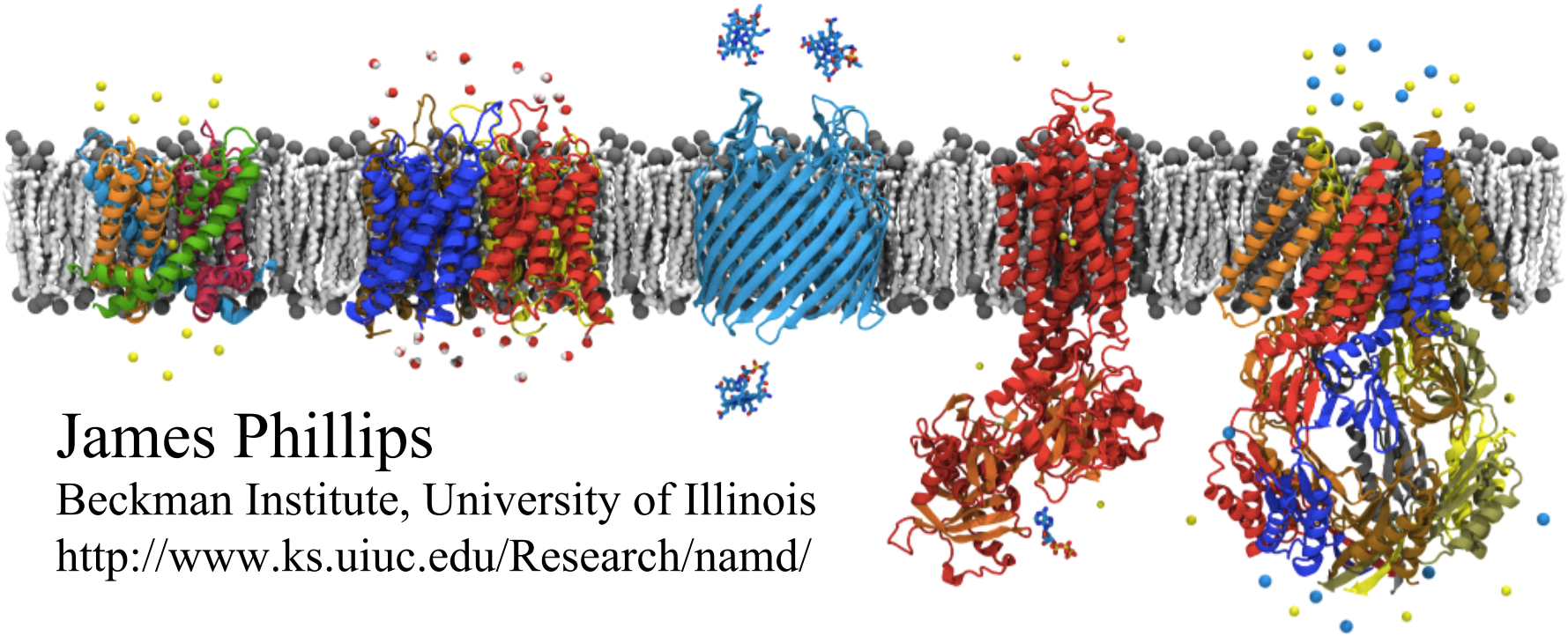
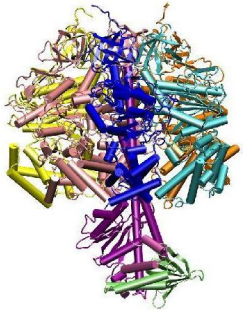
NAMD: GPU-Accelerated Petascale Molecular Dynamics Simulations on Titan and Blue Waters



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

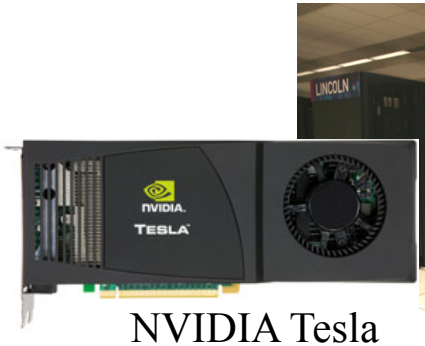
60,000 Users, 3000 Citations

Computational Biophysics Summer School

Blue Waters Target Application

Illinois Petascale Computing Facility

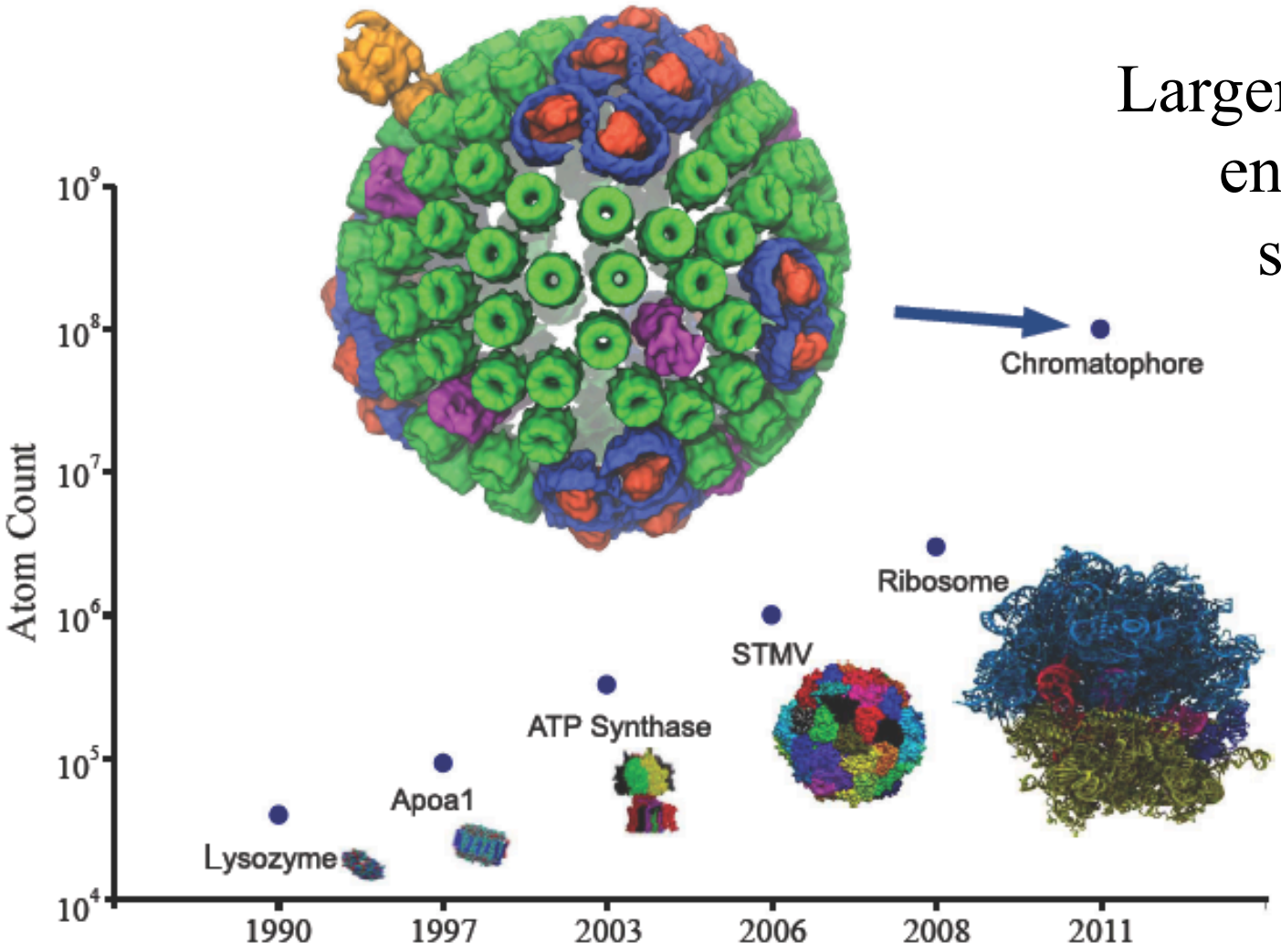
GPU Acceleration



NVIDIA Tesla

NCSA Lincoln

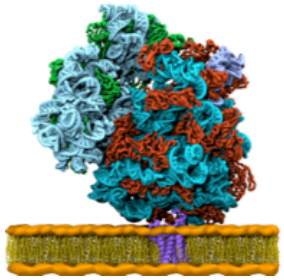
Larger machines enable larger simulations

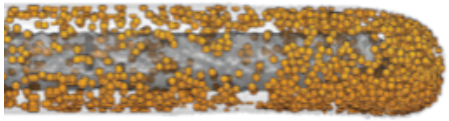


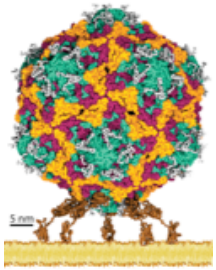
Beckman Institute, UIUC

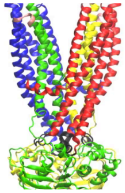

NIH BTRC for Macromolecular Modeling and Bioinformatics
1990-2017
Beckman Institute
University of Illinois at
Urbana-Champaign

Collaborative Driving Projects

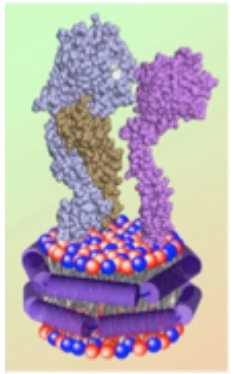


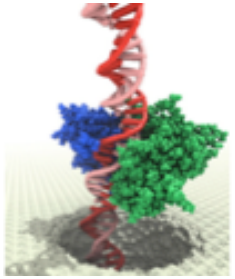


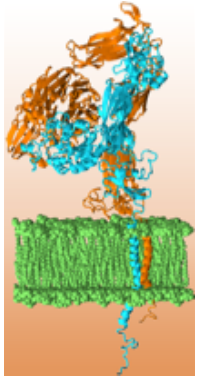




1. Ribosome	R. Beckmann (U. Munich) J. Frank (Columbia U.) T. Ha(UIUC) K. Fredrick (Ohio state U.) R. Gonzalez (Columbia U.)
2. Blood Coagulation Factors	J. Morrissey (UIUC) S. Sligar (UIUC) C. Rienstra (UIUC) G. Gilbert (Harvard)
3. Whole Cell Behavior	W. Baumeister (MPI Biochem.) J. Xiao (Johns Hopkins 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.)





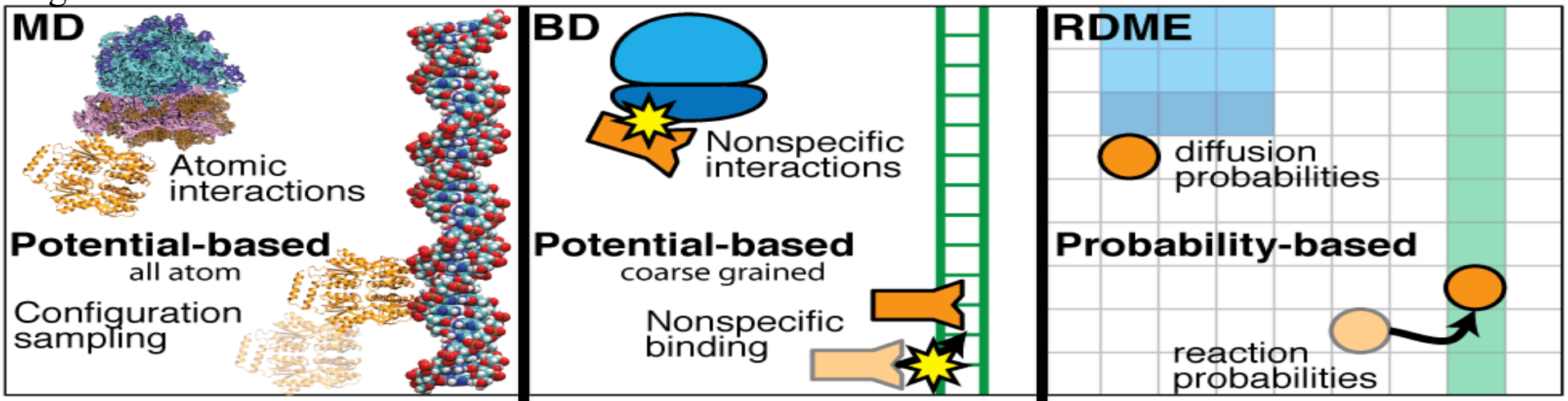


Physics of in vivo Molecular Systems

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

Parallel Programming Lab

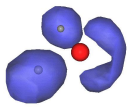
University of Illinois at Urbana-Champaign

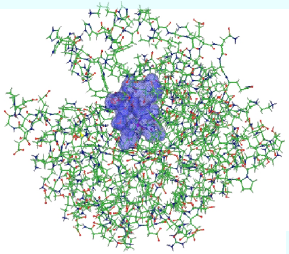
Siebel Center for Computer Science

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

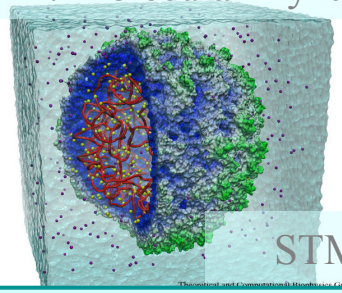
Develop abstractions in context of full-scale applications

Quantum Chemistry (QM/MM)

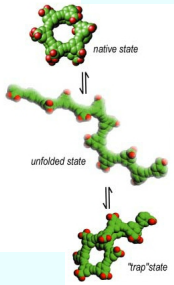




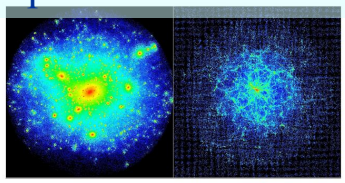
NAMD: Molecular Dynamics

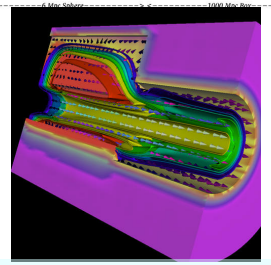


Protein Folding



Computational Cosmology



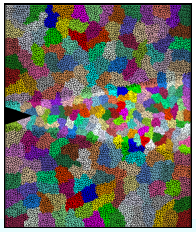


STM virus simulation

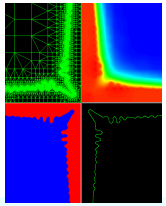
Parallel Objects,

Adaptive Runtime System

Libraries and Tools

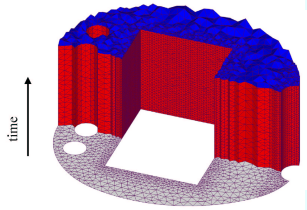


Crack Propagation



Rocket Simulation

Dendritic Growth



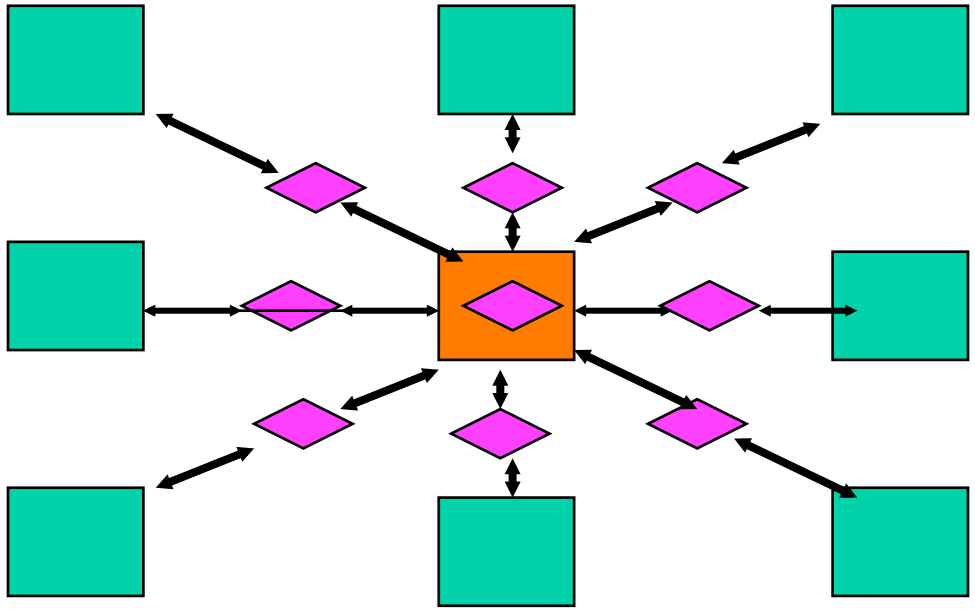
Space-time meshes

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

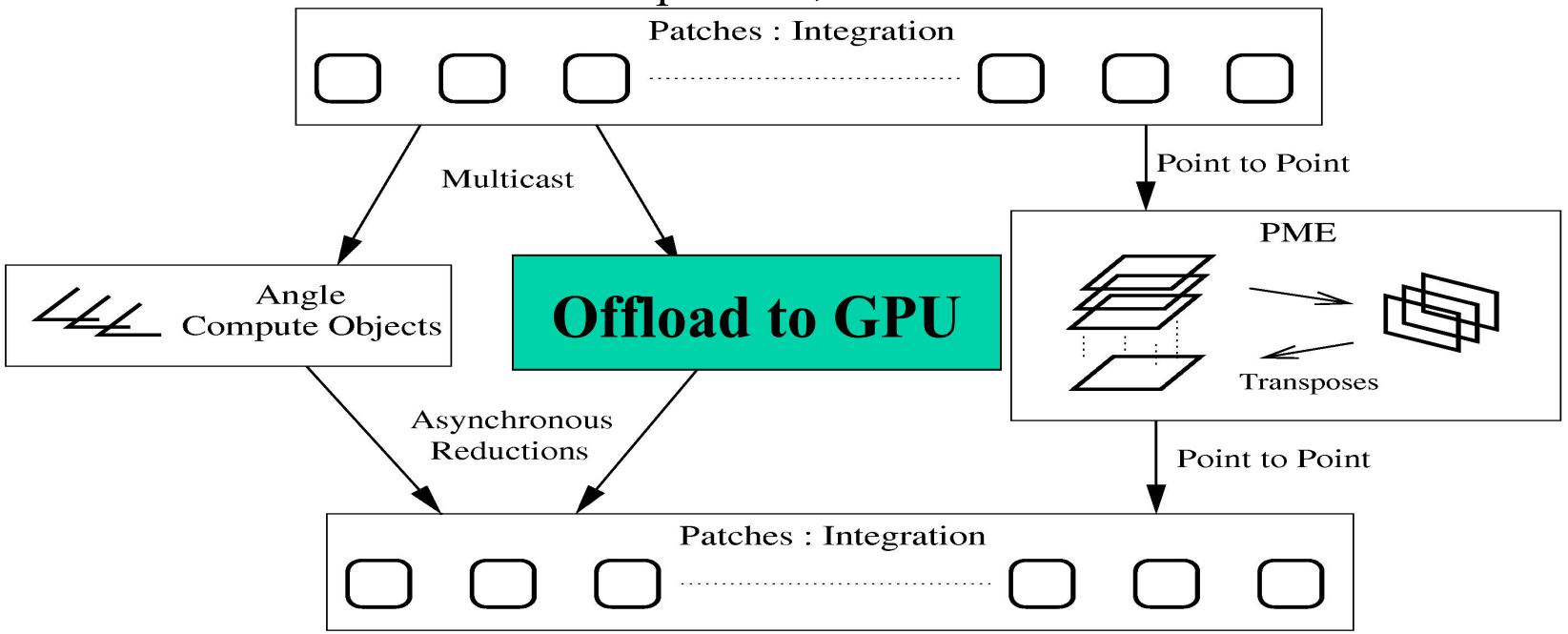
Kale *et al., J. Comp. Phys.* **151**:283-312, 1999.



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

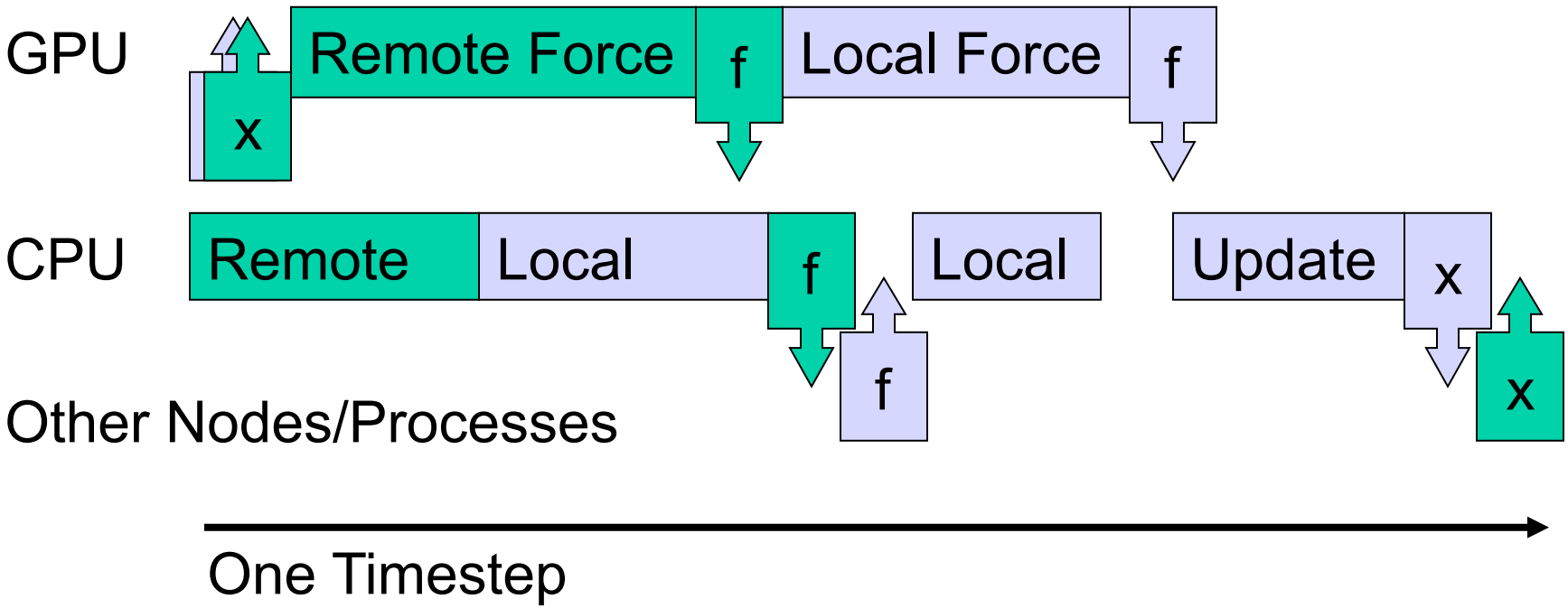
NAMD Overlapping Execution

Phillips *et al., SC2002.*



Objects are assigned to processors and queued as data arrives.

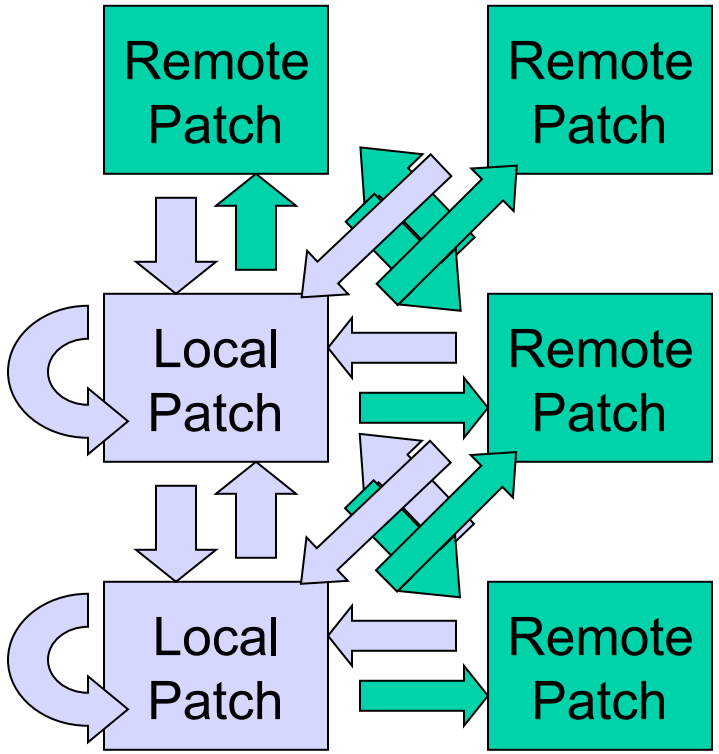
Overlapping GPU and CPU with Communication



Phillips *et al.*, SC2008

“Remote Forces”

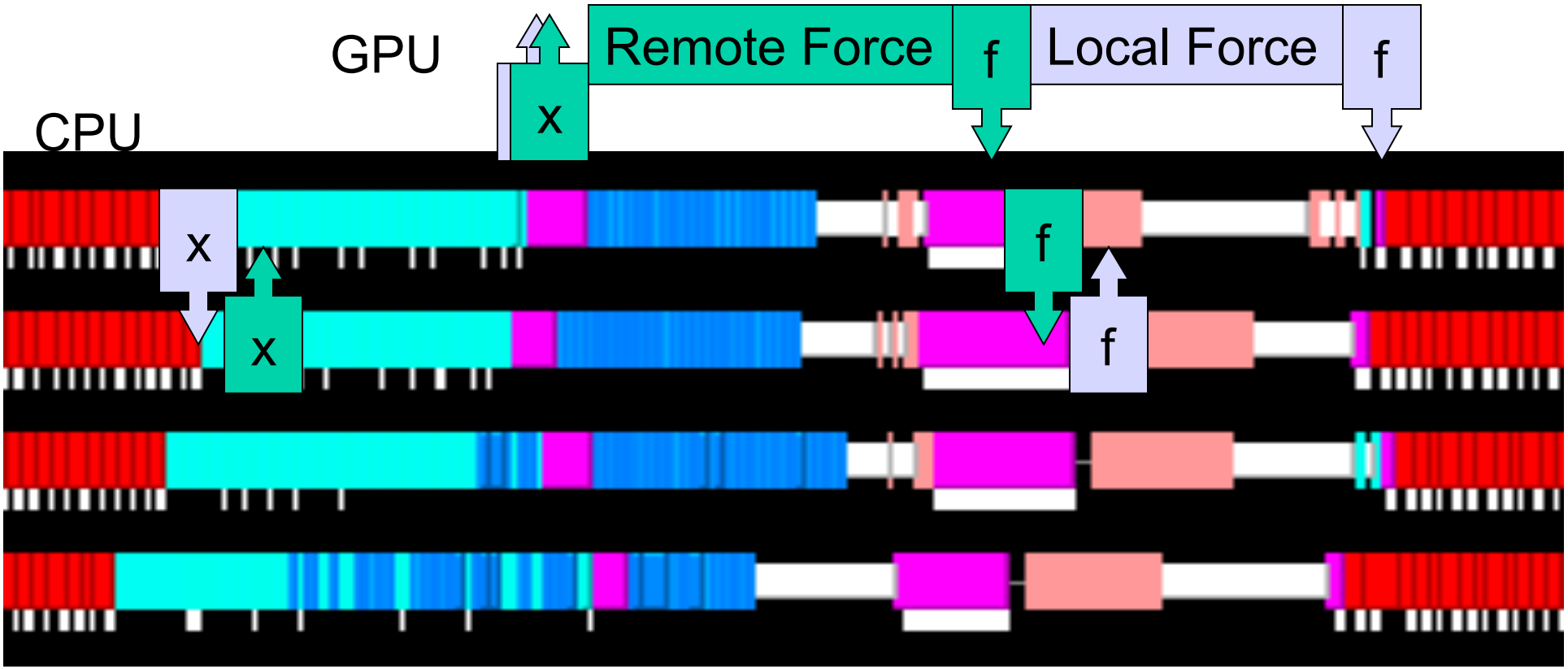
- Forces on atoms in a local patch are “local”
- Forces on atoms in a remote patch are “remote”
- Calculate remote forces first to overlap force communication with local force calculation
- Not enough work to overlap with position communication



Work done by **one** processor

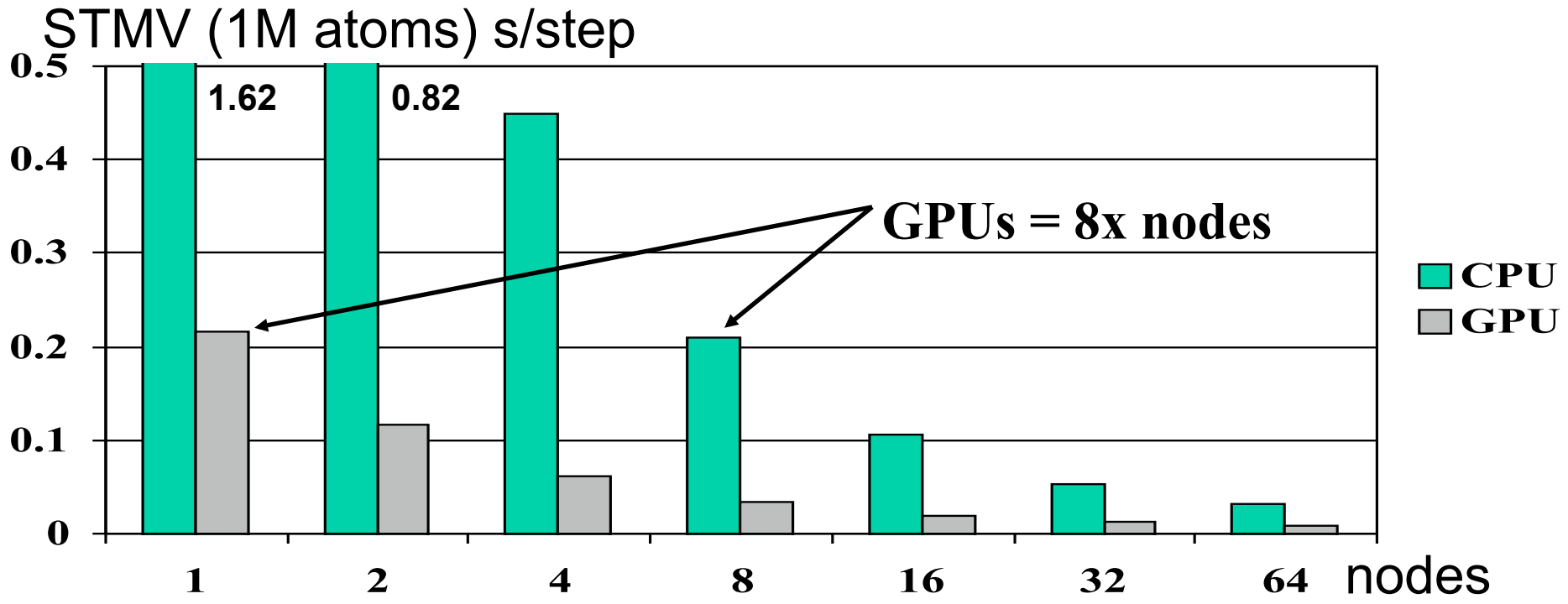
Actual Timelines from NAMD

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



NAMD 2.9 on Keeneland ID

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



Know Your (Cray) Supercomputers

Titan

- Funded by DOE
- Allocated by INCITE, etc.
- NCCS (Oak Ridge)
- 18,688 XK7 compute nodes
- In full production

Blue Waters

- Funded by NSF
- Allocated by PRAC
- NCSA (U. Illinois)
- 22,000 XK6 compute nodes + 3,000 XK7 compute nodes
- Closed for remodeling and expansion (+ 1152 XK7)

Trends Affecting Performance

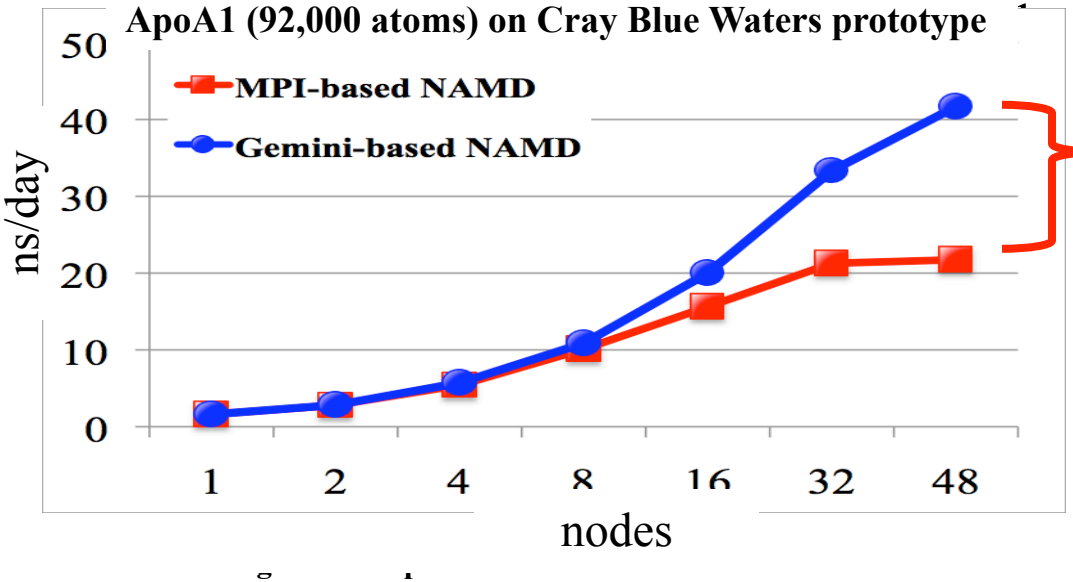
- GPU performance increasing
  - Performance limit will be code on CPU
  - Most highly tuned CPU code moved to GPU
  - Remaining CPU code is also less efficient
  - Therefore CPU must run serial code well
- CPU serial performance static
- CPU core counts increasing

Suggested Strategy

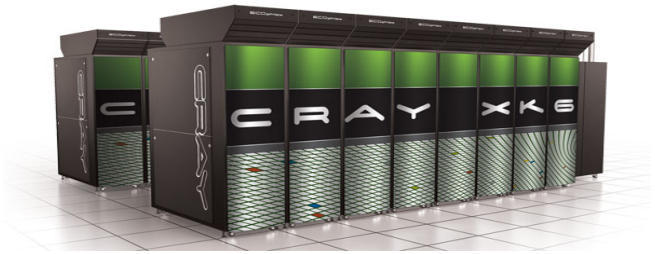
- Focus on CPU-side code
  - Port to GPU or optimize/paralellize on CPU
  - Stream results off GPU to increase overlap
  - Use CPUs with best single-thread performance
- Focus on communication
  - Reduce communication overhead on CPU
  - Deal with multithreaded MPI issues
  - General parallel scalablity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

Cray Gemini Optimization

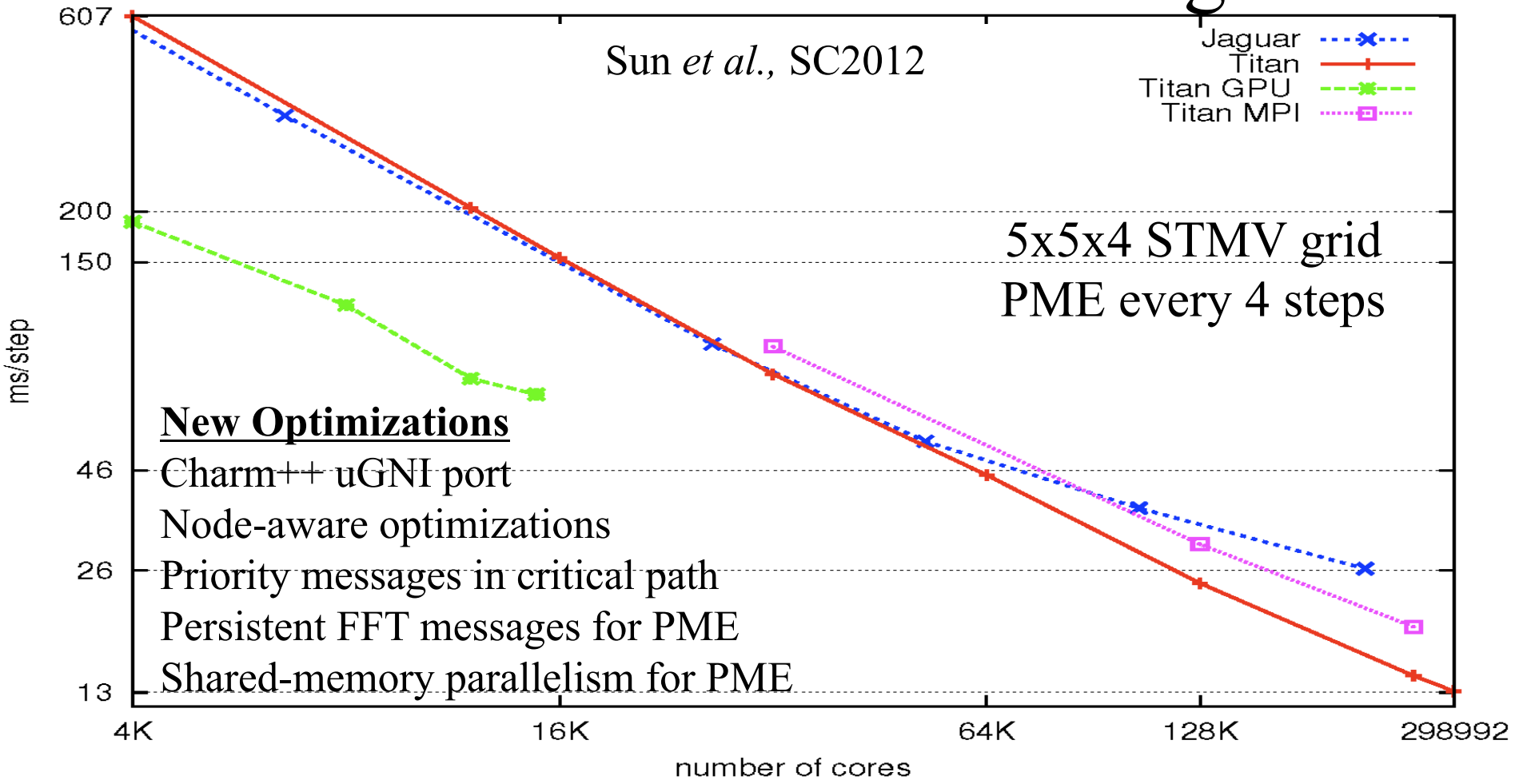
- The new Cray machine has a better network (called Gemini)
- MPI-based NAMD scaled poorly
- BTRC implemented direct port of Charm++ to Cray
  - *uGNI* is the lowest level interface for the Cray Gemini network



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



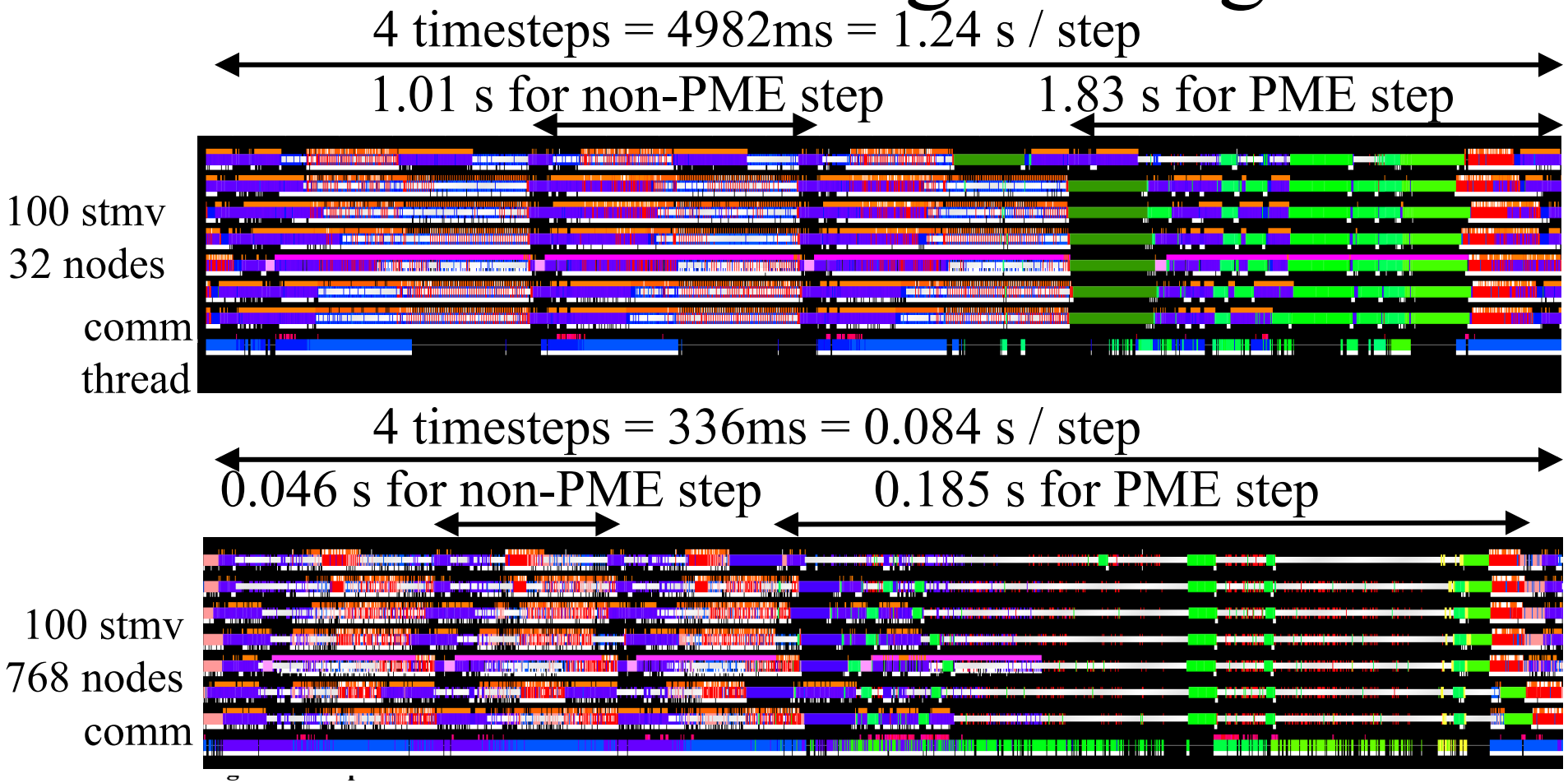
100M Atoms on Titan vs Jaguar



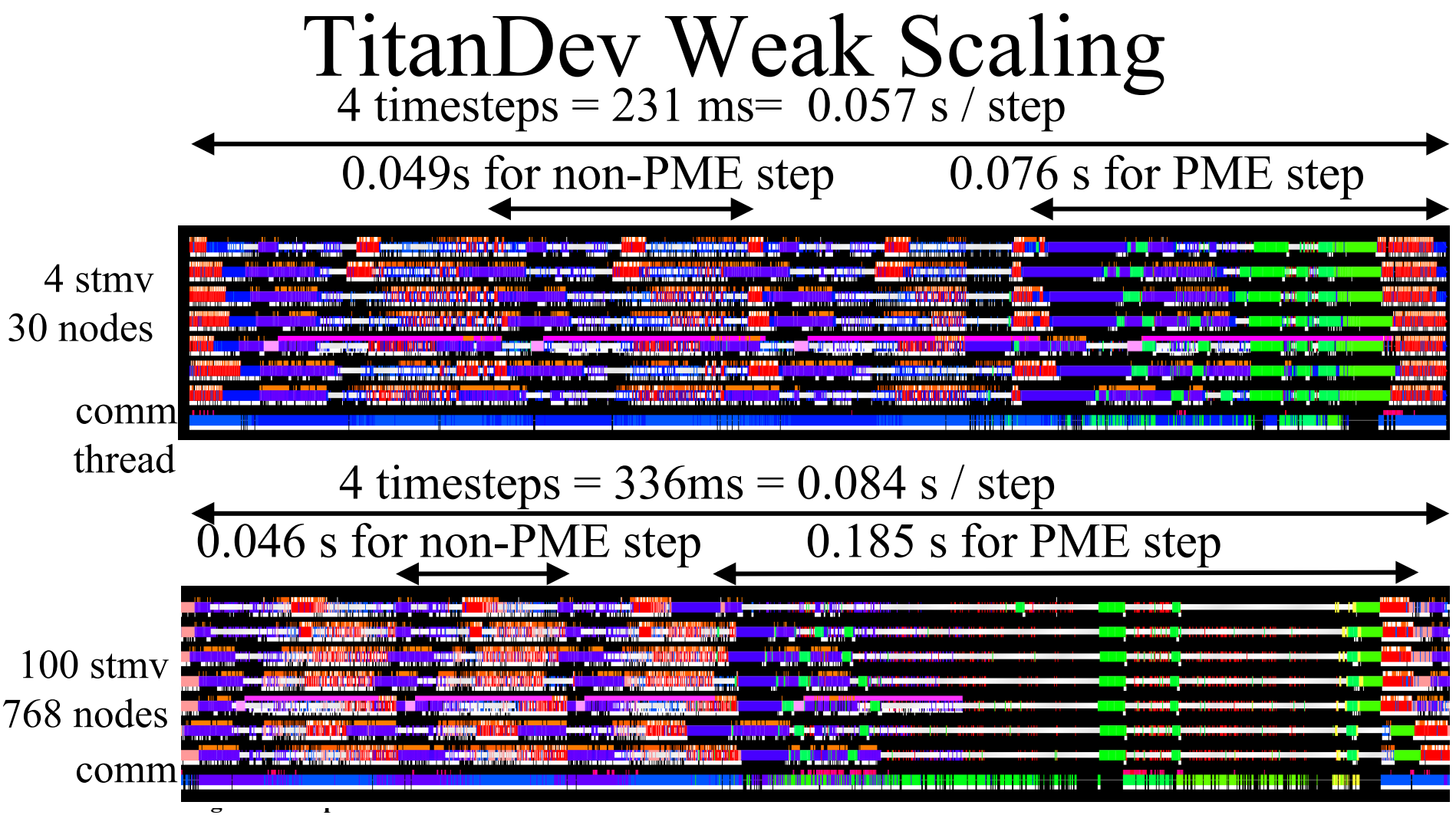
Streaming GPU Results to CPU

- 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

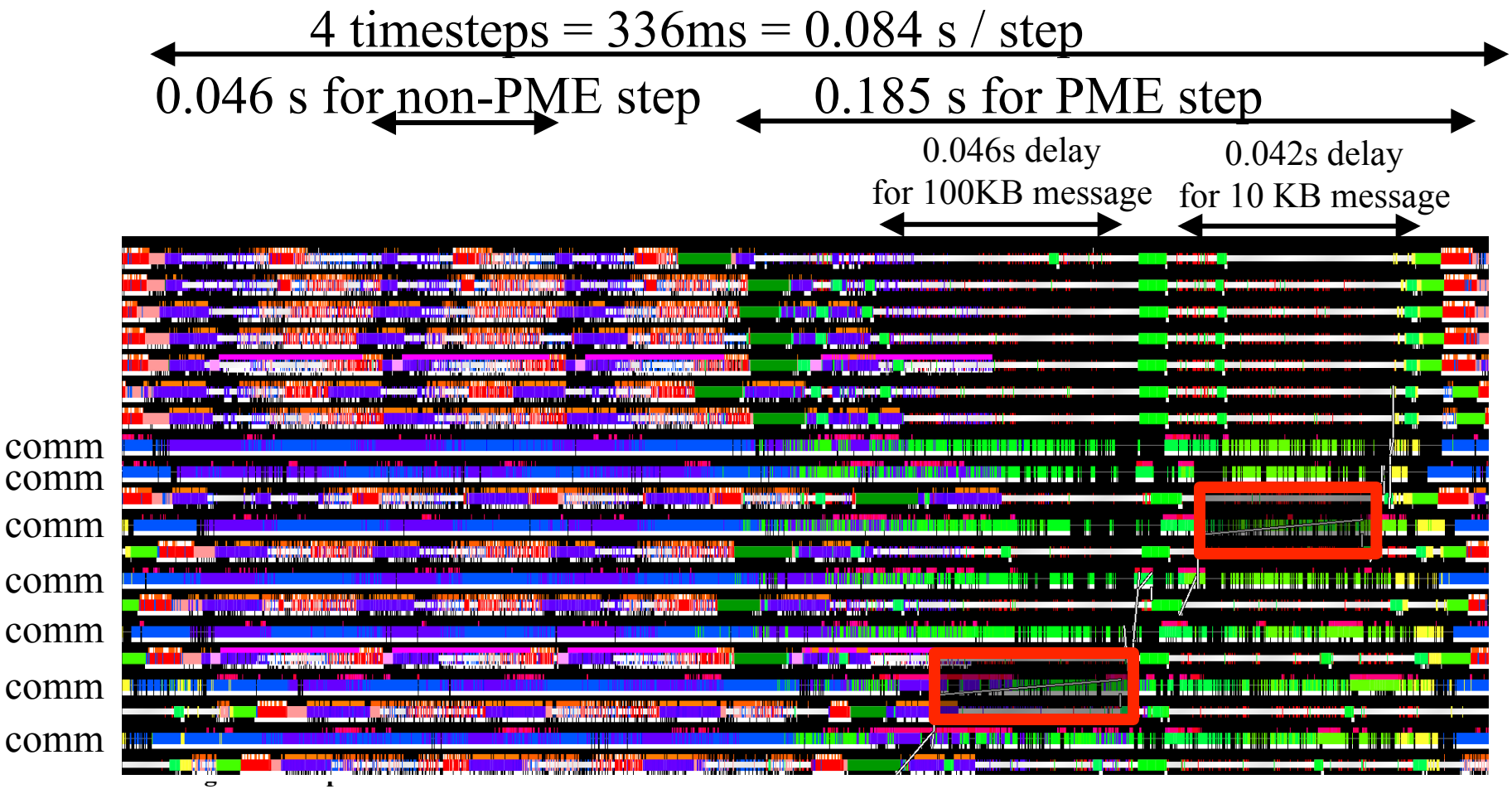
TitanDev Strong Scaling



TitanDev Weak Scaling



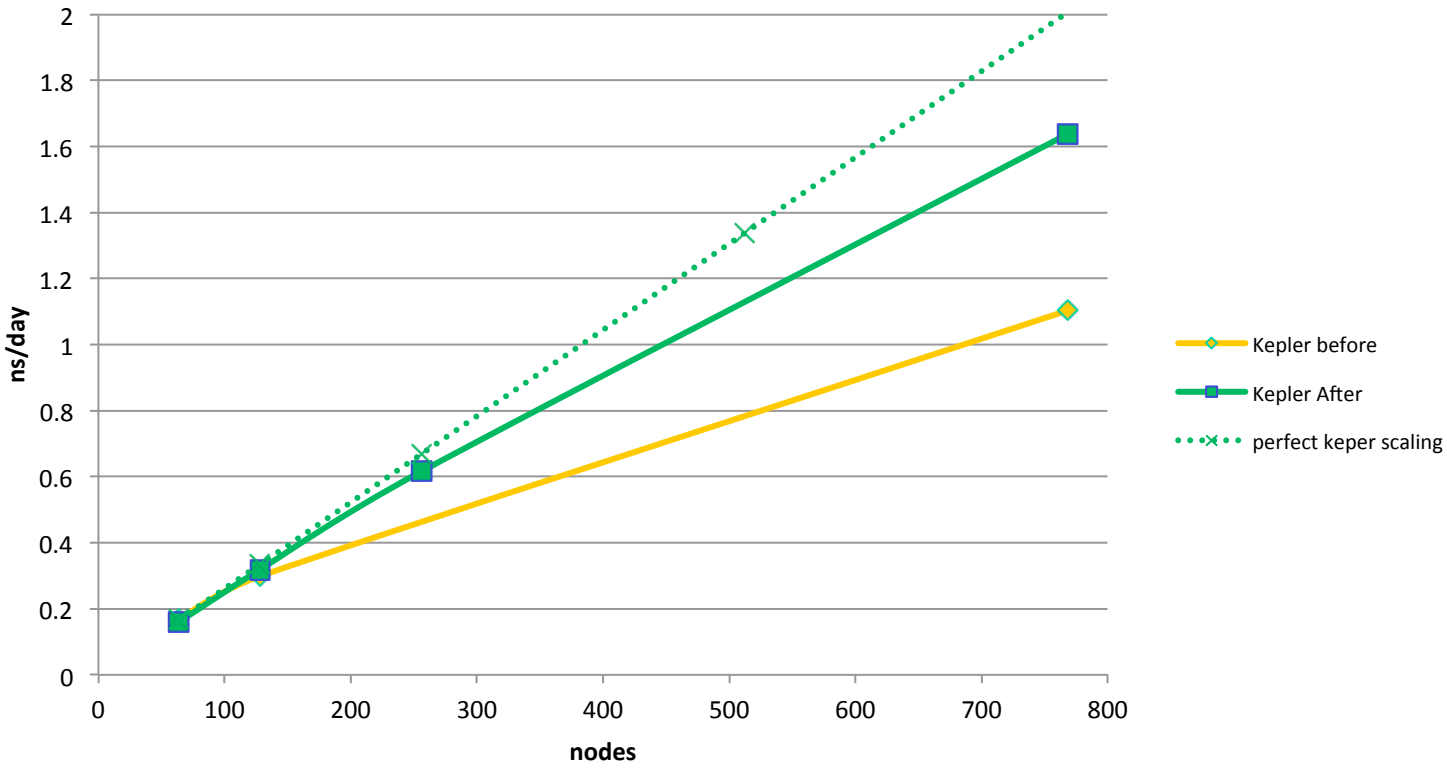
PME delays – tracing data needed for one ungrid calculation

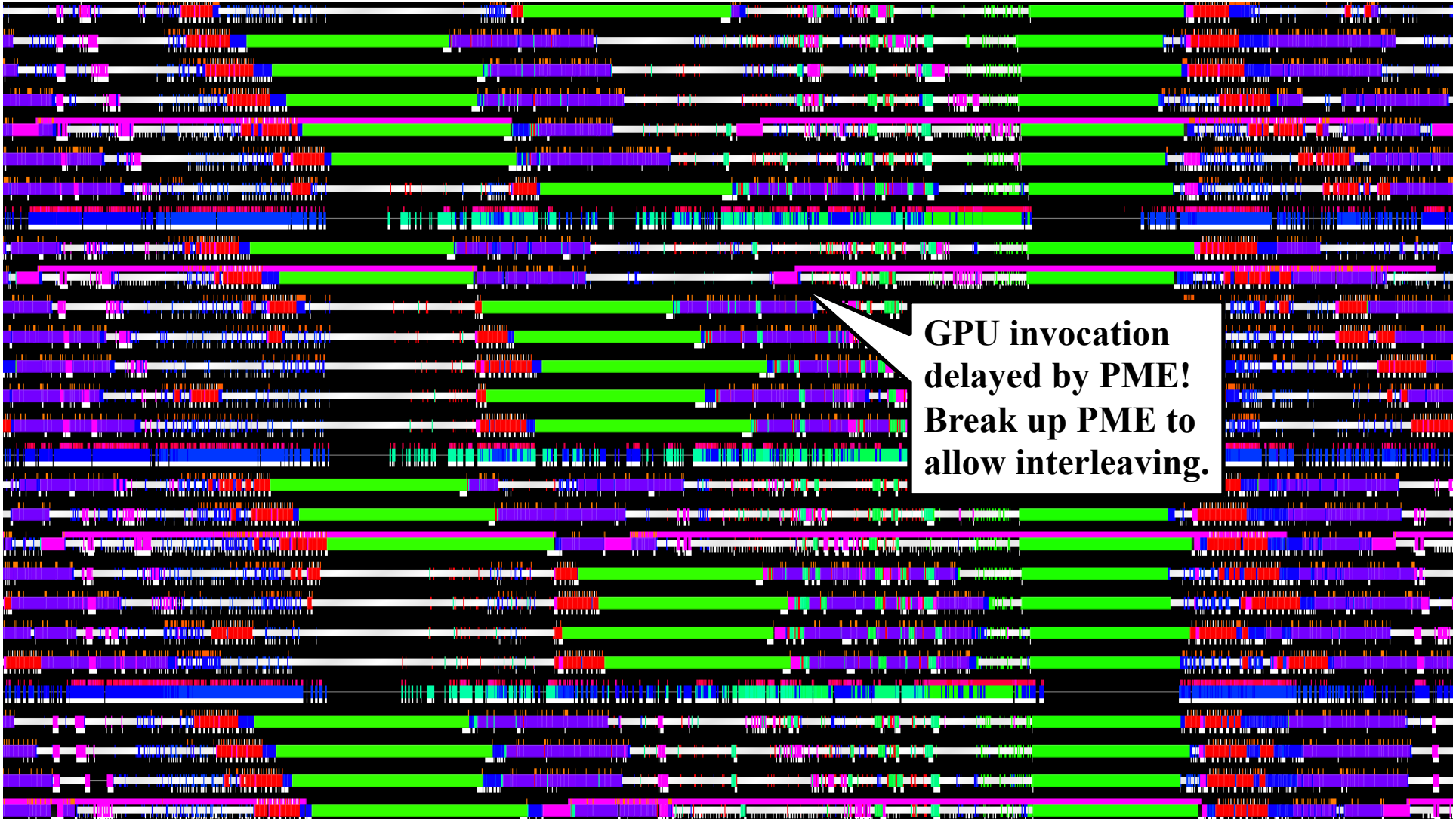


Strategy to improve scalability

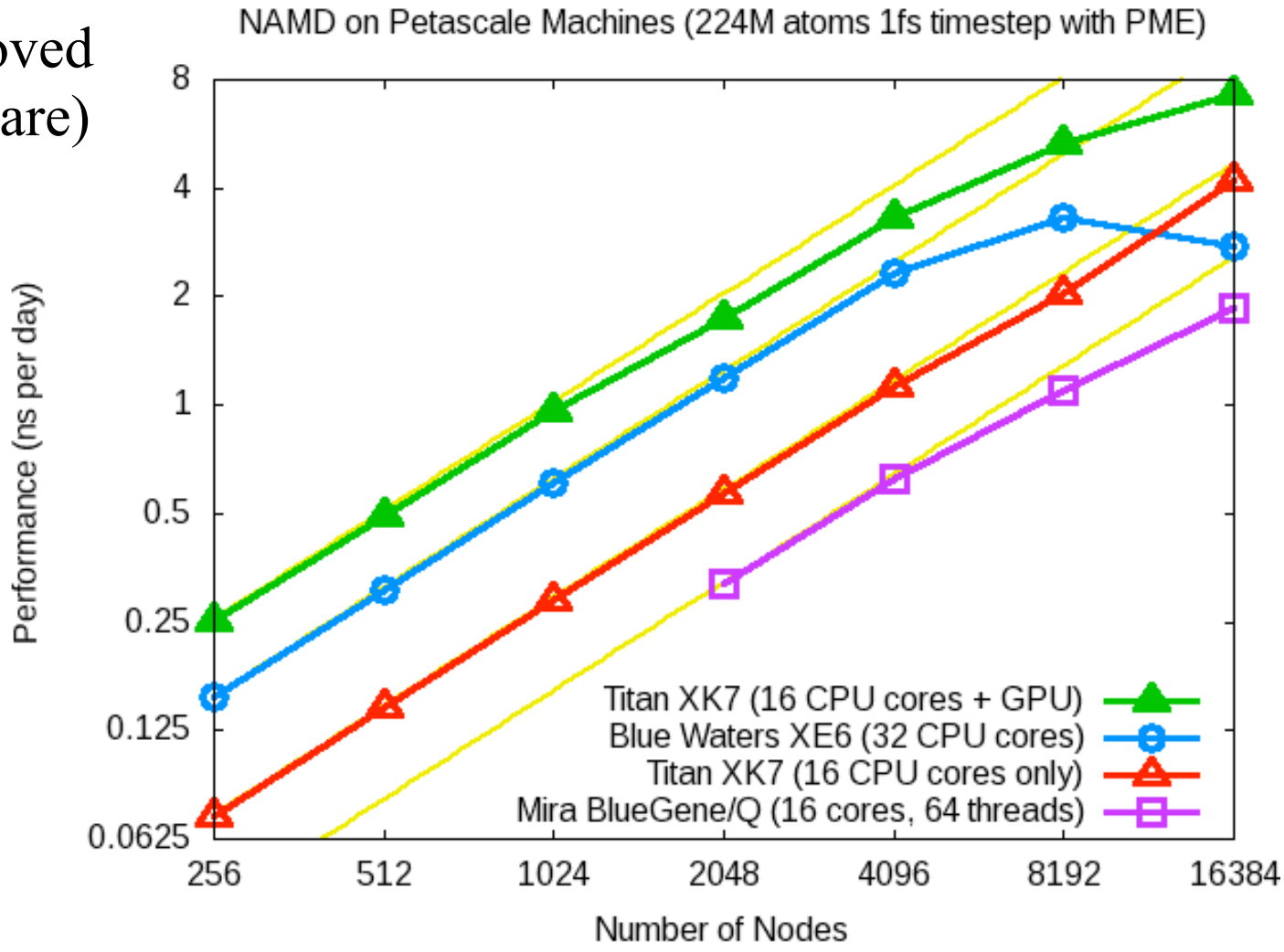
- Coarsen PME grid with higher-order interpolation
  - Reduces communication (factor of 8)
  - Does not increase short-range work or communication
- Start GPU work sooner
  - Currently waiting for all coordinate receives
  - Break up (now longer PME) to avoid delays
- Fix issues with communication
  - Improved mapping of patches to nodes
- Push PME work to the GPU
  - Charge gridding overlaps coordinate receive

Effect of Coarsening PME Grid

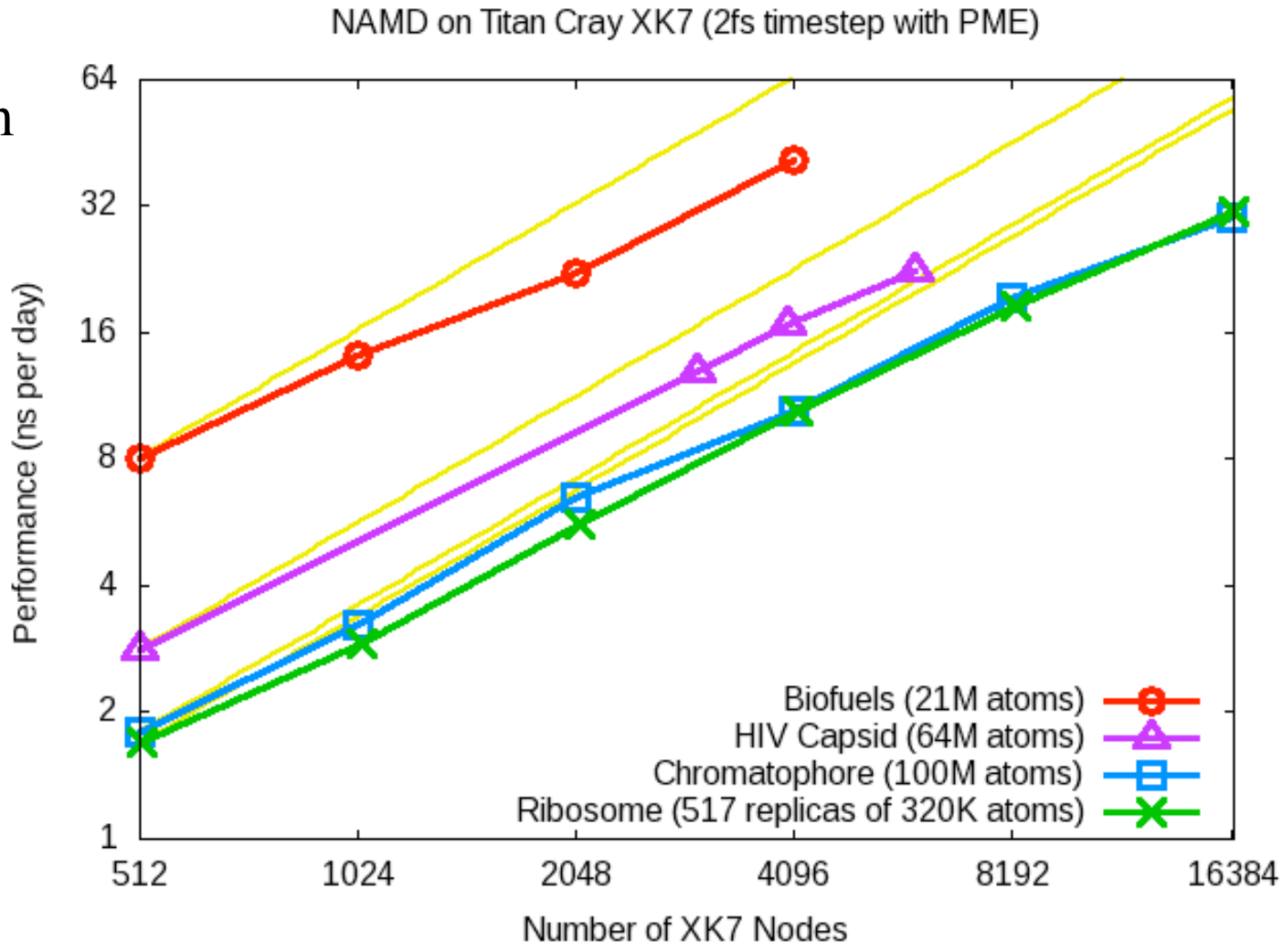



GPU invocation
delayed by PME!
Break up PME to
allow interleaving.

June 2013: Improved (but not torus-aware) patch placement.



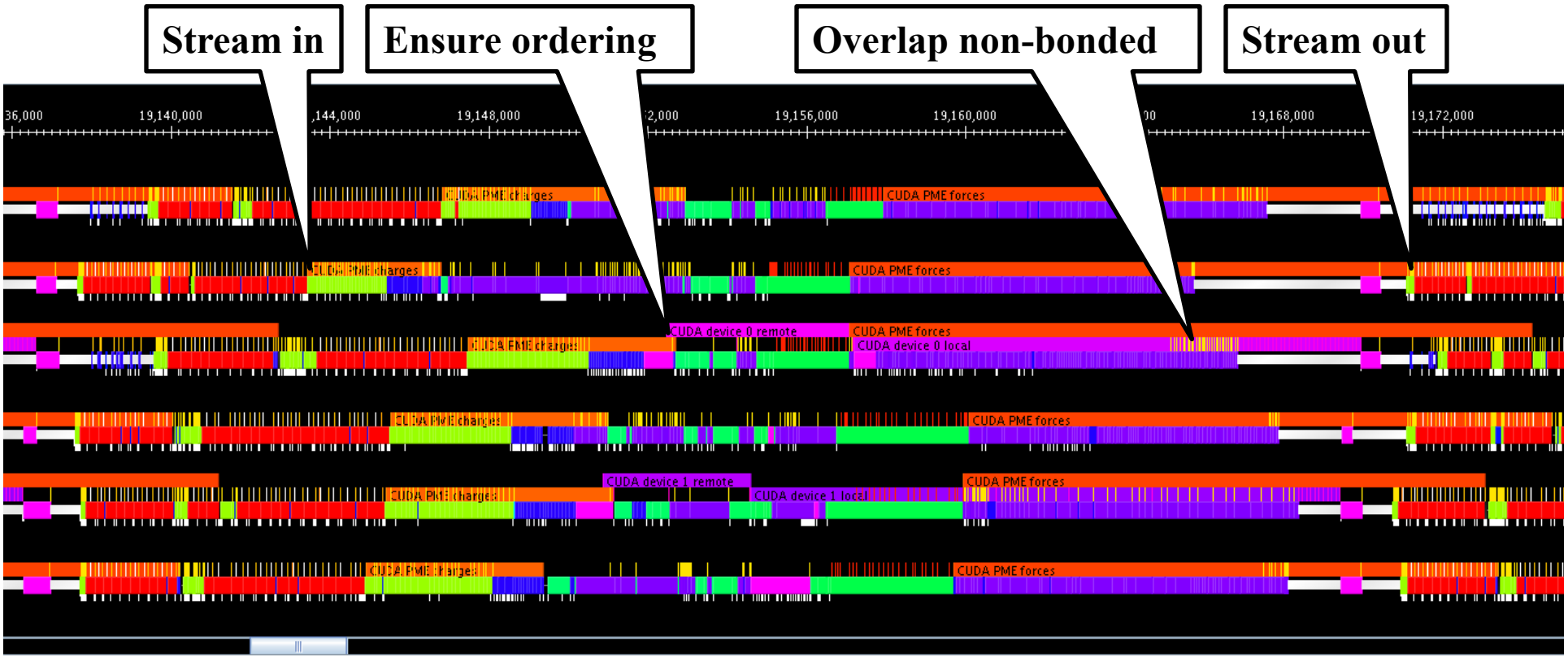
Aug 2013: **after** torus-aware patch placement.

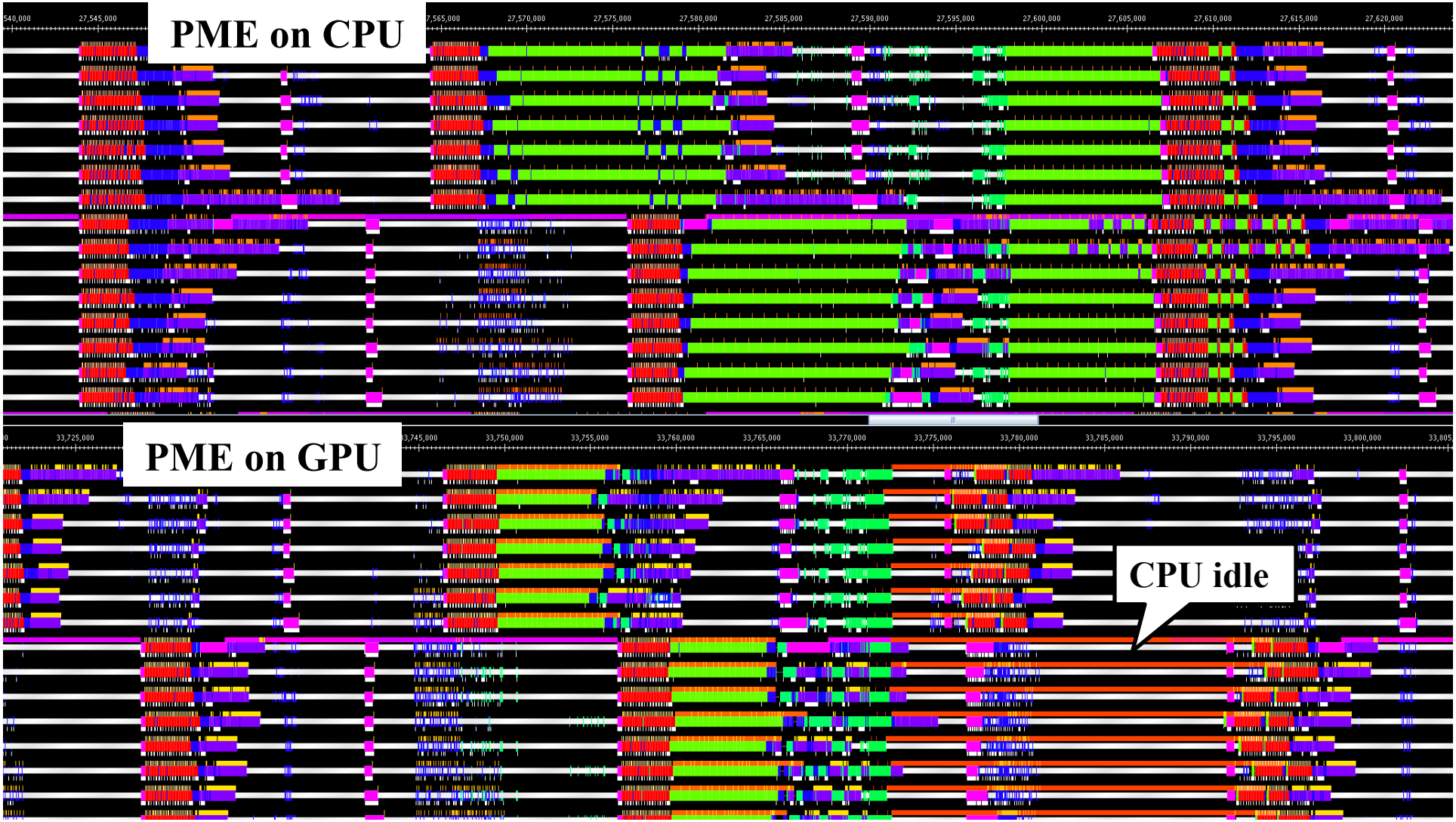


NAMD PME CUDA Kernel

- 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
  - 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



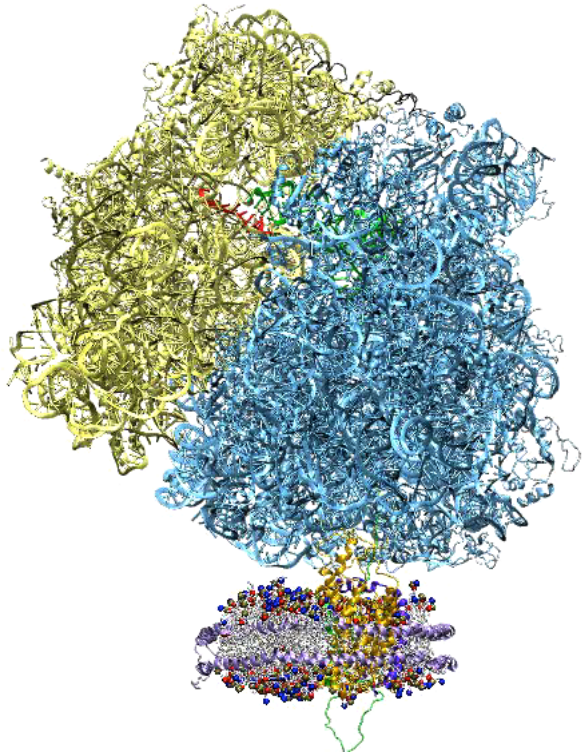

PME on CPU
PME on GPU
CPU idle

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

NAMD 2.9: What is accelerated?

Accelerated

- Short-range non-bonded
  - Cutoff or with PME
  - w/ or w/o energy calculation
- Implicit solvent
- NVIDIA GPUs only

Not Accelerated

- Bonded terms
- PME reciprocal sum
- Integration
- Rigid bonds
- Grid forces
- Collective variables
- Etc.

NAMD 2.9: What is disabled?

Disabled

- Alchemical (FEP and TI)
- Locally enhanced sampling
- Tabulated energies
- Drude (nonbonded Thole)
- Go forces
- Pairwise interaction
- Pressure profile

Not Disabled

- Memory optimized builds
- Conformational free energy
- Collective variables
- Grid forces
- Steering forces
- Almost everything else

NAMD 2.9: What is different?

- Forces
  - Slightly less accurate than CPU
    - Different interpolation scheme, single precision
    - Also affects pressure calculation
- Energies
  - Don't match forces as closely as on CPU
  - Constant for interactions less than 1 Angstrom
    - Seems to be causing minimizer issues
    - Velocity-quenching minimizer should work better

NAMD 2.9: Performance

What to expect

- 1 GPU = ~24 CPU cores
  - Depending on CPU and GPU
- Scaling to 10K atoms/GPU
  - Assuming fast network
- Must use smp/multicore
  - Many cores share each GPU
  - Use multicore for single node
  - At most one process per GPU

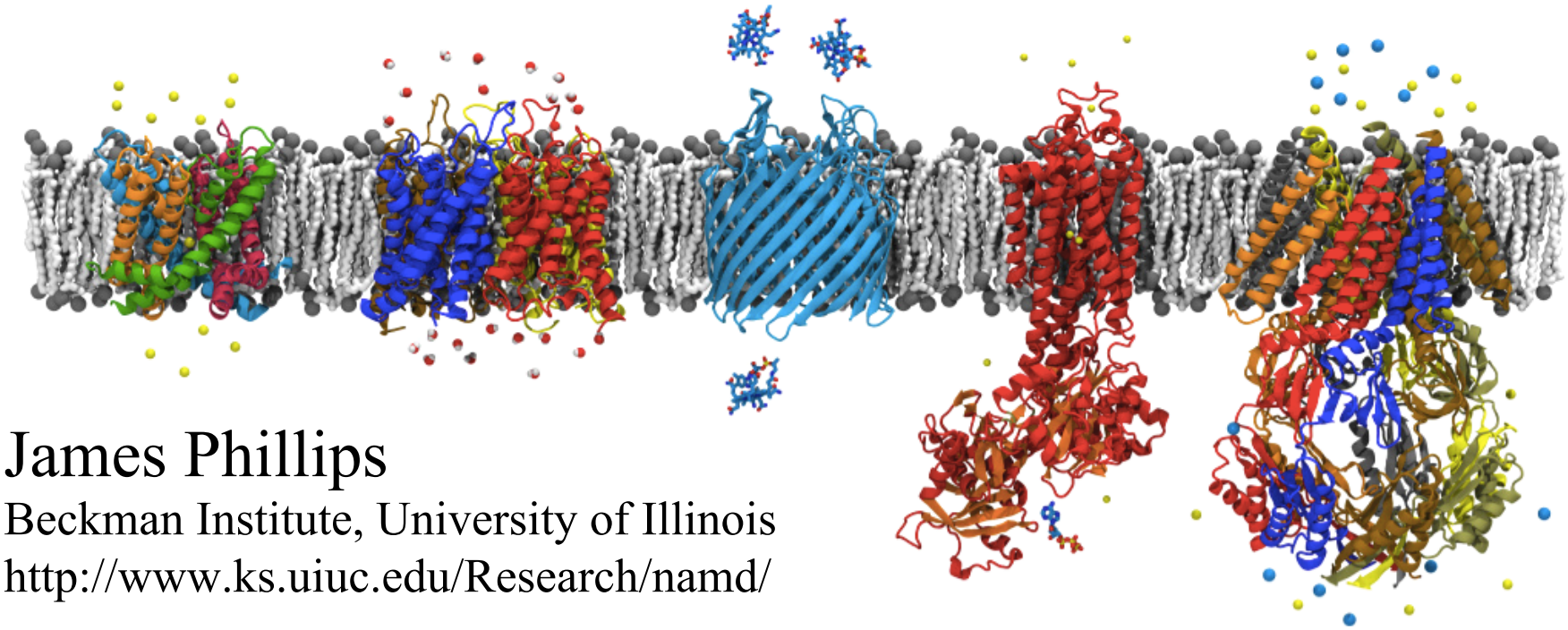
Why it may be worse

- Weak GPU (e.g., laptop)
- Too few CPU cores used
- Coarse-grained simulation
- Too few atoms per GPU
- Limited by network
- Limited by MPI (use ibverbs)
- Limited by special features

NAMD 2.10: Kepler optimization

- Current NAMD kernel design is from 2007
- Kepler is current NVIDIA GPU architecture
  - Used on Titan and Blue Waters
  - Adds new capabilities relevant for MD codes
- Designing and optimizing new kernel for Kepler
  - New version should be faster and scale better
  - Force/energy interpolation closer to CPU version
    - Minimizer should work as well as on CPU
  - Backport to Fermi, possibly earlier if possible

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/