

# Performance and GPU Acceleration in NAMD

**Julio Maia**  
**Research Programmer**

**NIH Center for Macromolecular Modeling and Bioinformatics**  
**University of Illinois at Urbana-Champaign**

**jmaia@ks.uiuc.edu**  
**[www.ks.uiuc.edu/~jmaia](http://www.ks.uiuc.edu/~jmaia)**



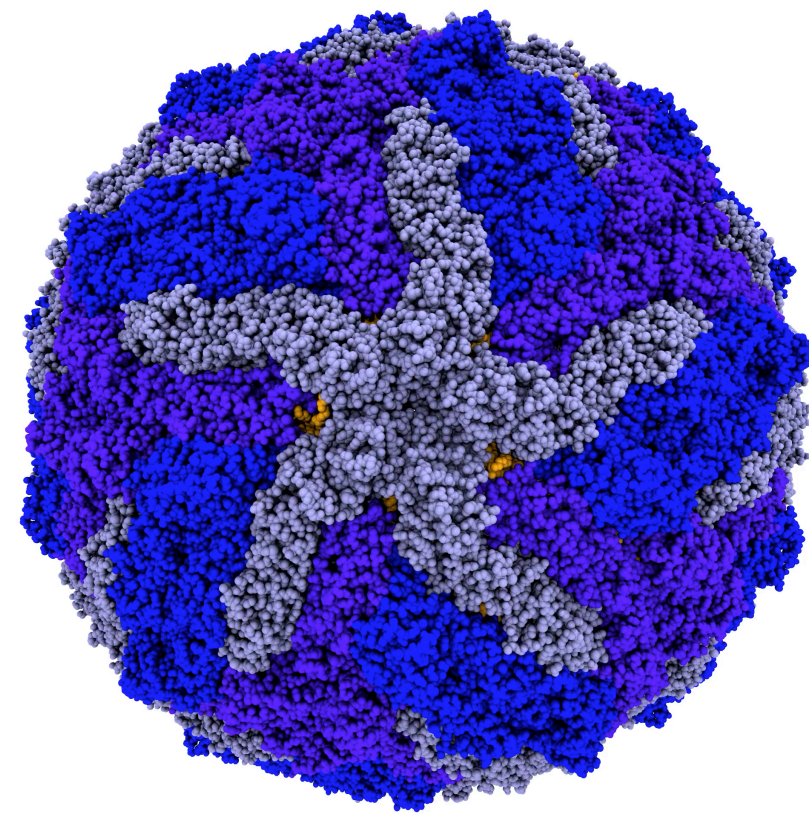
# NAMD as a Molecular Dynamics Engine

- Simulates the movement of biological systems according to Newton's Equations of motion

$$F_{\alpha} = - \frac{\partial}{\partial \vec{r}_{\alpha}} U_{total}(r_1, r_2, \dots, r_n), \alpha = 1, 2 \dots N$$

$$U_{total} = U_{bond} + U_{angle} + U_{dihedral} + U_{vdW} + U_{coulomb}$$

- Verlet integration method in order to propagate the system in time:



Zika Virus

$$v_{n+1/2} = v_n + M^{-1} F_n \cdot \frac{\Delta t}{2}$$

$$r_{n+1} = r_n + v_{n+1/2} \Delta t$$

$$F_{n+1} = F(r_{n+1})$$

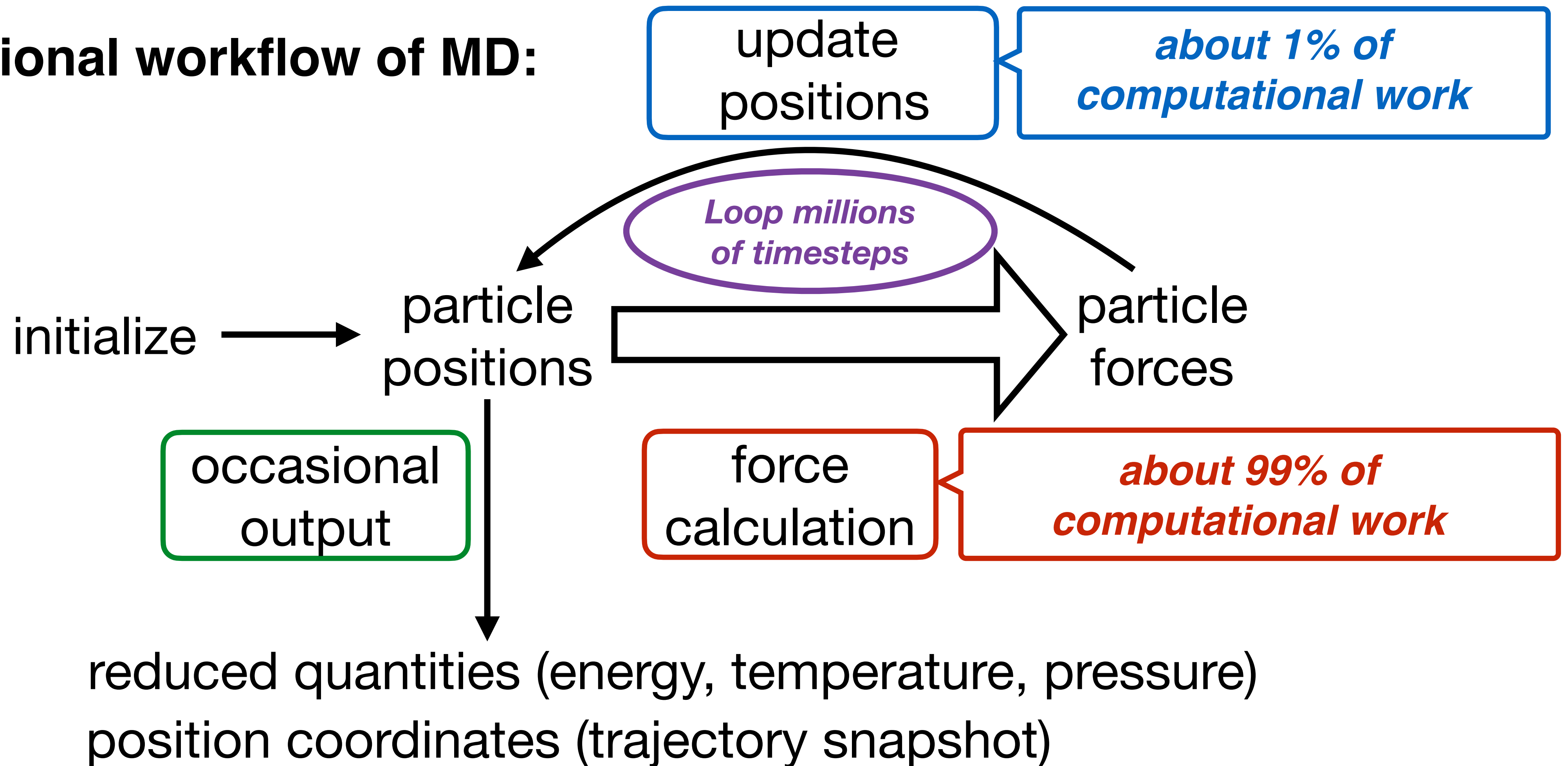
$$v_{n+1} = v_{n+1/2} + M^{-1} F_{n+1} \cdot \frac{\Delta t}{2}$$

$\Delta t$  = 1 or 2 femtoseconds usually



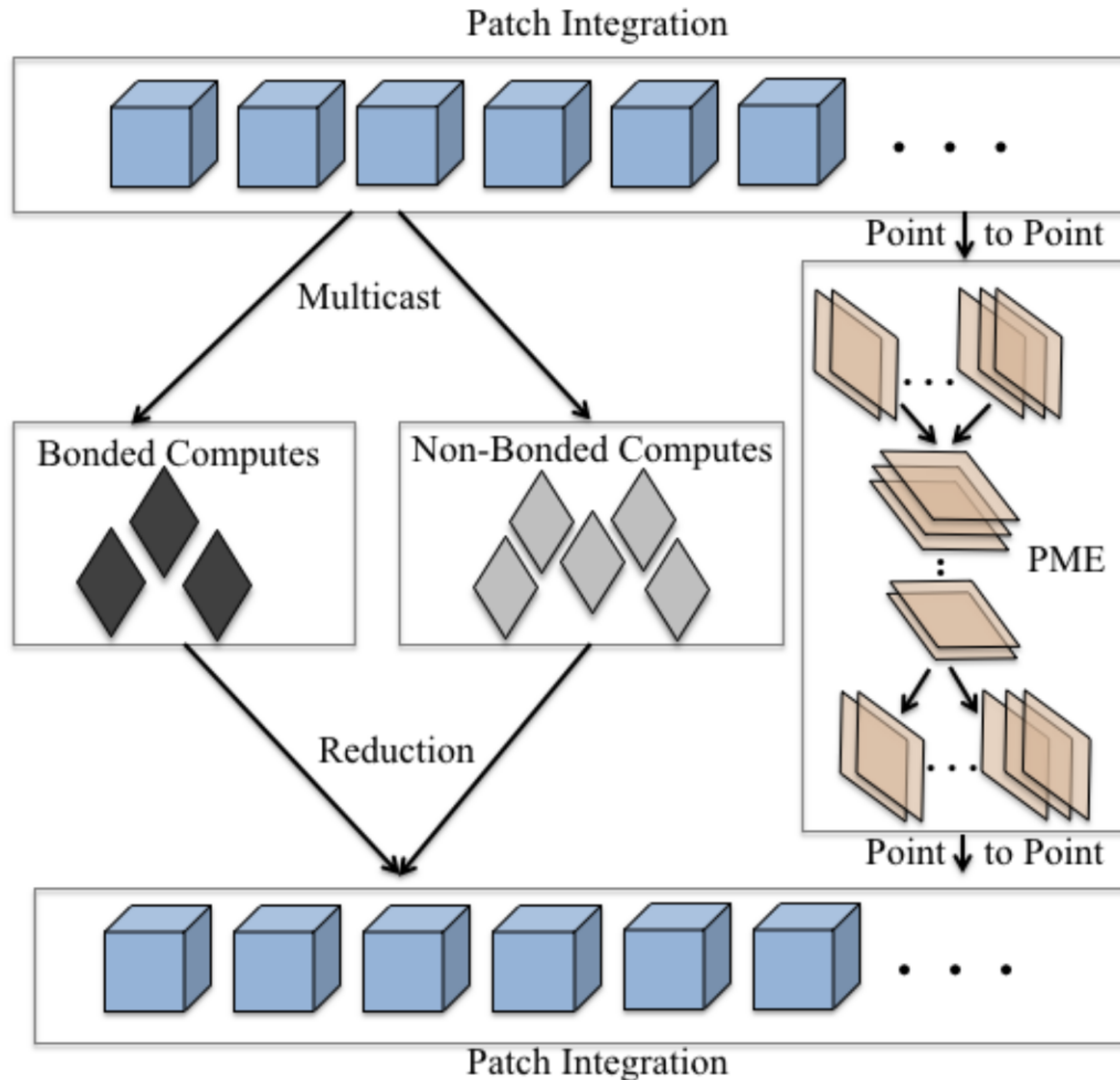
# Parallelism for MD Simulation Limited to Each Timestep

## Computational workflow of MD:



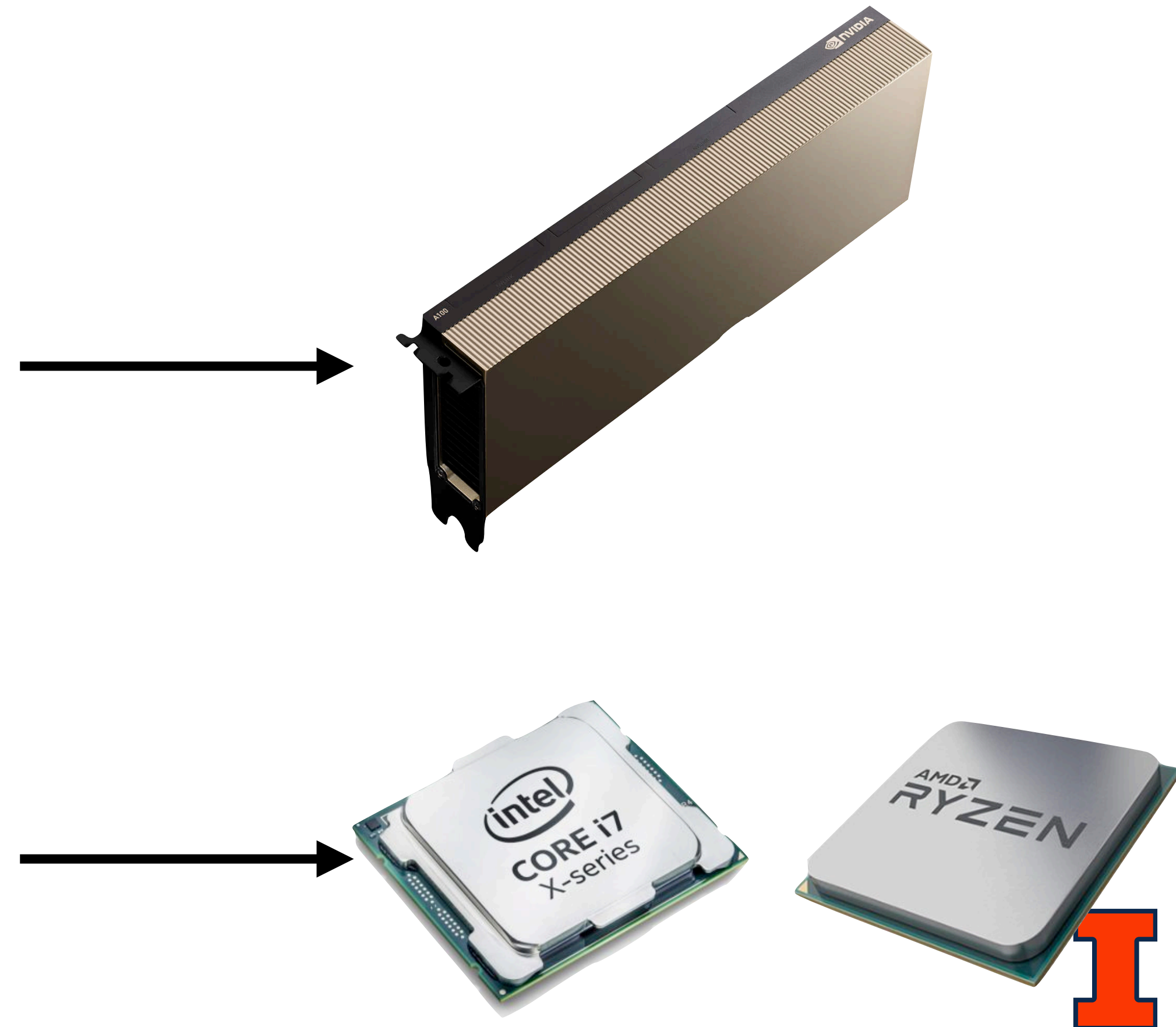
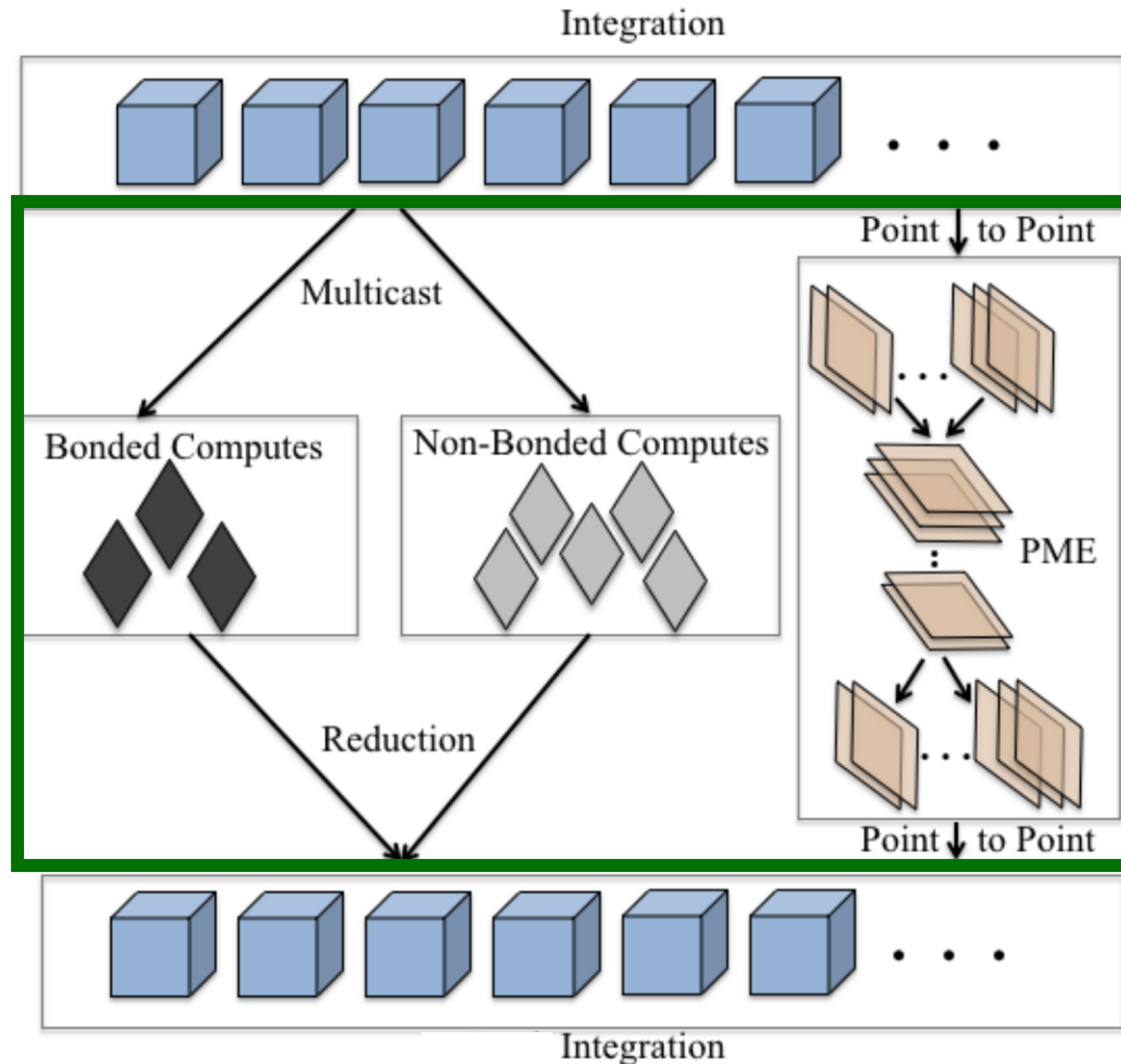


# NAMD 2.14 Decomposes Force Terms in Fine-Grained Objects for Scalability



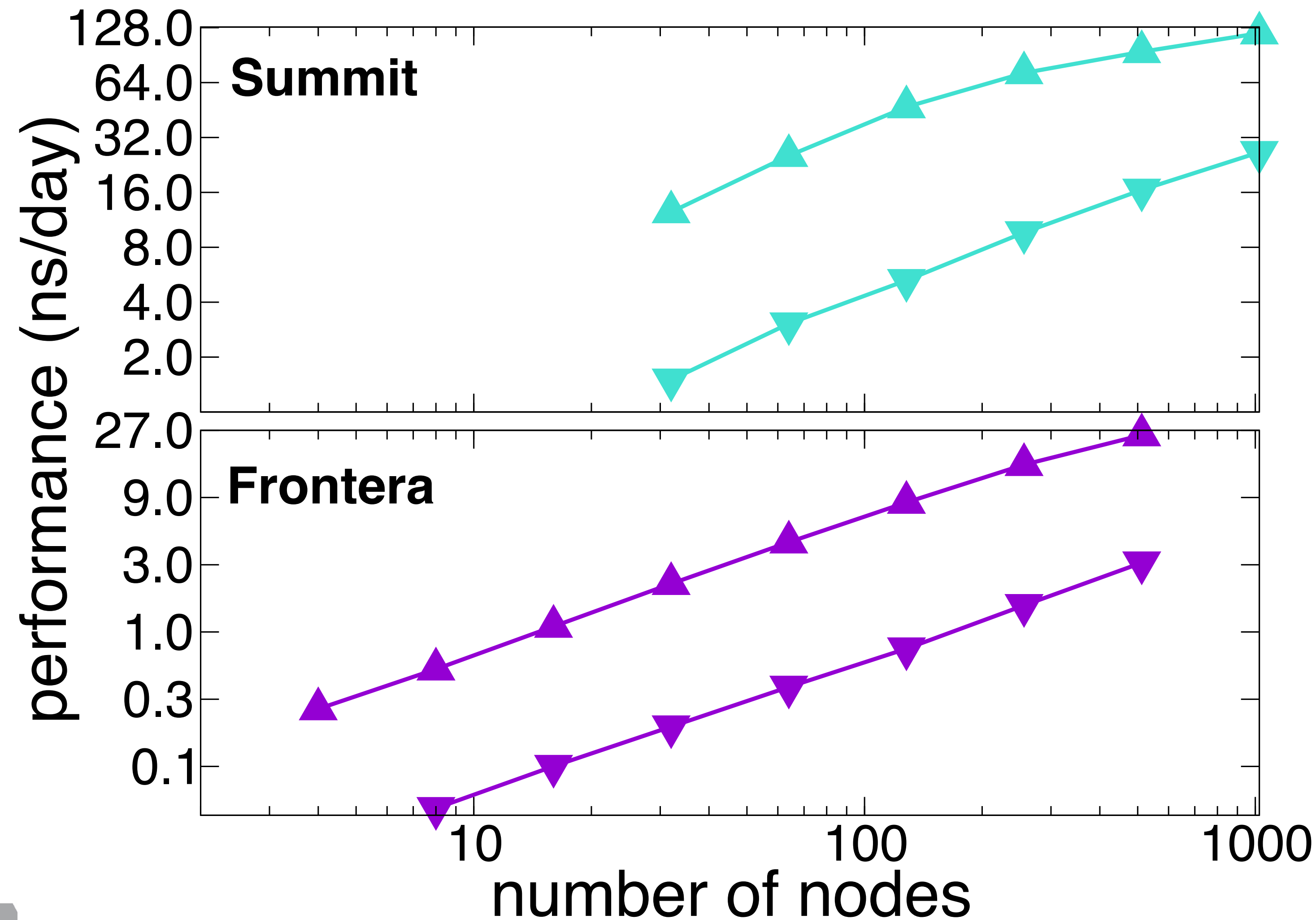


# NAMD 2.14 Decomposes Force Terms in Fine-Grained Objects for Scalability

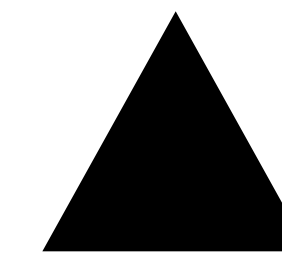
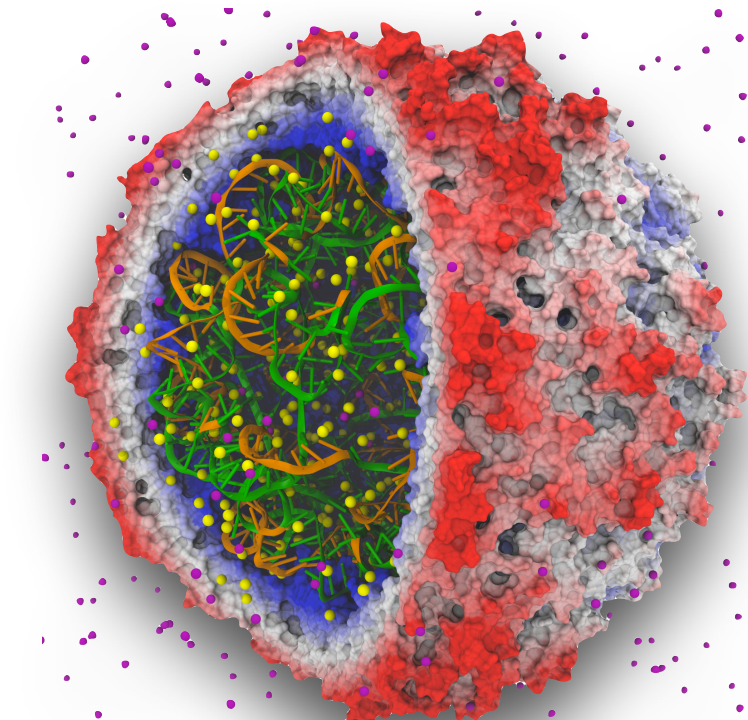




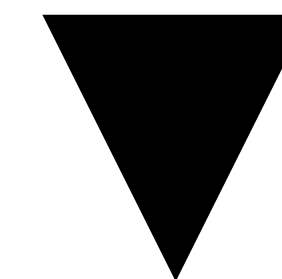
# NAMD 2.14 Excels at Scalable Parallelism on CPUs and GPUs



Replications of the  
Satellite Tobacco Mosaic Virus (STMV)



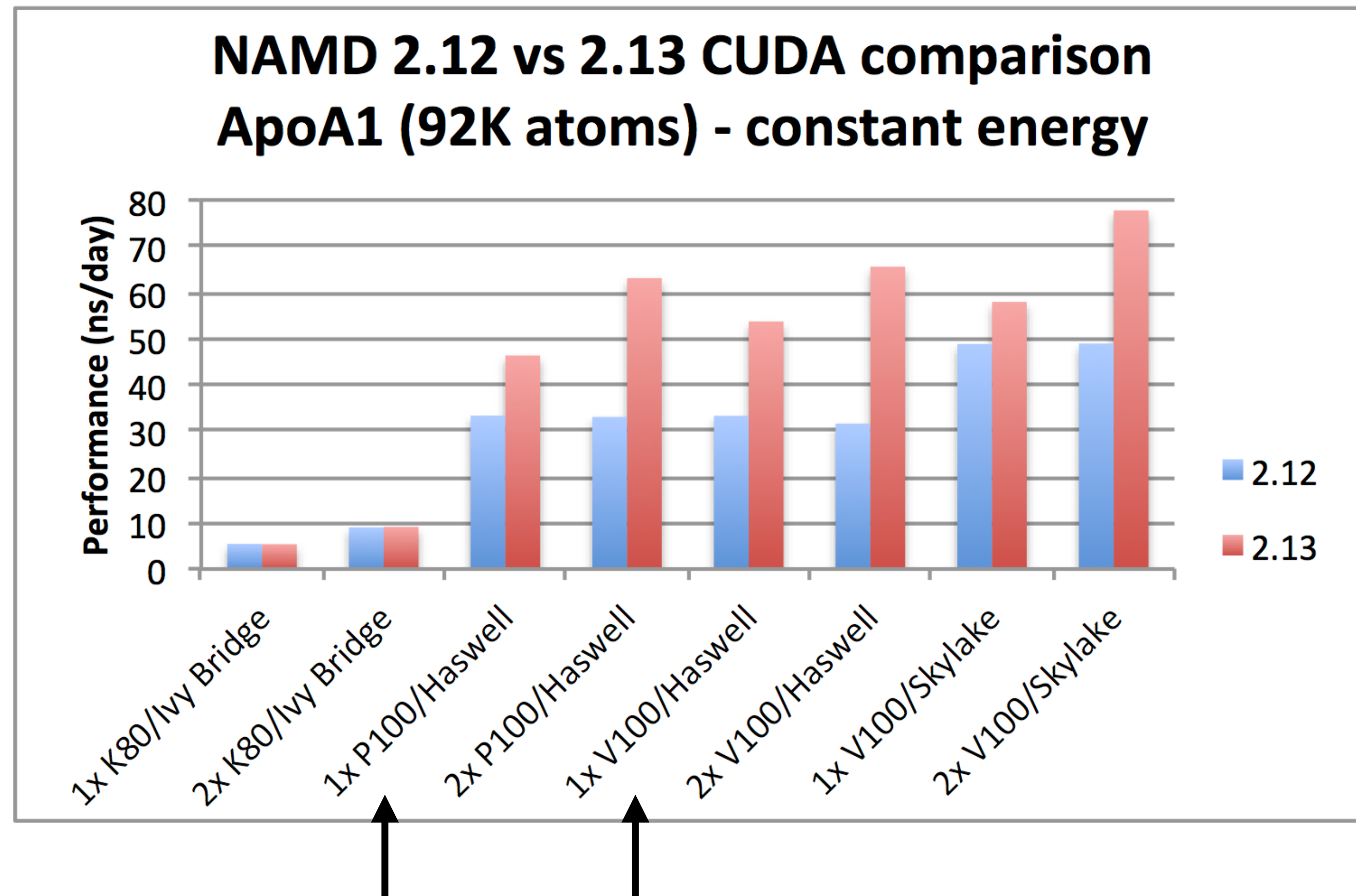
= 5x2x2 grid = 21M Atoms



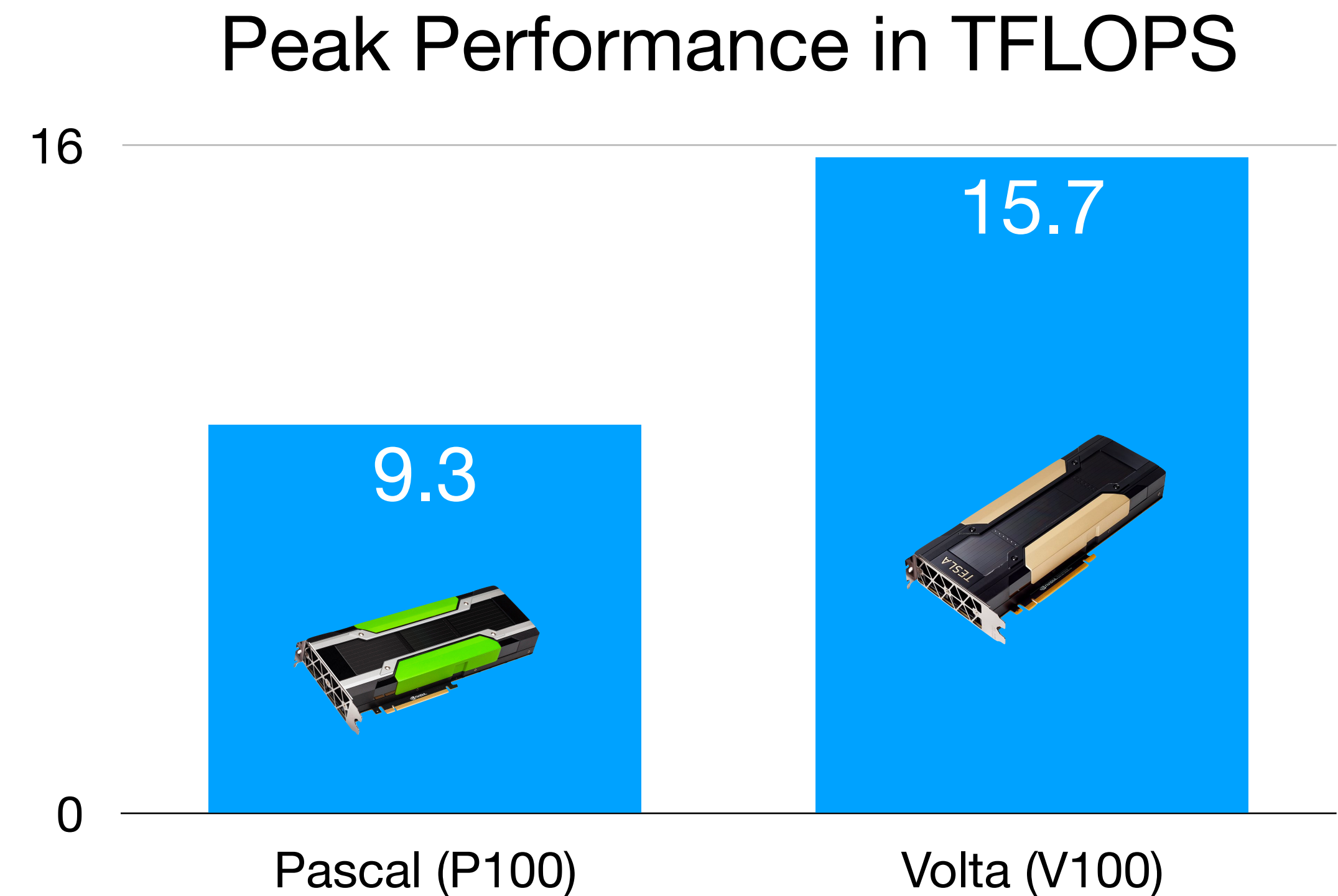
= 7x6x5 grid = 224M Atoms



# Benchmarks on single nodes and Newer GPUs Reveals Problems



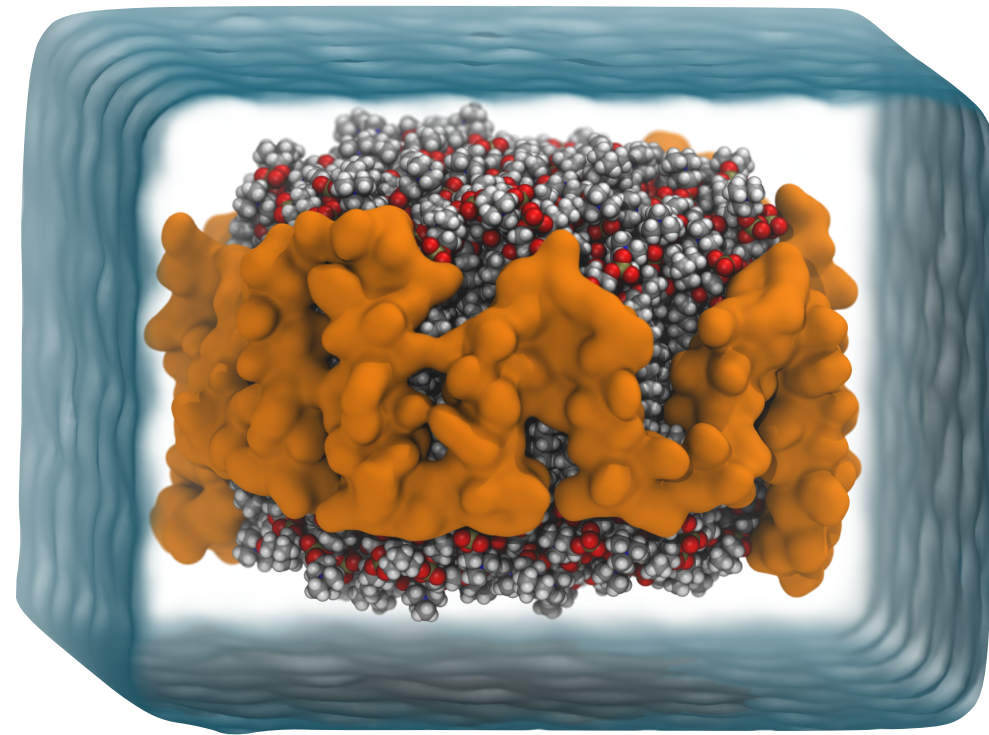
**NAMD 2.13 (2018) has ~20% perf improvement from P100 to V100**



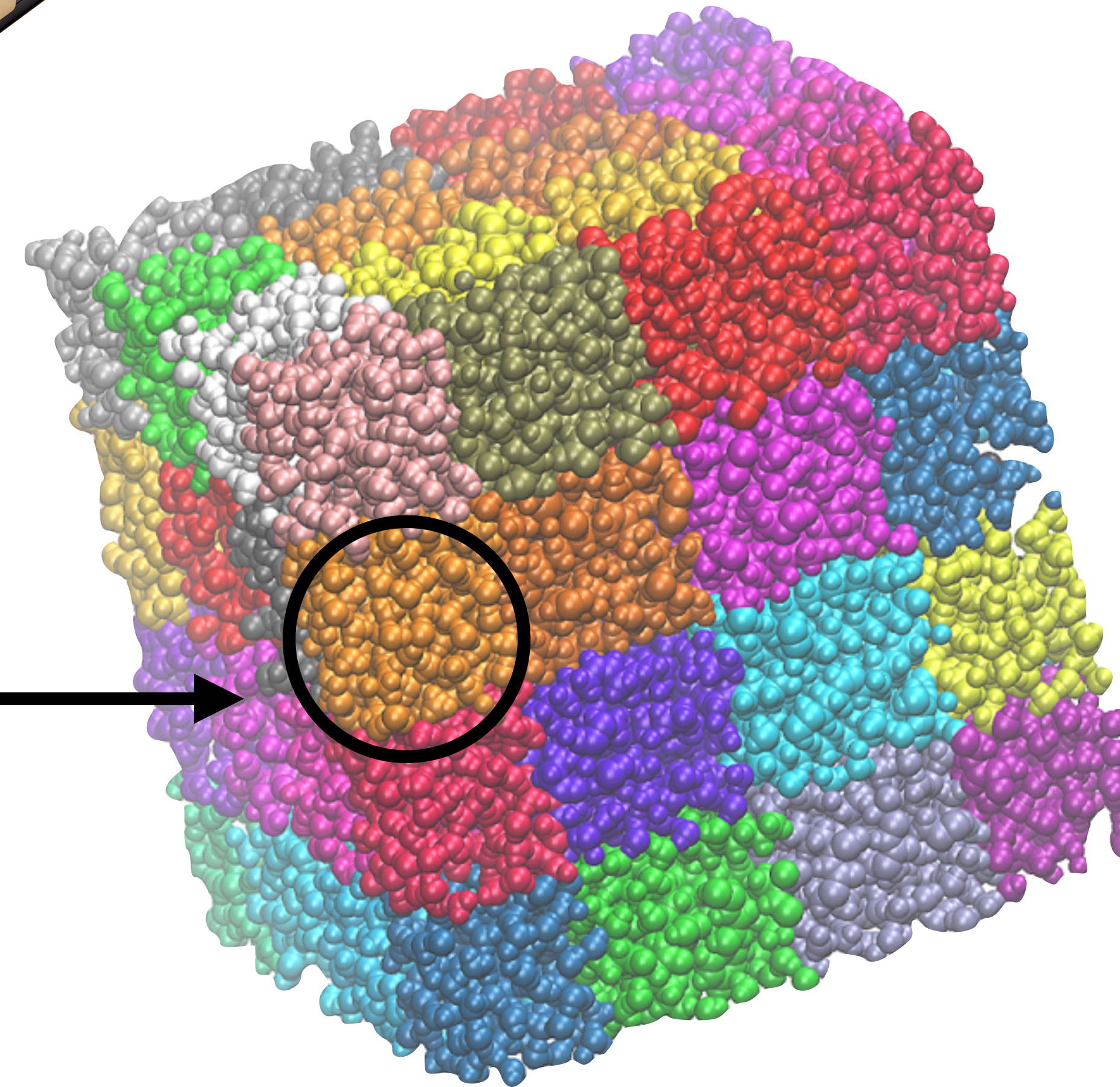
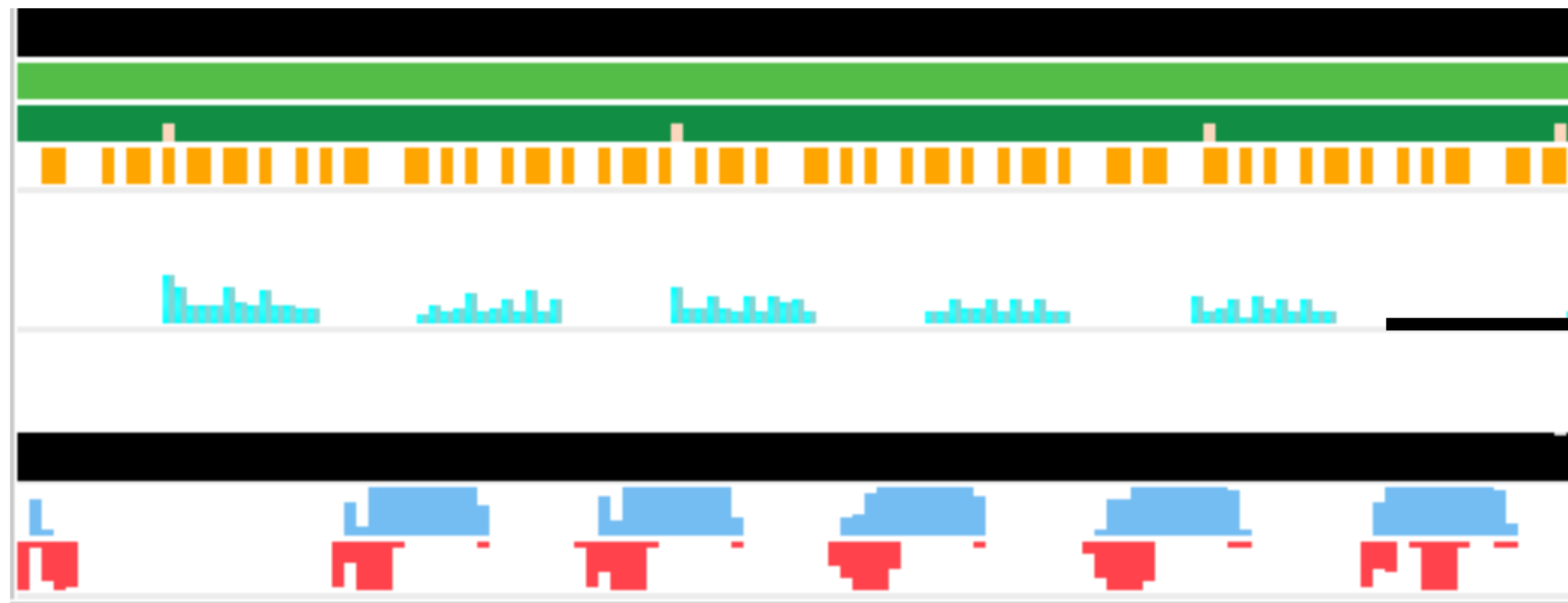
**Hardware has ~70% perf improvement!**



# Profiling on Modern GPUs



Profiling ApoA1, 92k atoms  
NAMD 2.13, 16 cores and 1  
GPU Volta



Gaps in the blue strip = GPU is idle!



# NAMD 2.13 had Limited GPU Performance

- Offloading force calculation was not enough!
- Overall utilization of modern GPUs is limited
- We want better single GPU performance
  - Majority of MD users run system sizes  $< 1\text{M}$  atoms on a single GPU
- Must transition from **GPU-accelerated** to **GPU-resident!**

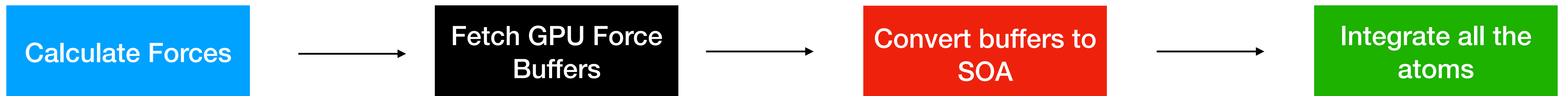
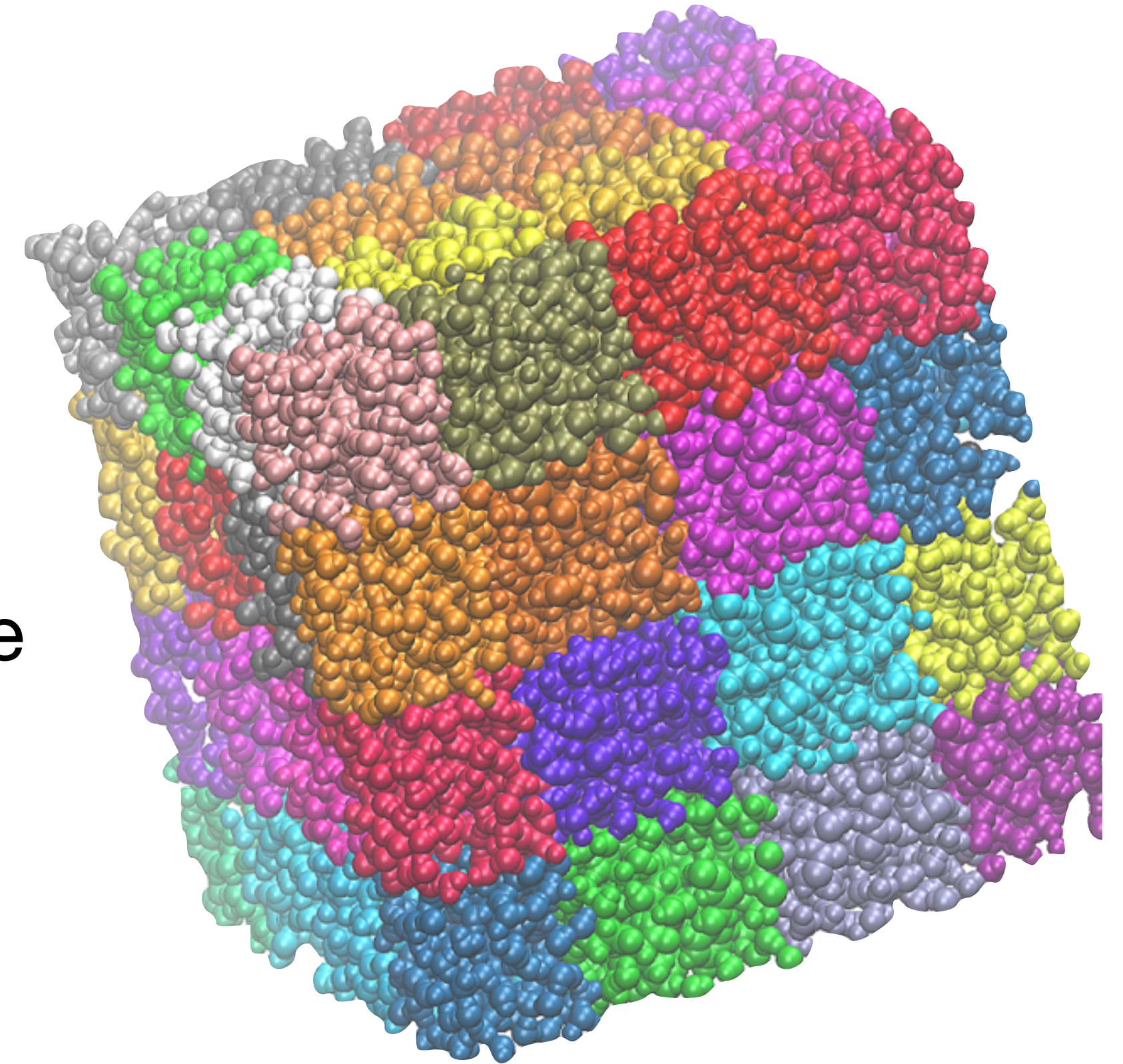


The DGX-2 has 16 V100 GPUs but only 48 CPU cores: We need to do more GPU work with less CPU Power



# NAMD 3.0: GPU-Resident NAMD

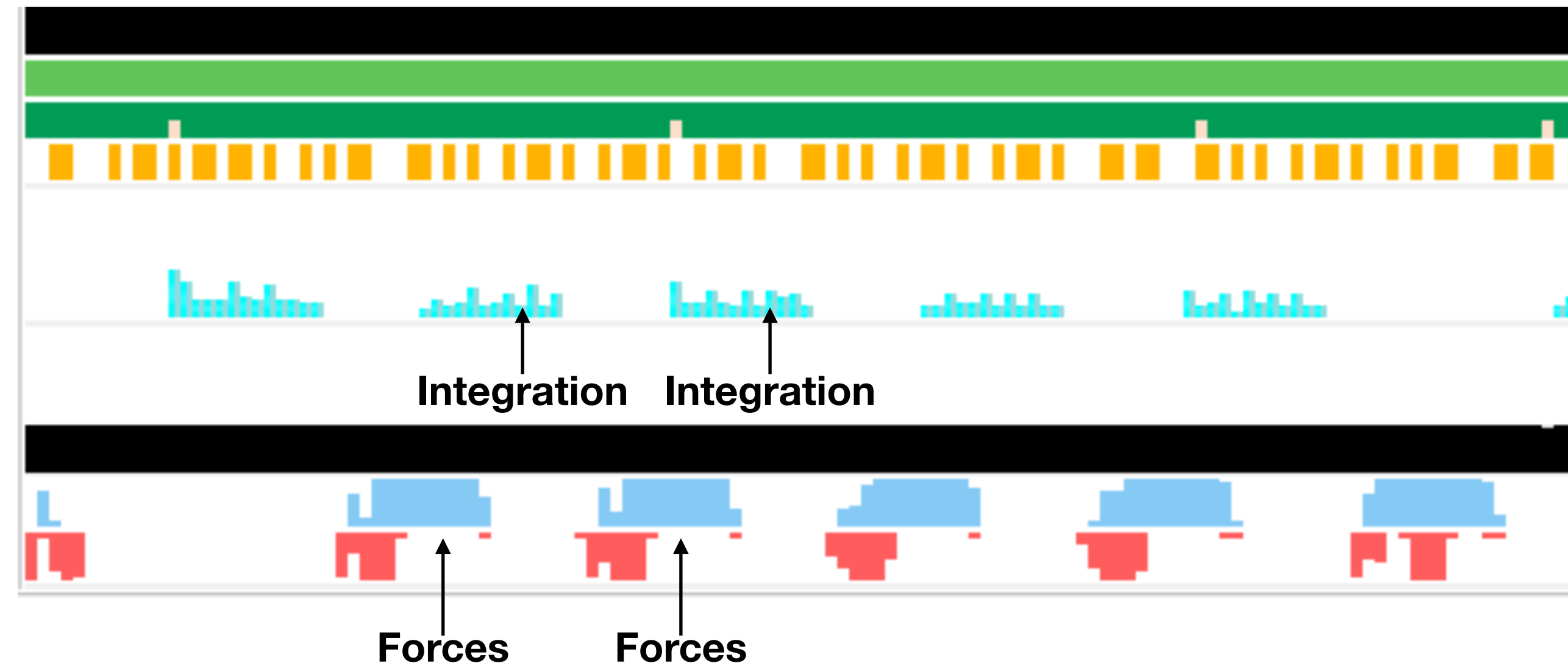
- Fetches GPU force buffers directly from the force module
- Bypass any CPU-GPU memory transfers - *only call GPU kernels!*
- Convert forces in a *structure-of-arrays* (SOA) data structure using the GPU
- Invoke GPU Integration Tasks Once



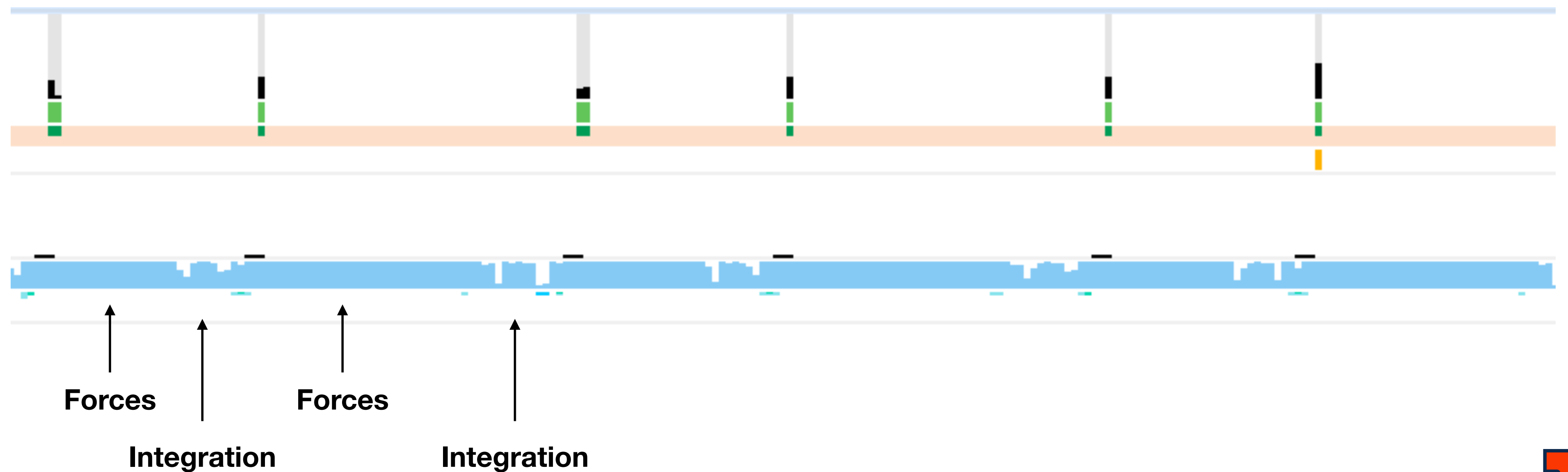


# NAMD 3.0 has Better GPU Utilization

NAMD 2.14  
Gaps between GPU Tasks



NAMD 3.0  
No CPU Bottlenecks





# NAMD 3.0: Performance on Different Systems

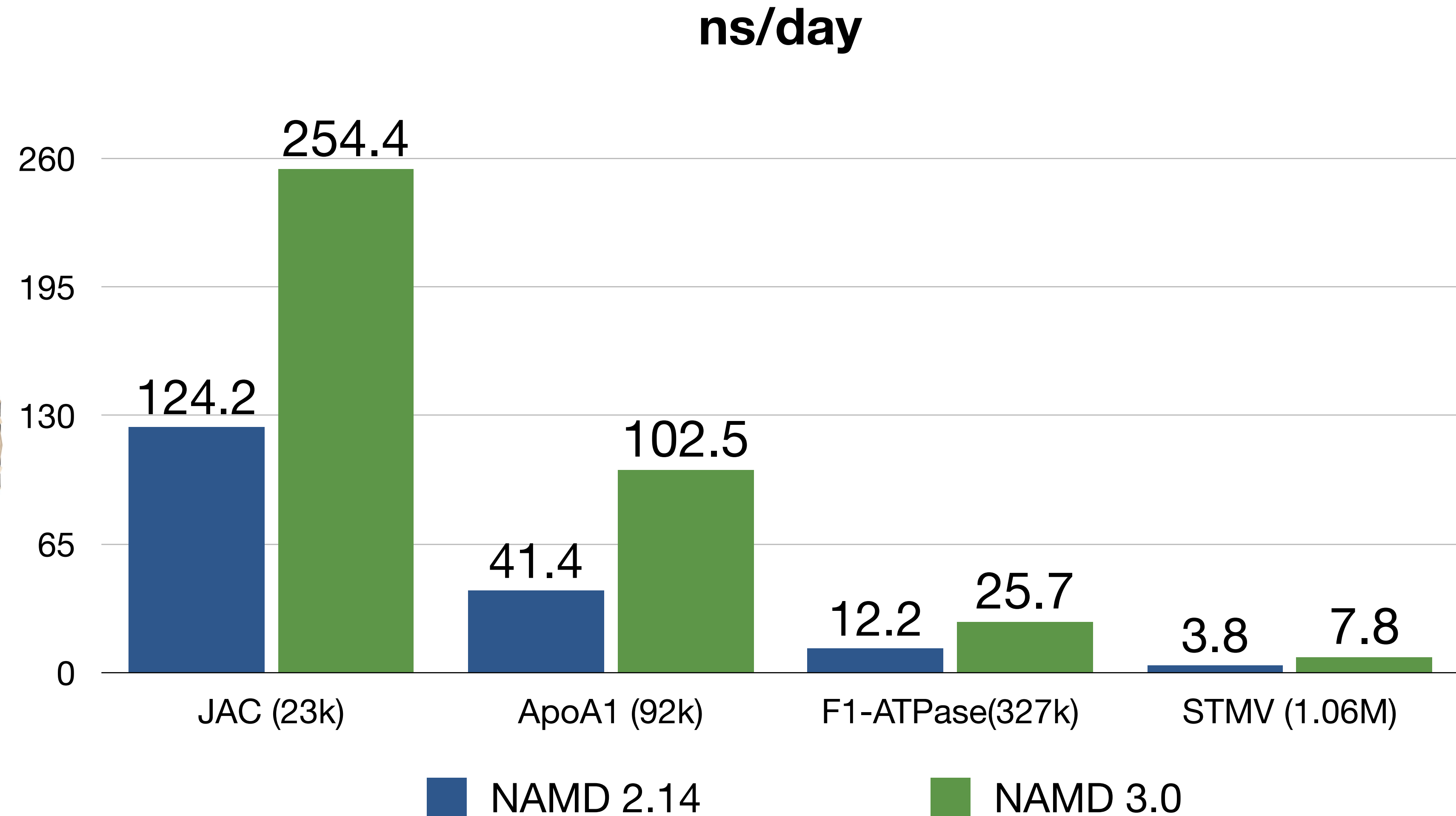
**NVE**  
**12A Cutoff**  
**2fs timestep**



Intel Xeon E5-2650  
V2 w/ 16 physical  
cores

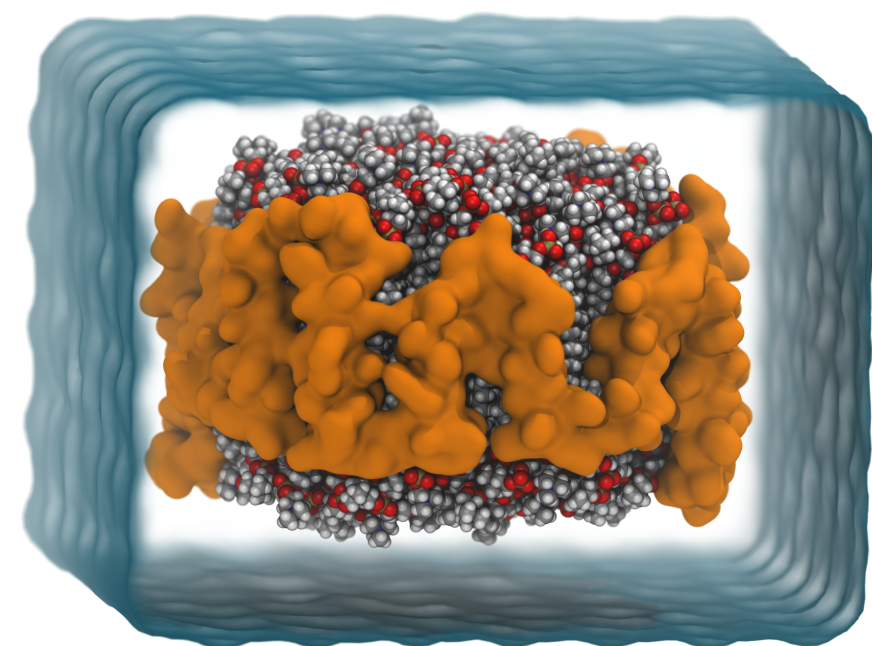


Nvidia Titan V





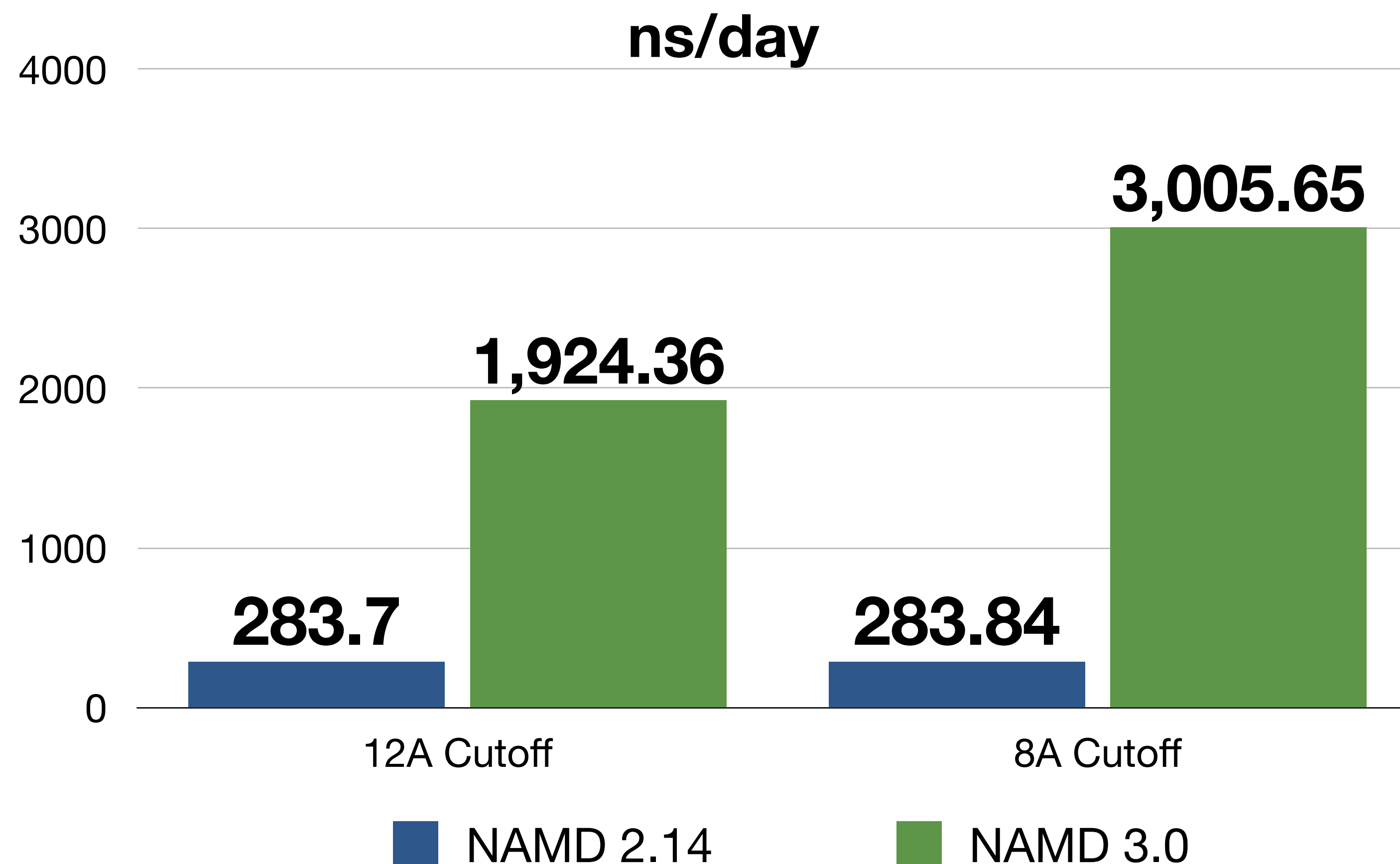
# NAMD 3.0: Multi-Copy Performance - Aggregate Throughput With DGX-2



**ApoA1**  
**92k atoms**

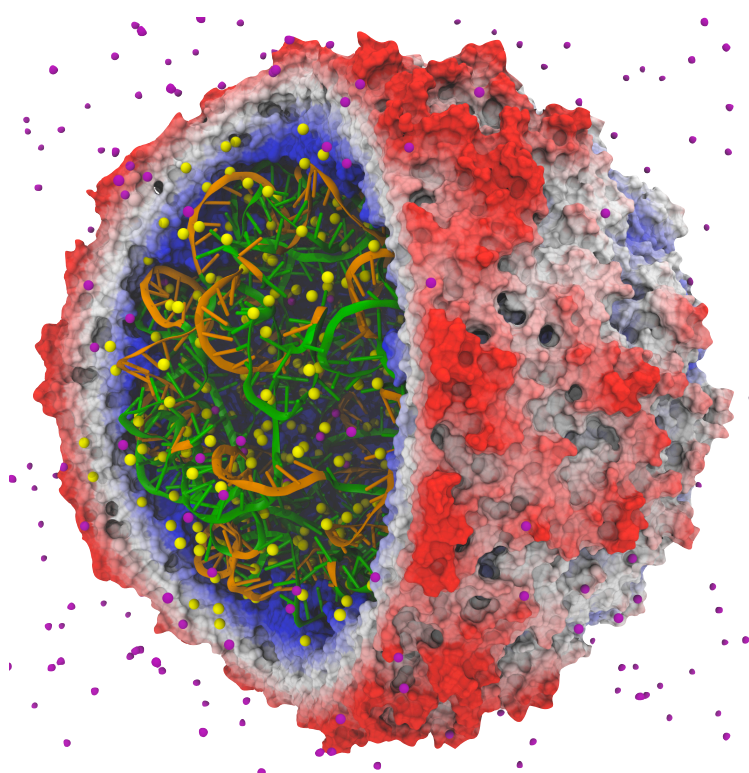


**16 Replicas**  
**1 for each NVIDIA V100**

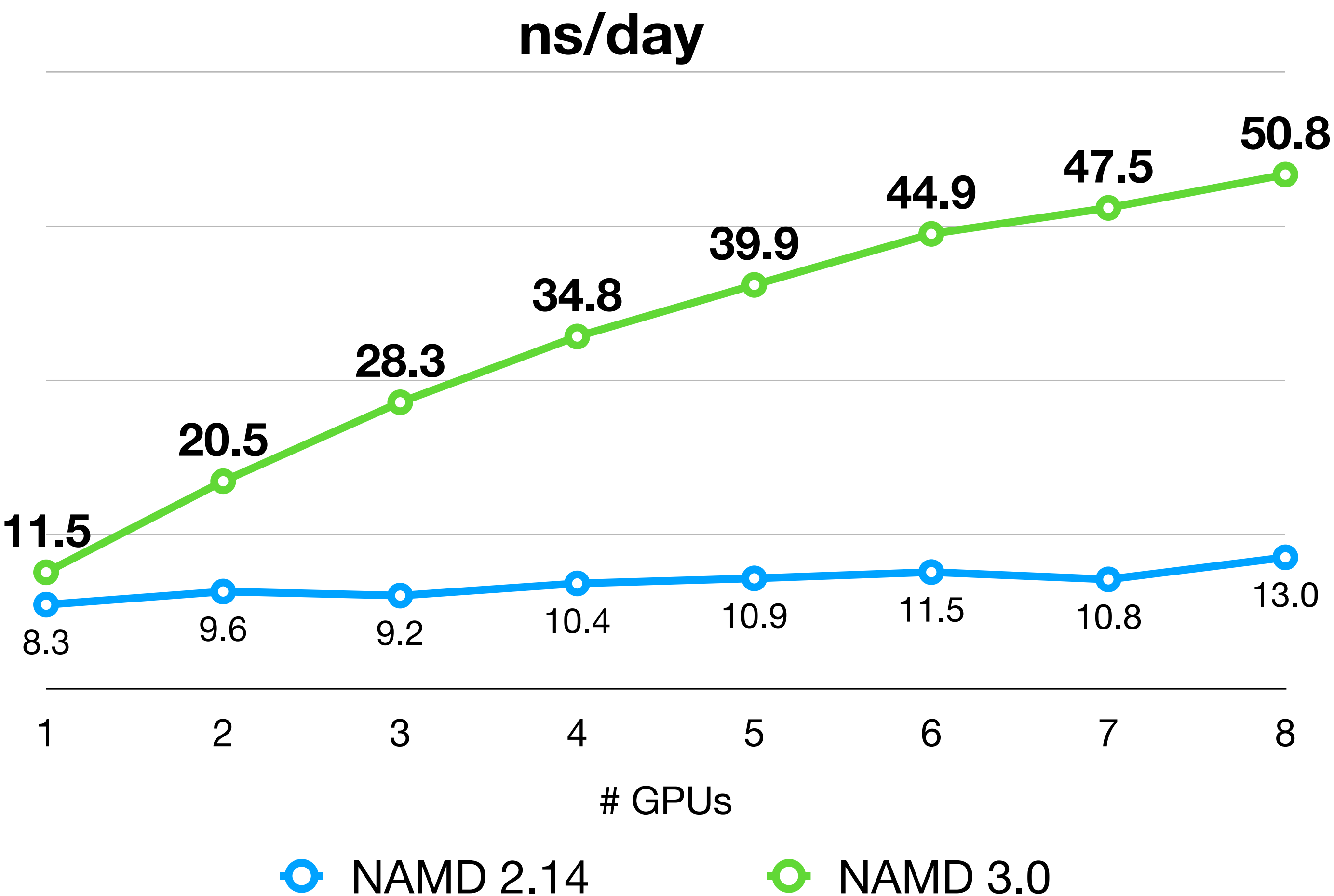




# NAMD 3.0: Single trajectory - Multiple GPU Performance



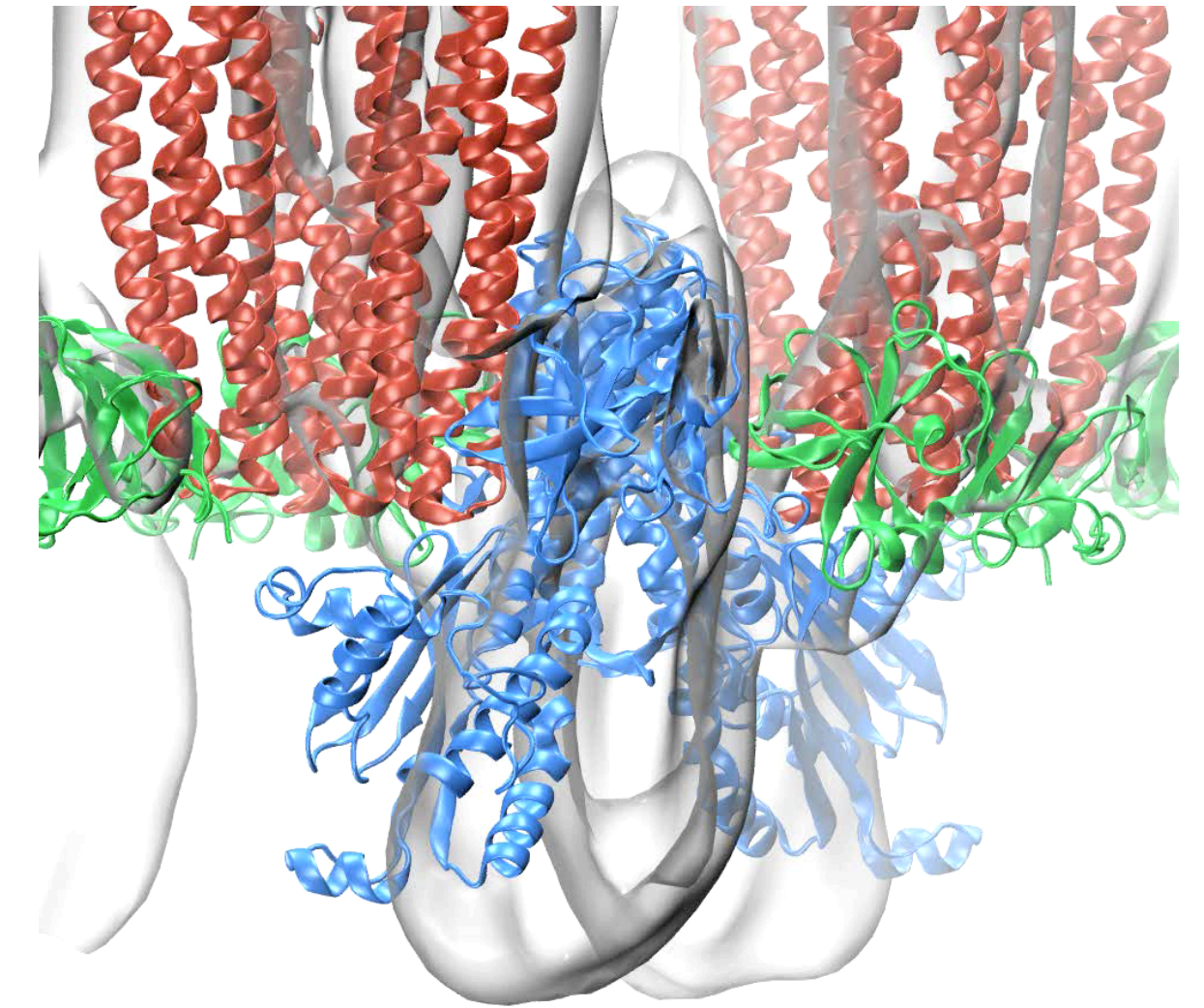
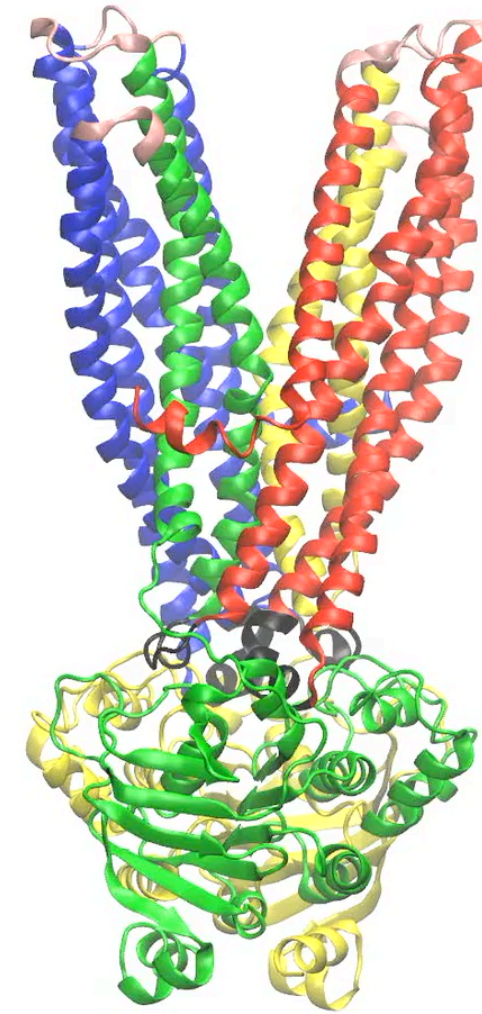
**STMV**  
**1.06M atoms**  
**2fs timestep**  
**No PME yet**



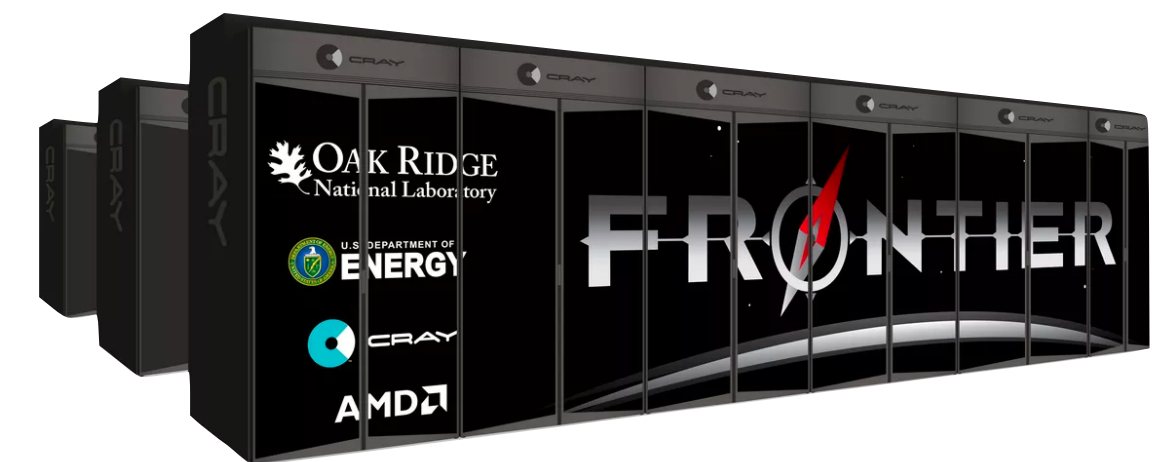


# Challenge: Feature-Complete GPU-resident scheme

- Supports a subset of NAMD features so far
- Incorporate external biasing libraries (Mainly Colvars)
- Introduce support to different GPU Vendors



Intel GPUs



AMD GPUs



# Acknowledgements



Prof. Emad Tajkhorshid  
NIH Center for Macromolecular Modeling  
and Bioinformatics Director  
University of Illinois



Prof. Chris Chipot  
Centre National de la Recherche  
Scientifique Research Director  
University of Illinois



Prof. Rafael Bernardi  
Physics Department  
Auburn University



John Stone  
Senior Research Programmer  
University of Illinois



David Hardy  
Senior Research Programmer  
University of Illinois



Mohammad Soroush  
Research Programmer  
University of Illinois



Peng Wang  
HPC Developer  
NVIDIA



Josh Vermaas  
Computational Scientist  
Oak Ridge National Lab



Jim Phillips  
NCSA  
University of Illinois



# Thank You!



jmaia@ks.uiuc.edu

@jdmaia92

[www.ks.uiuc.edu/~jmaia](http://www.ks.uiuc.edu/~jmaia)