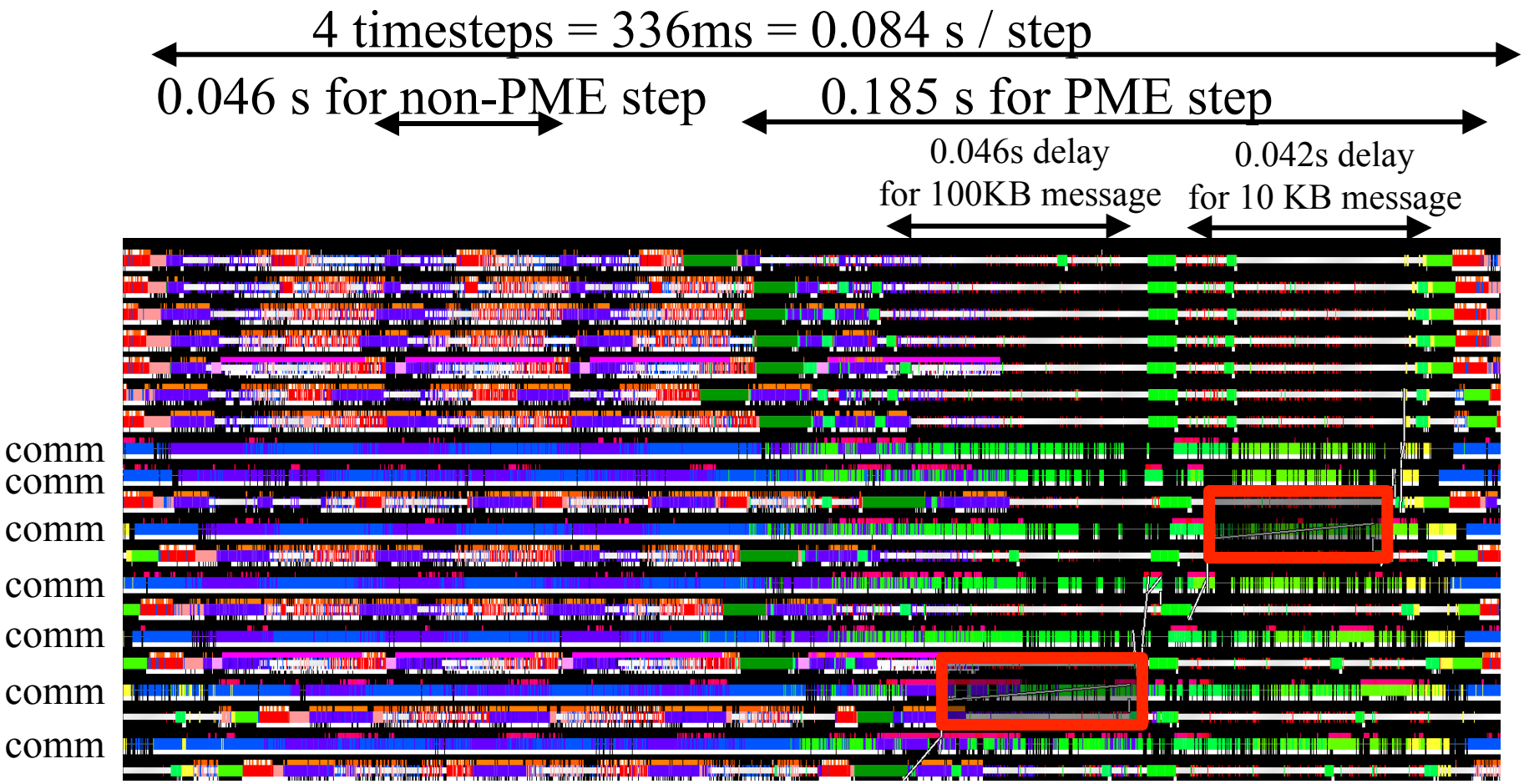
TitanDev Strong Scaling



TitanDev Weak Scaling

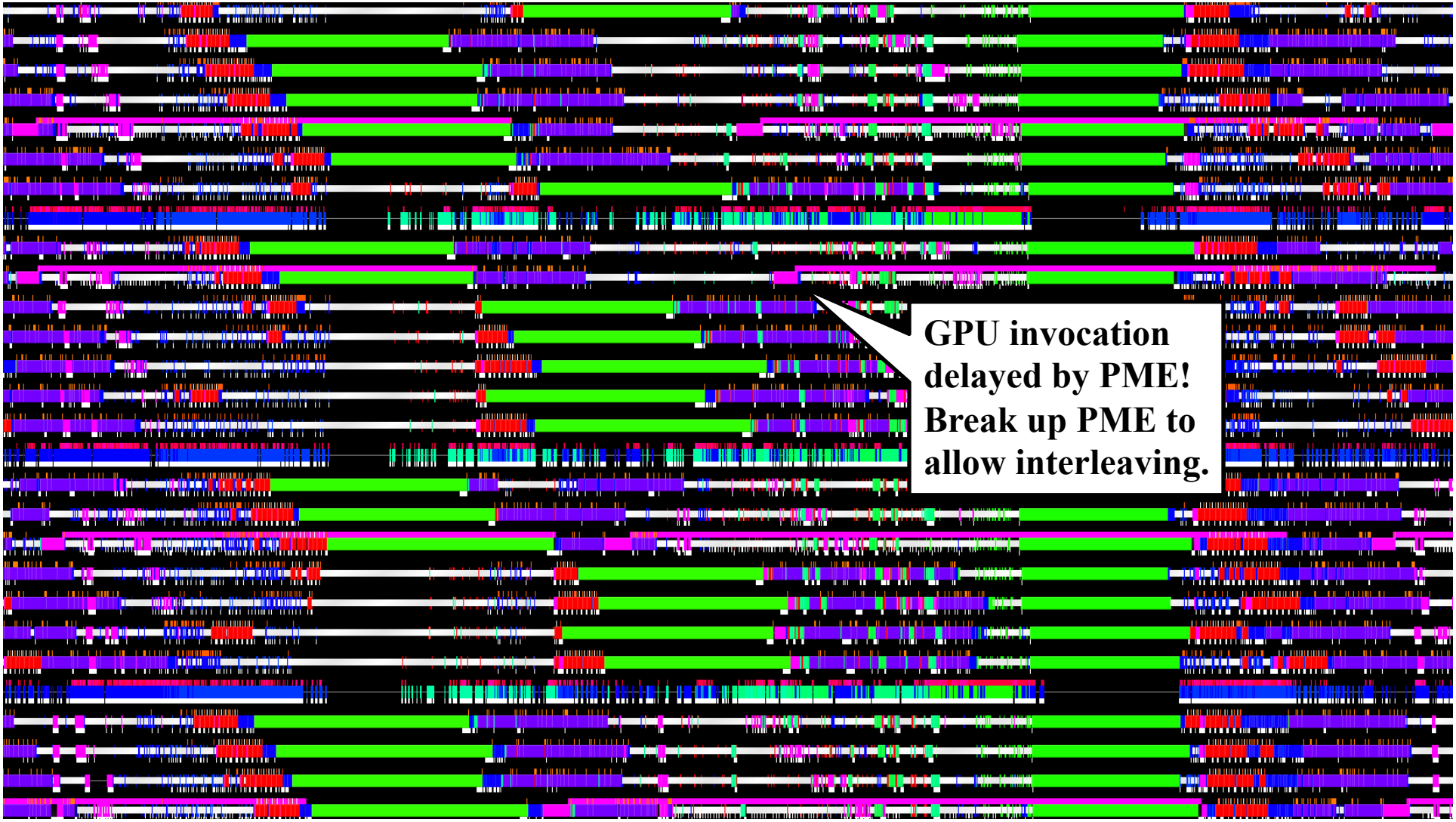


PME delays – tracing data needed for one ungrid calculation

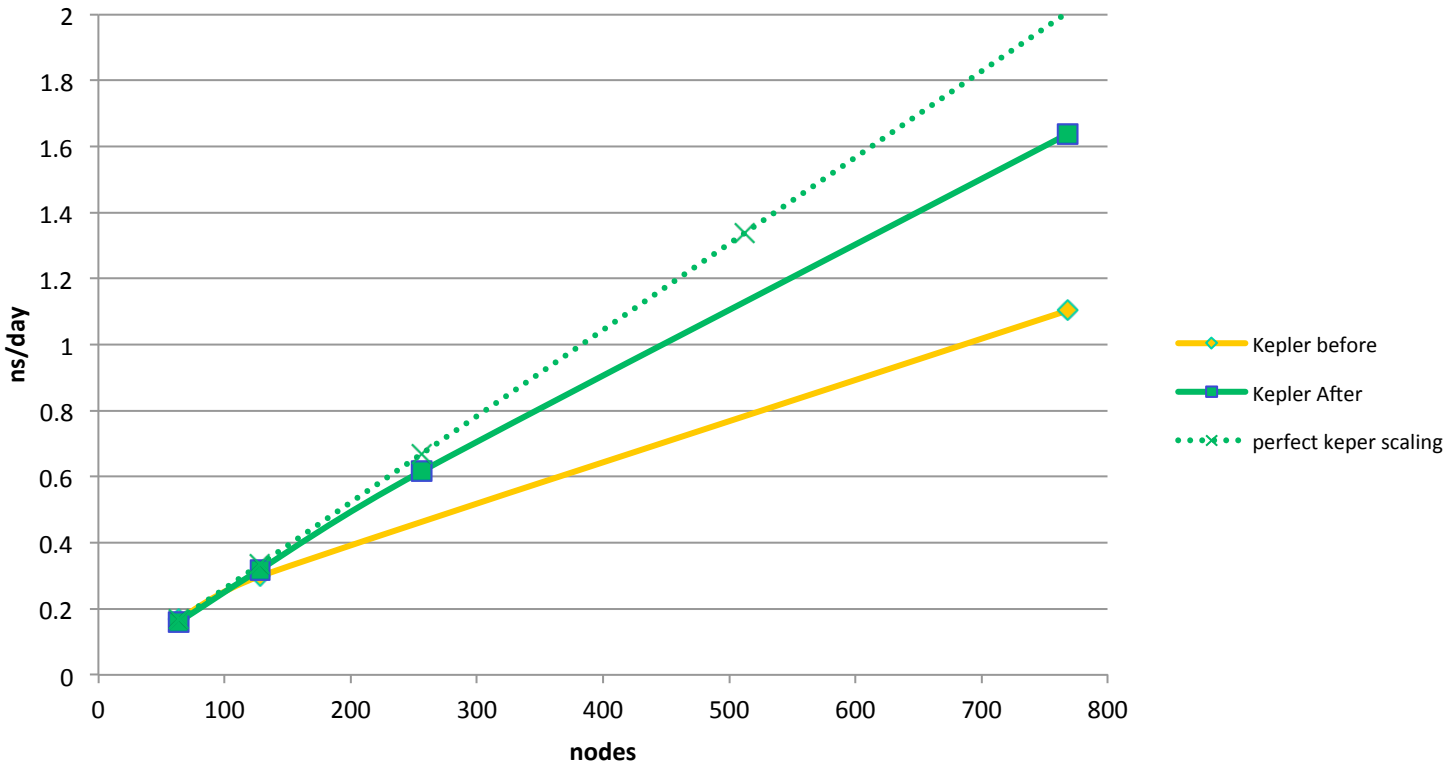


Strategy to improve scalability

- Fix issues with communication
  - 23x16x2 topology limits bisection bandwidth
- Coarsen PME grid with higher-order interpolation
  - Reduces communication (factor of 8)
  - Does not increase short-range work or communication
- Push PME work to the GPU
  - Charge gridding overlaps coordinate receive
- Start GPU work sooner
  - Currently waiting for all coordinate receives
  - Use streams to launch work as data arrives


GPU invocation
delayed by PME!
Break up PME to
allow interleaving.

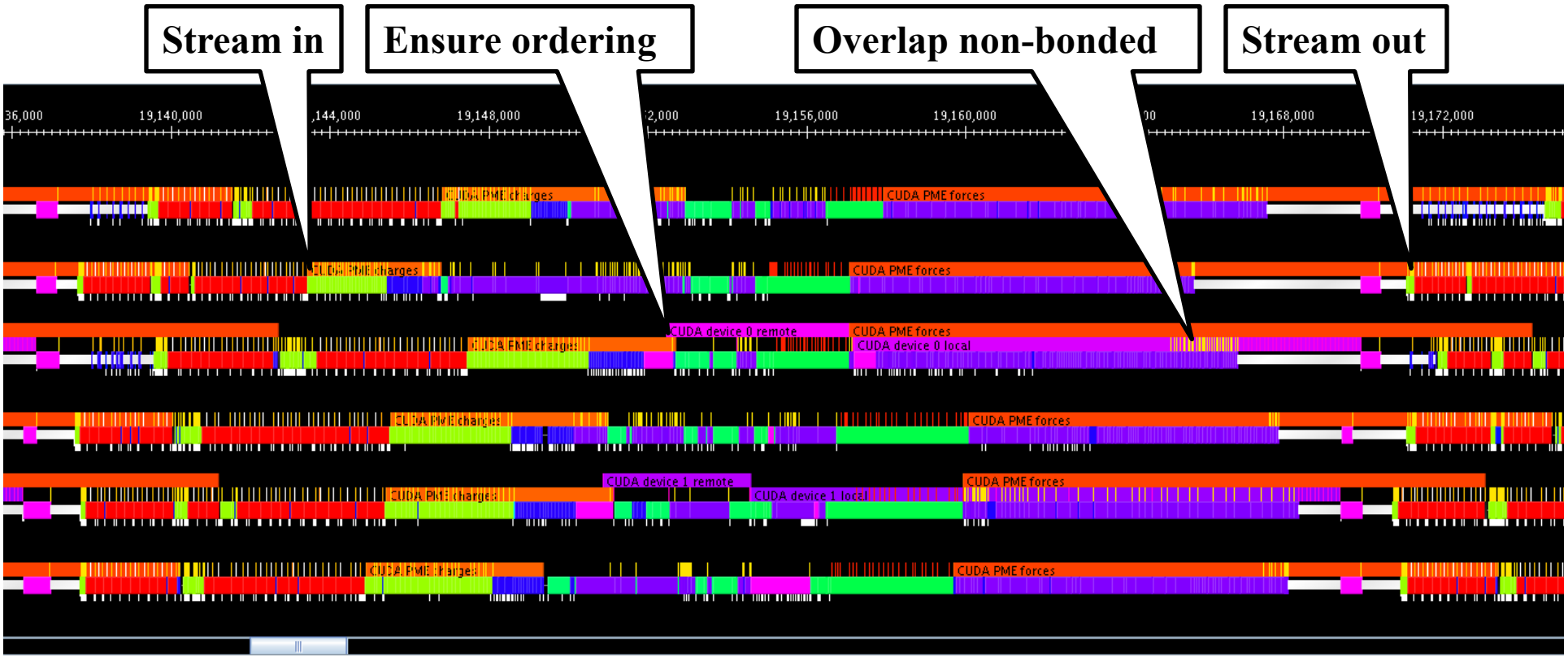
Effect of Coarsening PME Grid

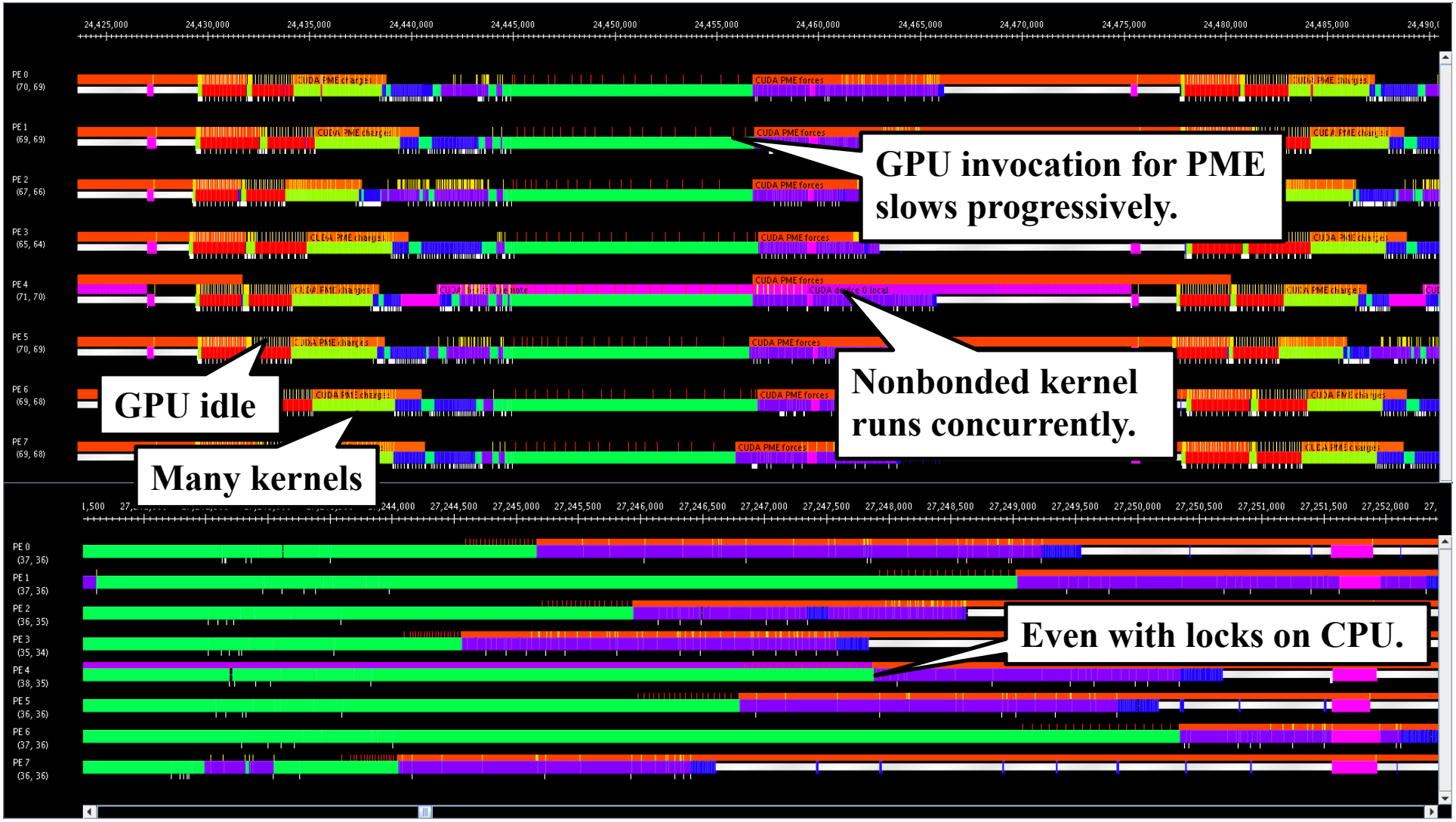


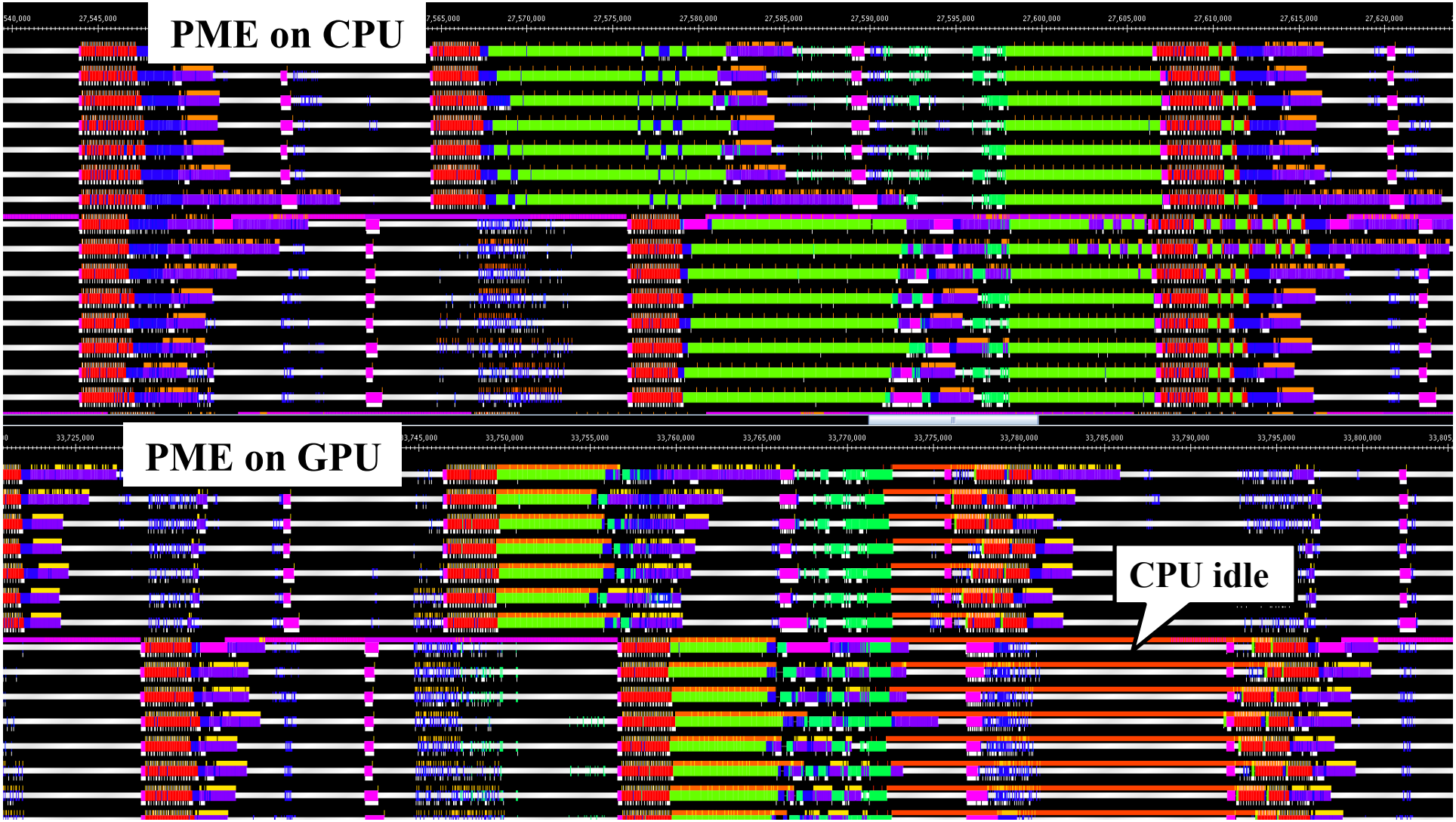
NAMD PME CUDA Kernel

- CPU may be bottleneck for higher-order PME
  - Especially once the Kepler non-bonded kernel is finished…
- Target Kepler, test new features
- Simplest design that might possibly work:
  - One stream per host PE (preserve control flow)
  - One atom per warp with warp-synchronous programming
    - Failed with old-style volatile __shared__, had to add __syncthreads()
  - 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

NAMD PME CUDA Kernel




GPU invocation for PME slows progressively.
Nonbonded kernel runs concurrently.
GPU idle
Many kernels
Even with locks on CPU.


PME on CPU
PME on GPU
CPU idle

PME CUDA Kernel Plans

- Single charge/potential grid per node, not per host PE
  - Aggregates data before send, reduces inter-node messages
  - Would require coordination on CPU but trivial on GPU
- Dynamic Parallelism and GPUDirect
  - Data-dependent packing for inter-node messages on GPU
- Shuffle instructions for warp-synchronous programming
  - Somewhat harder to code than shared memory
  - Would be simpler with __warp_shared__ and __syncwarp()
  - Also has applications in nonbonded kernel

Results: Topology Matters

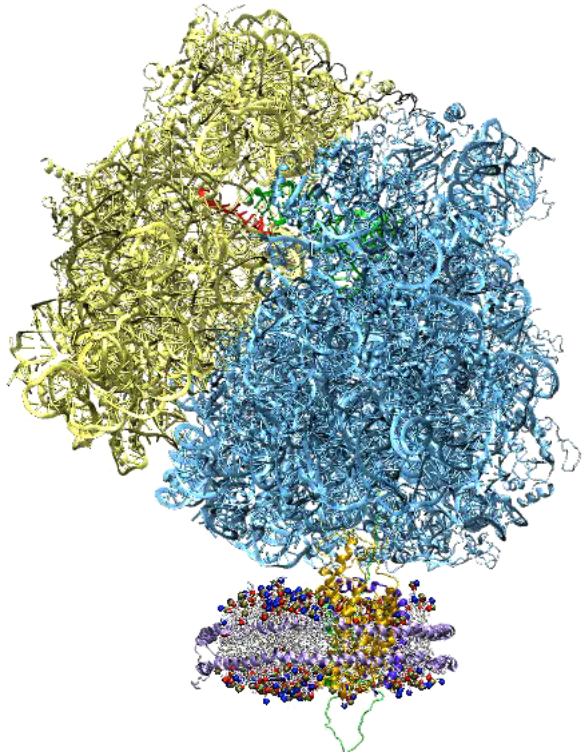
- Running experiments on Blue Waters GPU nodes (3000)
  - 100M atoms, 7 pes and one communication thread per node
  - Scales to 1000 nodes (50 ms/step), same at 2000, slower on 3000
  - Lucky runs hit 30 ms/step, can't reproduce experiments
  - Runs faster with PME on CPU at these node counts
- Speculation:
  - Interference from other jobs running on machine
  - Similar slowdowns also seen during large CPU runs
- Conclusions:
  - Traditional space-sharing may not work for torus on XE6/XK7
  - Topology-aware scheduling (like Blue Gene) would likely help

A Smaller Driving Project: The Ribosome

Target of over 50%
of antibiotics

Many related diseases. e.g. Alzheimer's disease due to dysfunctional ribosome (J. Neuroscience 2005, 25:9171-9175)

Localization failure of nascent chain lead to neurodegenerative disease (Mol. Bio. of the Cell 2005, 16:279-291)



NAMD 2.9 Scalable Replica Exchange

- Easier to use *and* more efficient:
  - Eliminates complex, machine-specific launch scripts
  - Scalable pair-wise communication between replicas
  - Fast communication via high-speed network
- Basis for many enhanced sampling methods:
  - Parallel tempering (temperature exchange)
  - Umbrella sampling for free-energy calculations
  - Hamiltonian exchange (alchemical or conformational)
  - Finite Temperature String method
  - Nudged elastic band

} Released in NAMD 2.9

- Great power *and* flexibility:
  - **Enables petascale simulations of modestly sized systems**
  - Leverages features of Collective Variables module
  - Tcl scripts can be highly customized and extended

NAMD 2.10 Scalable Replica Exchange

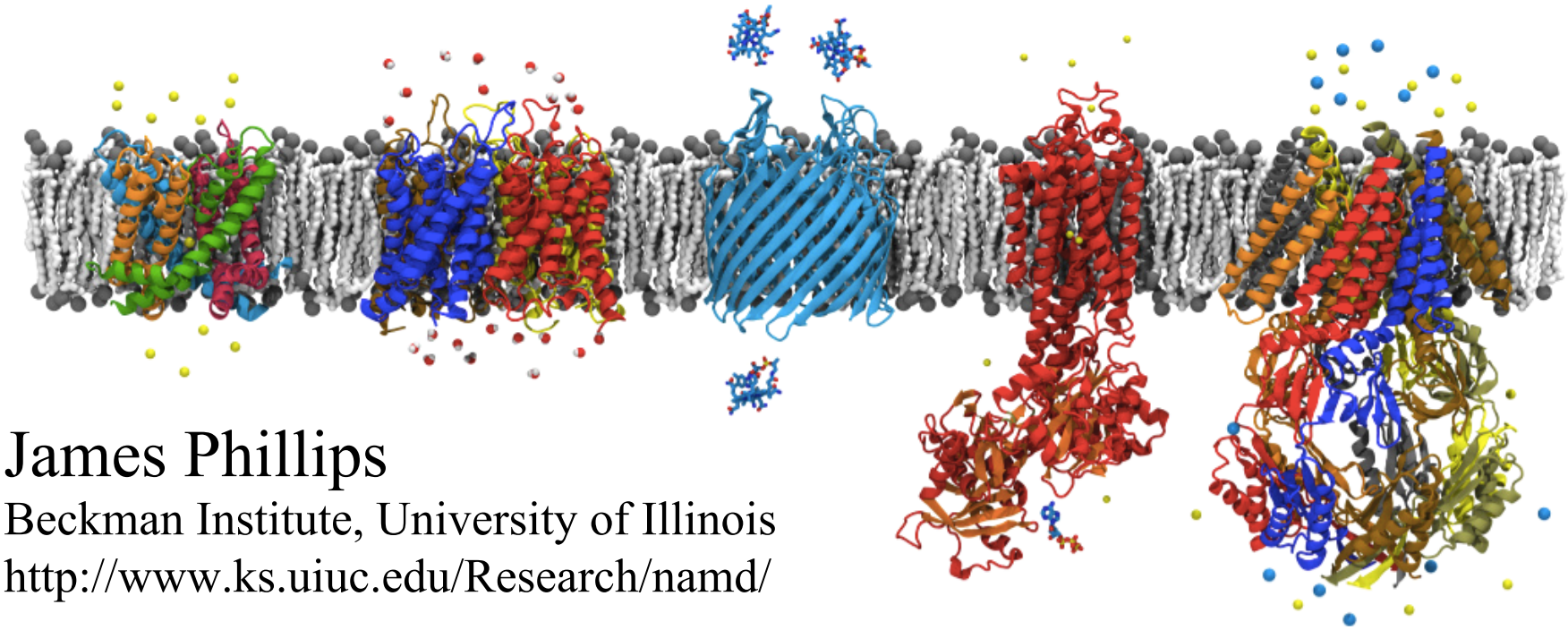
- More general Charm++ integration:
  - NAMD 2.9 used MPI communicator splitting
  - NAMD 2.10 splits replicas in Converse low-level runtime (LRTS)
  - LRTS underlies MPI, Cray (uGNI), and BlueGene/Q (PAMI) implementations
- Basis for many enhanced sampling methods:
  - Parallel tempering (temperature exchange)
  - Umbrella sampling for free-energy calculations
  - Hamiltonian exchange (alchemical or conformational)
  - Finite Temperature String method
  - Nudged elastic band

Same Tcl scripts as NAMD 2.9

Future work enabled by Charm++ integration

- Better scaling for individual replicas:
  - **Cray uGNI layer essential for multi-node GPU replicas**
  - IBM BlueGene/Q will benefit similarly from PAMI layer
  - Porting native InfiniBand (ibverbs) layer to LRTS

Thanks to: NIH, NSF, DOE, NCSA,
NVIDIA (**Sarah Tariq**, Sky Wu, Justin Luitjens, Nikolai Sakharnykh),
Cray (Sarah Anderson, Ryan Olson), NCSA (Robert Brunner),
PPL (Eric Bohm, Yanhua Sun, Gengbin Zheng, Nikhil Jain)
and 18 years of NAMD and Charm++ developers and users.



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