Porting LAMMPS to the Titan System

W. Michael Brown
National Center for Computational Sciences
Oak Ridge National Laboratory

Cray Technical Workshop
October 10, 2012

LAMMPS

- Large-scale Atomic/Molecular Massively Parallel Simulator
  - http://lammps.sandia.gov
- General Purpose Classical Molecular Dynamics Software
  - Simulations for biology, materials science, granular, mesoscale, etc.
- Open source (GPL), with a user community
  - 1700 Citations
  - 100K downloads, 25K mail list messages
  - 100 contributors to code base
- Flexible and extensible:
  - 80% of code-base is add-ons by developers and users
  - styles: atom, pair, fix, compute, etc

Force Fields

- Biomolecules: CHARMM, AMBER, OPLS, COMPASS (class 2), Gromacs, long-range Coulombics via PPPM, point dipoles, ...
- Polymers: all-atom, united-atom, coarse-grain (bead-spring FENE), bond-breaking, bond-forming, ...
- Materials: EAM/MEAM for metals, Buckingham, Morse, Yukawa, Tersoff, COMB, AI-REBO, ReaxFF, GAP, …
- Mesoscale: granular, DPD, SPH, PD, colloidal, aspherical, triangulated, …

Porting Strategy

- LAMMPS is a large legacy code with many routines and is constantly growing with new features and algorithms
  - Impractical to port the entire code base
  - Use a “modular” acceleration strategy that targets computationally intensive routines while allowing full compatibility with LAMMPS many features/options
  - In order to get efficient acceleration with this strategy, we must
    - Parallelize routines that are not ported on the CPU cores
    - Exploit as much CPU/accelerator computation and data transfer concurrency as possible

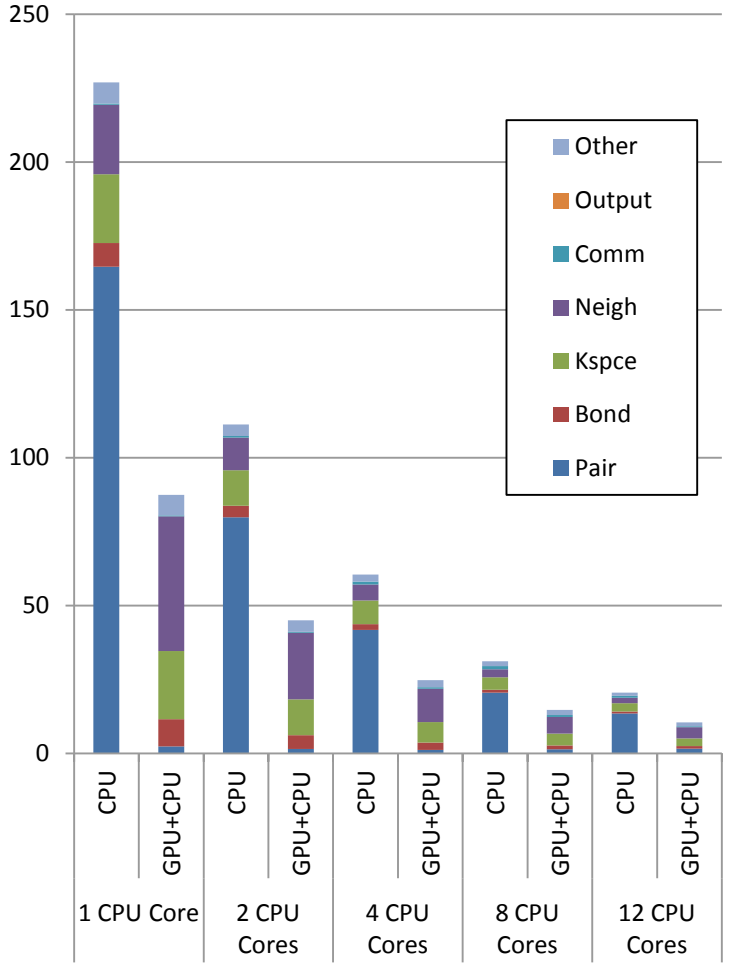
LAMMPS with Accelerators

- Acceleration for neighbor-list builds, short-range force calculation, and long-range electrostatics
- CPU/Accelerator/Data-transfer concurrency to achieve efficient simulation on hybrid nodes
- Use existing spatial decomposition for MPI processes
- Variety of parallel decompositions for the accelerator depending on the routine
  - Atom, Force, Spatial, Mesh, Sorts, Etc.
- Deterministic Algorithms
  - Same result with same number of procs/random seed
- Portability to future CPUs, GPUs, APUs, and other accelerators
    - Geryon library

Brown, W.M., Wang, P. Plimpton, S.J., Tharrington, A.N. *Comp Phys Comm.* 2011. 182: p. 898-911.
Brown, W.M., Thompson, A.P., Schultz, P.A. *Comp Phys Comm.* 2012. 183: p. 449-459.

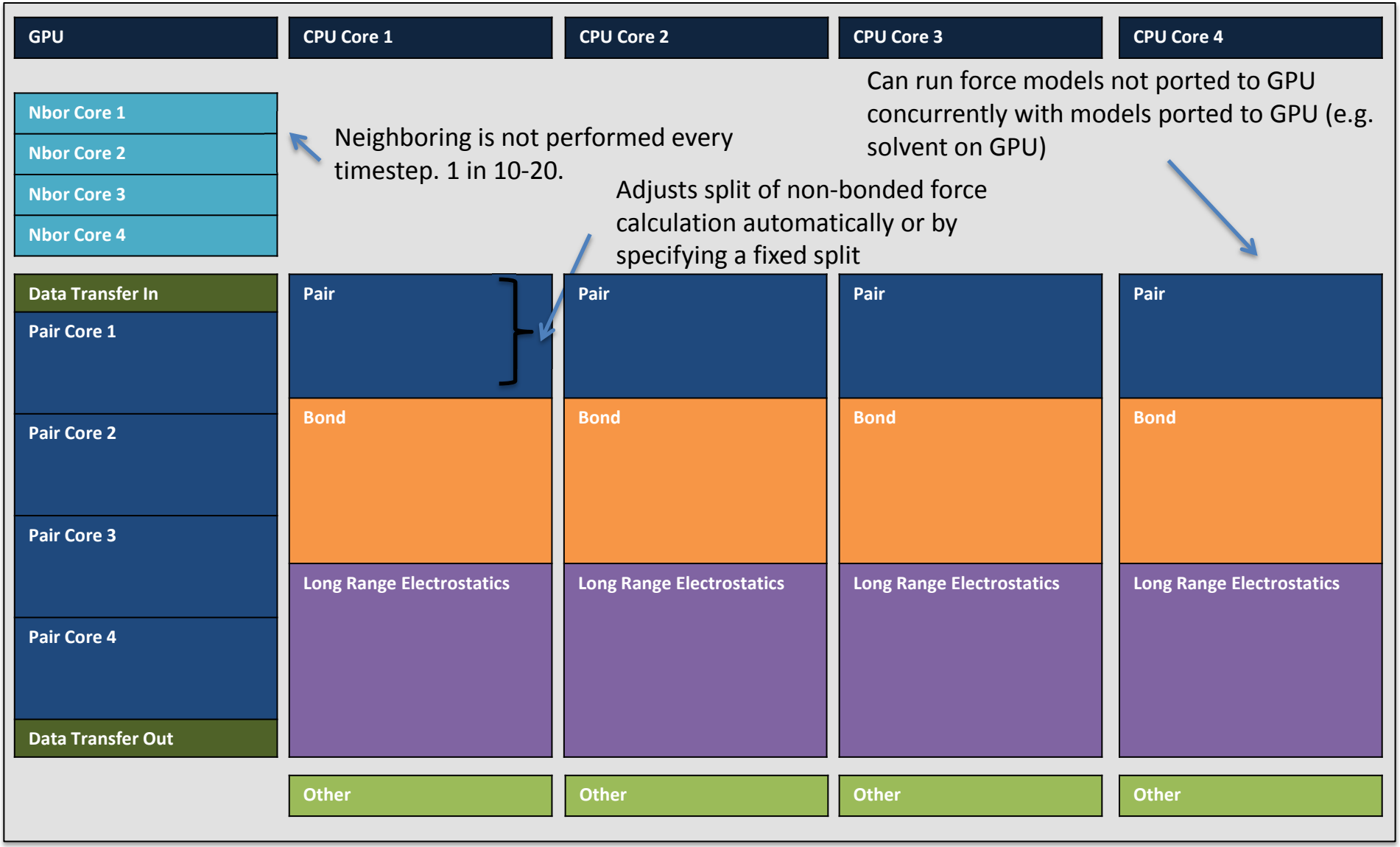
1. Minimize Porting Effort - Need to parallelize code not ported for the accelerator on the CPU cores

- Example: Rhodopsin Benchmark on a box with 12 CPU cores and 1 GPU
  - Here, only the non-bonded “pair” force calculation is accelerated for an example
  - Speedup on 1 CPU core with a GPU is only 2.6x with this approach
  - Speedup vs 12 CPU cores is still > 2x because we use all 12 cores in addition to the GPU
    - MPI Processes Share the Accelerator
    - Trivial parallelization of host code
    - Allows for trivial (from a programmer view) overlap of host-device comm and computation (but not optimal)
    - Improves data locality for random access with multiple kernel executions on smaller subdomains
    - Expands options for CPU/accelerator concurrency…

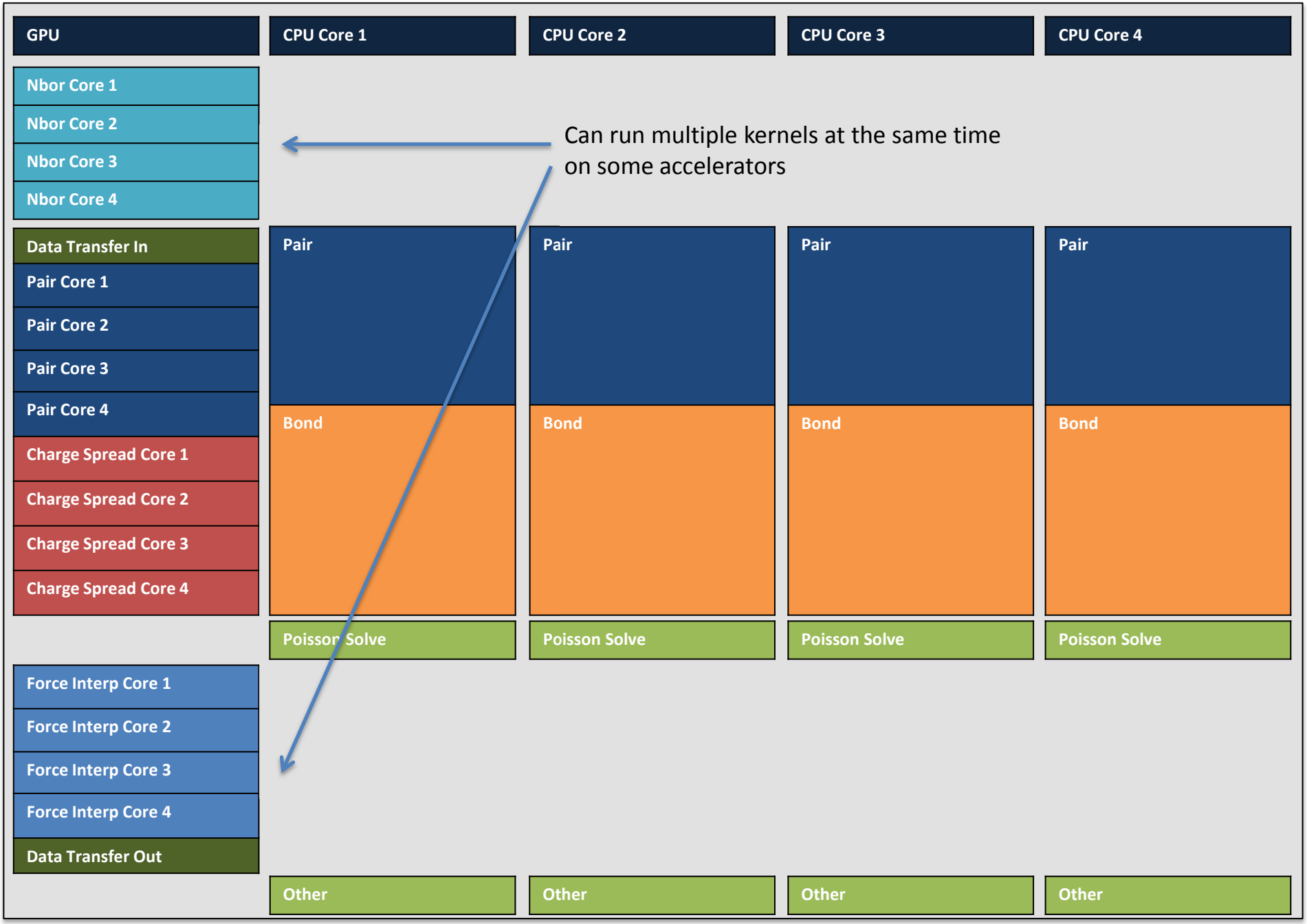


2. Host-Device Load Balancing

(- PPPM Acceleration) *Not to scale*

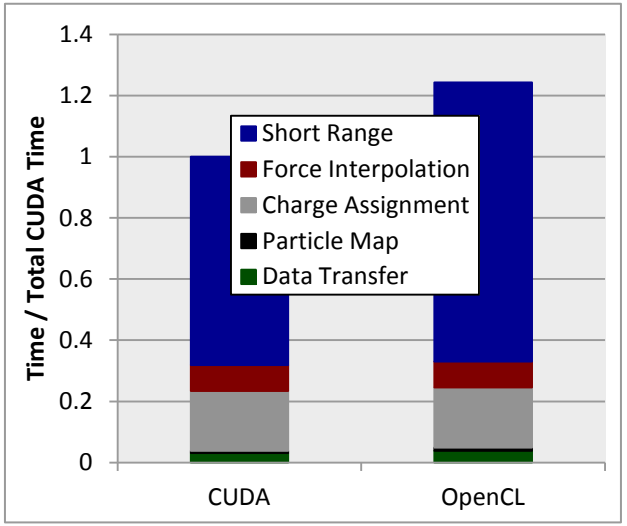


Host-Device Load Balancing (+PPPM Acceleration)



Geryon Library

- Allows same code to compile with CUDA Runtime, CUDA Driver, and OpenCL APIs
- Allows portability to accelerators, CPUs, and any future chips with OpenCL support
- Simple classes allow for more succinct code than CUDA-Runtime
  - Change from one API to another by simply changing the namespace
  - Use multiple APIs in the same code
  - Lightweight (only include files – no build required)
  - Manage device query and selection
  - Simple vector and matrix containers
  - Simple routines for data copy and type casting
  - Simple routines for data I/O
  - Simple classes for managing device timing
  - Simple classes for managing kernel compilation and execution



http://users.nccs.gov/~wb8/geryon/index.htm

Early Optimizations for the XK6

- Using MPI-only on multicore nodes can impact performance at scale
  - Increased MPI communication times
    - MPI Cartesian topologies (e.g. MPI_Cart_create) are designed to allow vendors to optimize process mapping for grids
      - Typically do nothing to optimize for multicore nodes
    - Implemented a two-level domain decomposition algorithm to minimize off-node communications
  - Increased K-space times for P^3M calculations
    - Many processes involved in effectively all-to-all communications for 3D FFTs
    - Use separate process partitions for the K-space calculation
      - Implemented by Yuxing Peng and Chris Knight from the Voth Group at the University of Chicago
- Improved data layout for neighbor list builds with an arbitrary number of threads assigned to each atom
    - Can efficiently use 4X the number of threads for some simulations
- Implementation of the embedded atom method (and other new models)
    - Many-body potential for simulation of metals

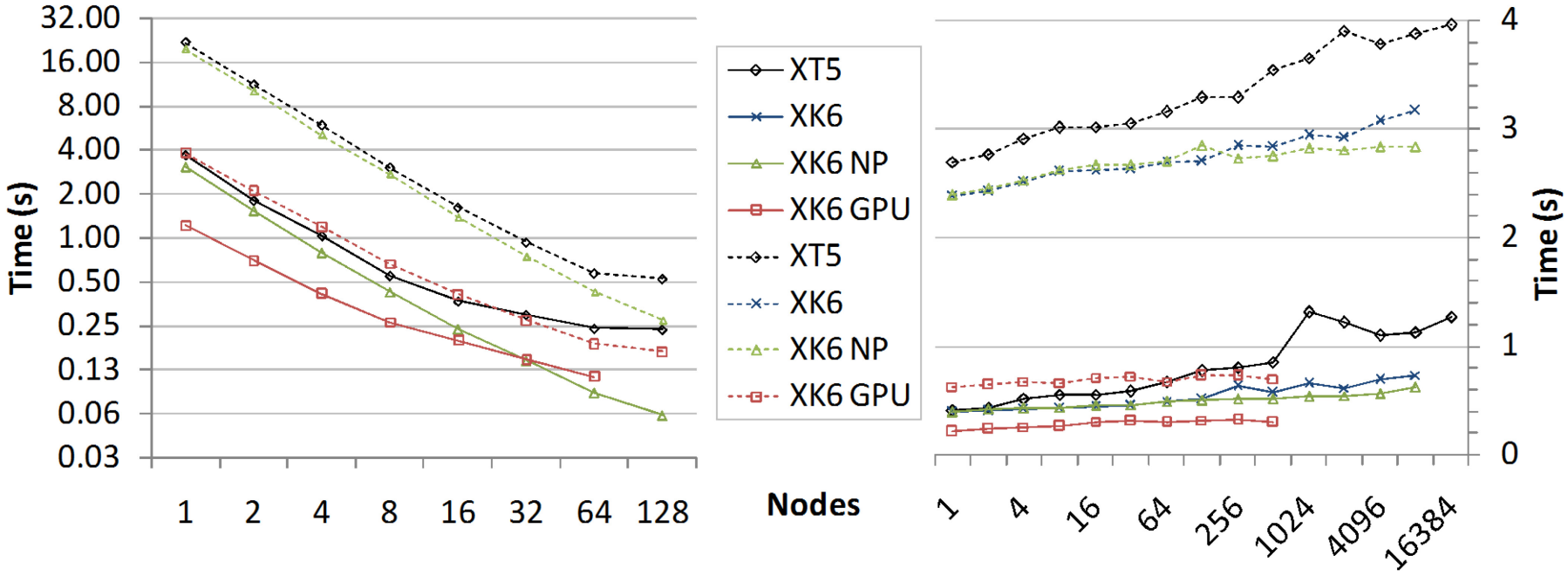
X2090 Benchmark Results

- Run on the Titan development partition with Fermi+ GPUs
- Benchmarks with acceleration used mixed-precision as compared to double precision
- Left: Strong Scaling for fixed-size simulations of approximately 256K particles
- Right: Weak Scaling with ~32K particles/node
- XK6 NP results use a new MPI process to grid mapping and a separate partition for long-range electrostatics (as do the GPU results)
- The next 4 plots are speedups versus the XT5 on a node per node basis
  - XK6 only comparisons later

Brown, W. M., Nguyen, T.D., Fuentes-Cabrera, M., Fowlkes, J. D., Rack, P. D., Berger, M., Bland, A. S. A**n Evaluation of Molecular Dynamics Performance on the Hybrid Cray XK6 Supercomputer**. *Proceedings of the ICCS Workshop on Large Scale Computational Physics*. 2012. Published in *Procedia Computer Science*, 2012. 9 p. 186-195.

Atomic Fluid

- Atomic fluid - microcanonical ensemble, Lennard- Jones potential, reduced density 0.8442, neighbor skin 0.3σ, cutoffs of 2.5σ and 5.0σ

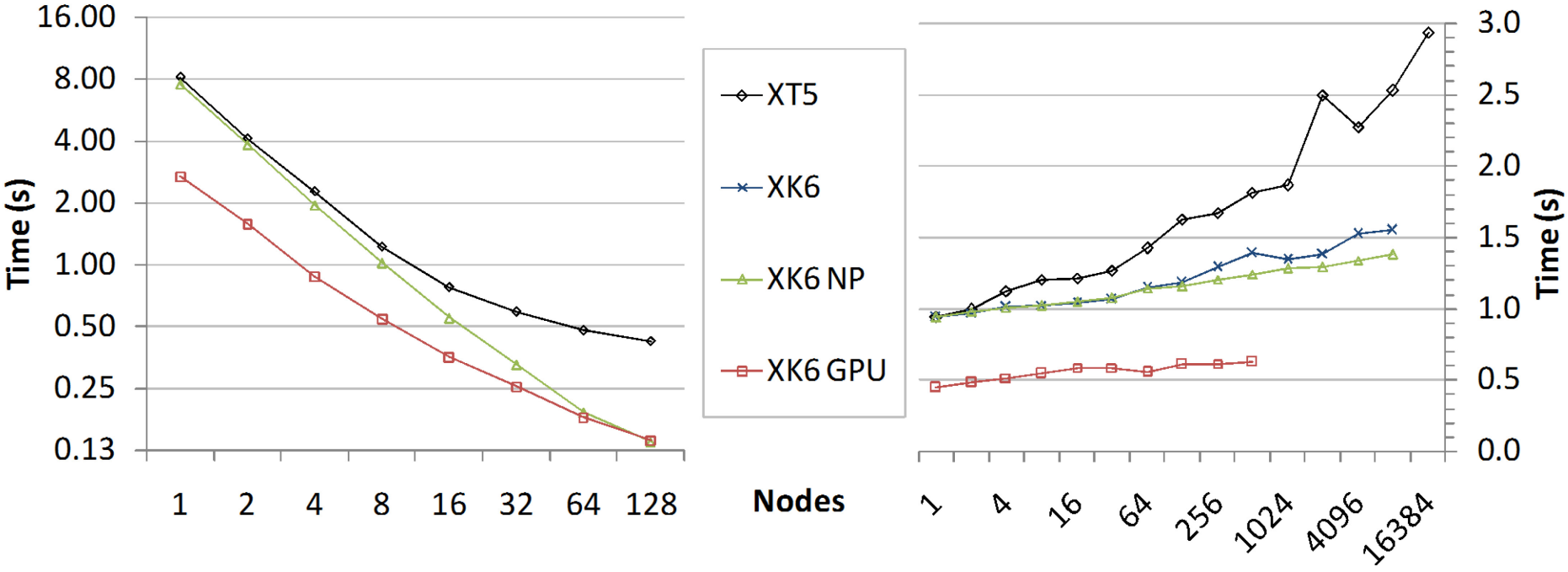


XK6 Single Node: 1.17X
XK6 GPU Single Node: 3.03X, 5.68X

XK6 Single Node: 1.04X
XK6 GPU Single Node: 1.92X, 4.33X
XK6 GPU 512 Nodes: 2.92X, 5.11X
NP gives up to 20% improvement

Bulk Copper

- Copper metallic solid in the microcanonical ensemble. The force cutoff is 4.95Å with a neighbor skin of 1.0A.
  - Requires additional ghost exchange during force calculation for electron densities

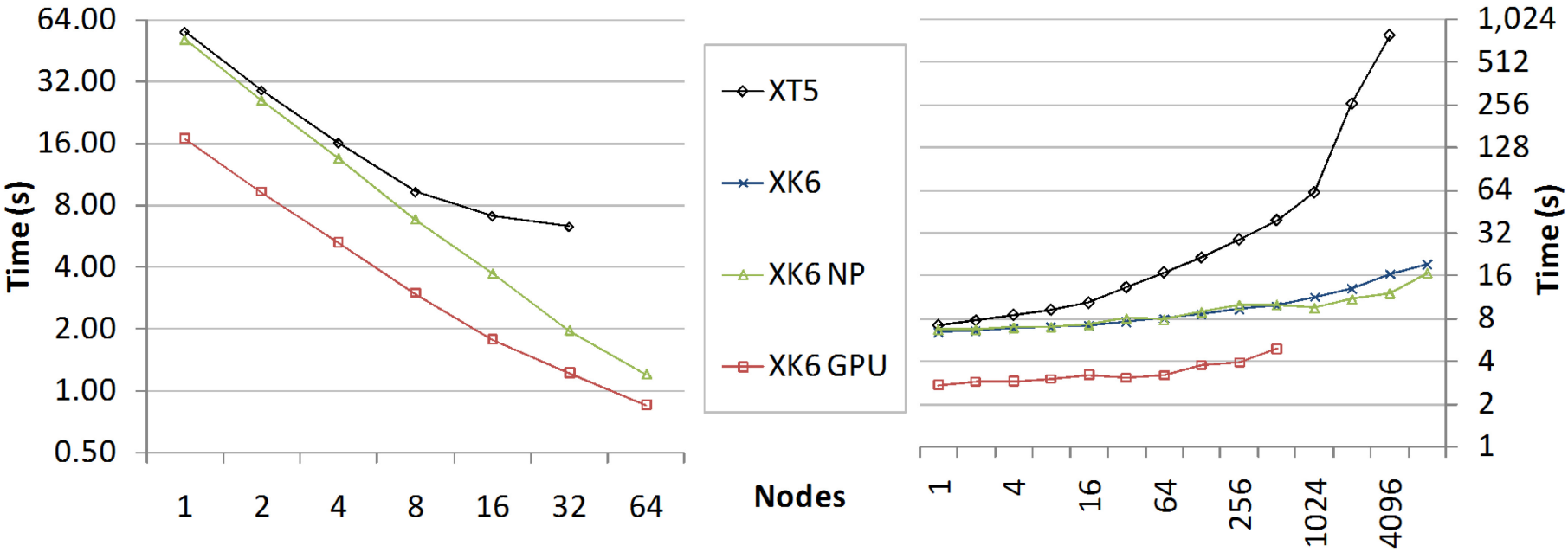


XK6 Single Node: 1.07X
XK6 GPU Single Node: 3.05X

XK6 Single Node: 1.0X
XK6 GPU Single Node: 2.12X
XK6 GPU 512 Nodes: 2.90X
NP gives up to 12% improvement

Protein

- All-atom rhodopsin protein in a solvated lipid bilayer with the CHARMM force field. Isothermal-isobaric ensemble, SHAKE constraints, 2.0fs timestep. Counter-ions, water, 8Å/ 10Å cutoff
  - Long range electrostatics are calculated with P3M.

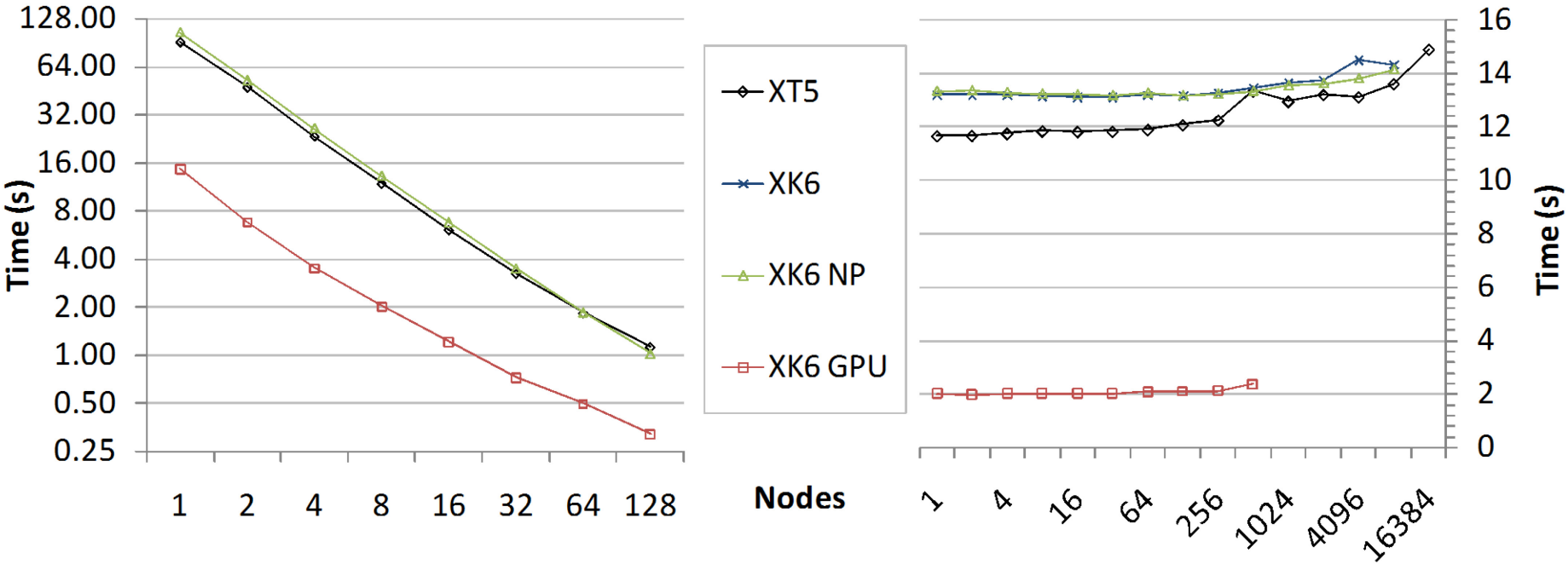


XK6 Single Node: 1.1X
XK6 GPU Single Node: 3.3X

XK6 Single Node: 1.1X
XK6 GPU Single Node: 2.6X
XK6 GPU 512 Nodes: 8X
NP gives up to 27% improvement

Liquid Crystal

- Liquid crystal mesogens are represented with biaxial ellipsoid particles, Gay-Berne potential, isotropic phase, isothermal-isobaric ensemble, 4σ cutoff with a 0.8σ neighbor skin
  - High arithmetic intensity

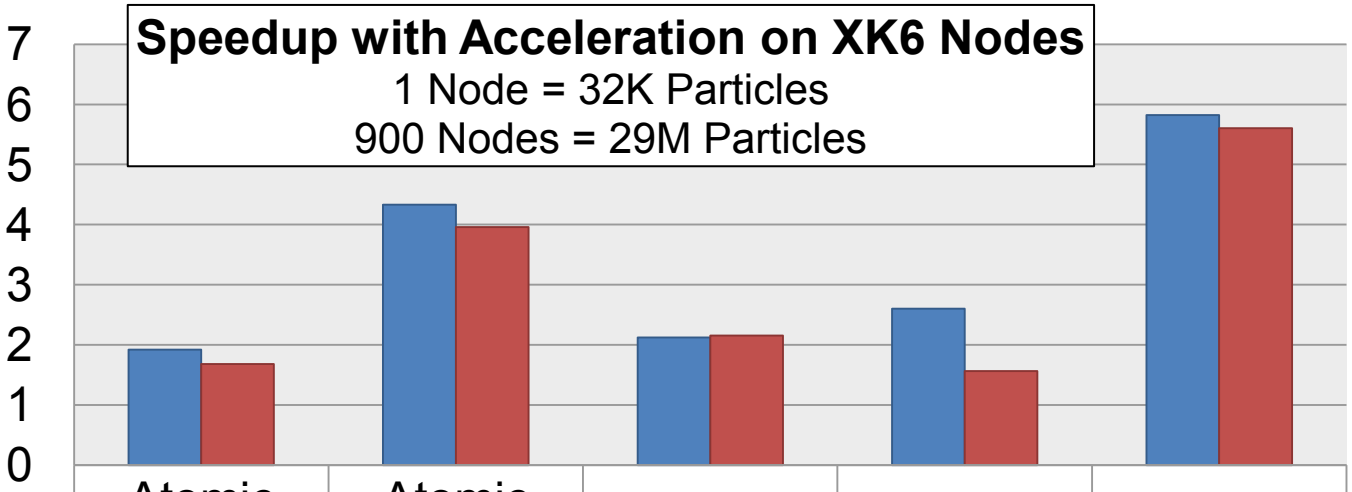


XK6 Single Node: .9X
XK6 GPU Single Node: 6.23X

XK6 Single Node: .9X
XK6 GPU Single Node: 5.82X
XK6 GPU 512 Nodes: 5.65X

XK6 vs XK6+GPU Benchmarks

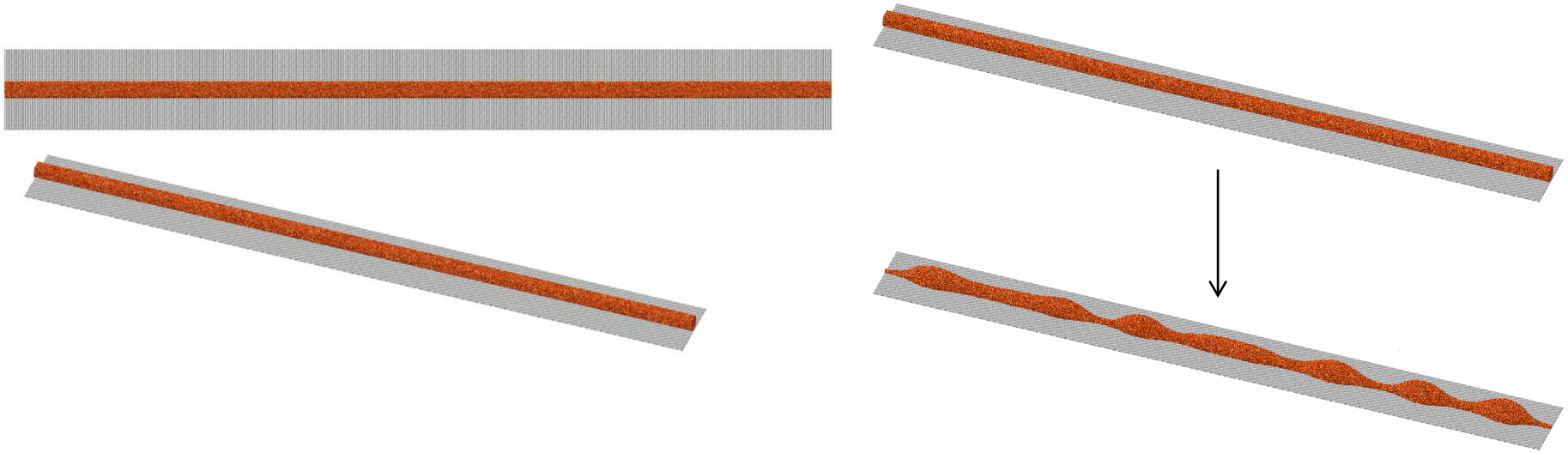
- 32K per node



	Atomic Fluid (cutoff = 2.5σ)	Atomic Fluid (cutoff = 5.0σ)	Bulk Copper	Protein	Liquid Crystal
Speedup (1 Node)	1.92	4.33	2.12	2.6	5.82
Speedup (900 Nodes)	1.68	3.96	2.15	1.56	5.60

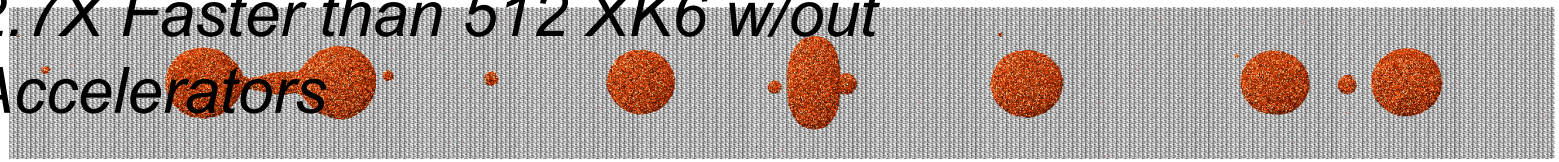
Example Science Problems 1.

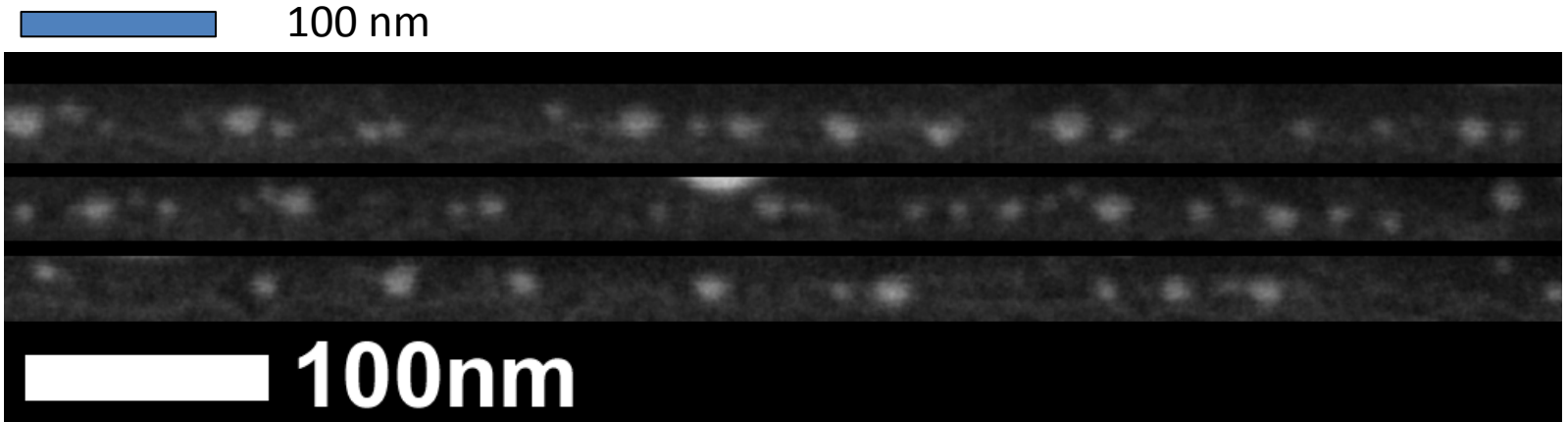
- Controlling the movement of nanoscale objects is a significant goal of nanotechnology.
- Pulsed laser melting offers a unique opportunity to dictate materials assembly where rapid heating and cooling rates and ns melt lifetimes are achievable
- Using both experiment and theory we have investigated ways of controlling how the breakage occurs so as to control the assembly of metallic nanoparticles
- Here, we illustrate MD simulation to investigate the evolution of the Rayleigh-Plateau liquid instability for copper lines deposited on a graphite substrate
- Simulations are performed with GPU acceleration on Jaguar at the same scales as experiment



11.4M Cu Atom Simulations on Graphitic Substrate

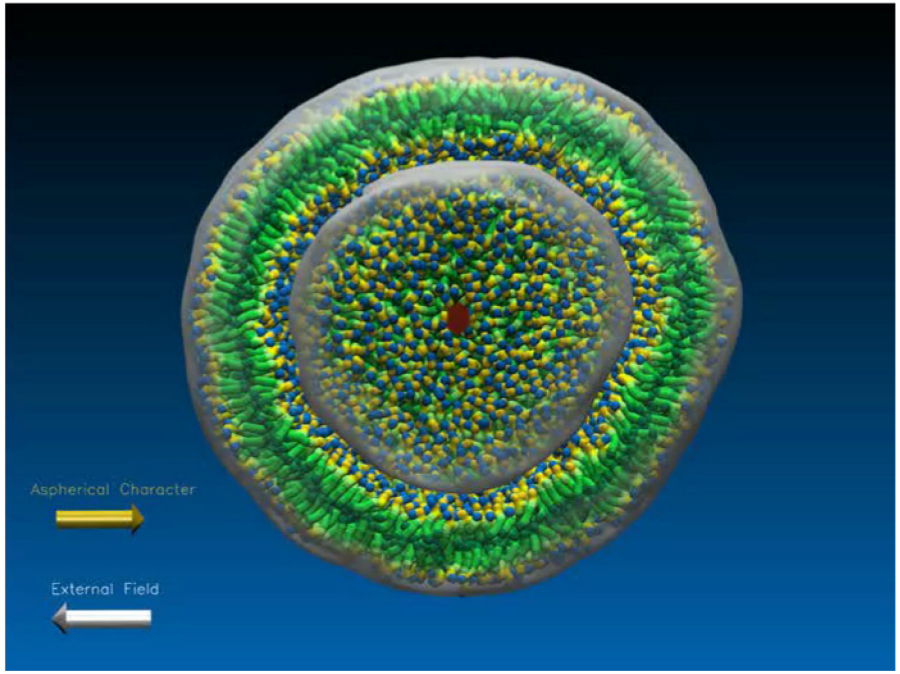
- *2.7X Faster than 512 XK6 w/out Accelerators*





Example Science Problems 2.

- Membrane fusion, which involves the merging of two biological membranes in a controlled manner, is an integral part of the normal life cycle of all living organisms.
- Viruses responsible for human disease employ membrane fusion as an essential part of their reproduction cycle.
- Membrane fusion is a critical step in the function of the nervous system
  - Correct fusion dynamics requires realistic system sizes
- Kohlmeyer / Klein INCITE Award



39M Particle Liposome System Simulated at 115 ns/day on 900 XK6 Nodes

- *2.7X Faster than 900 XK6 w/out Accelerators*

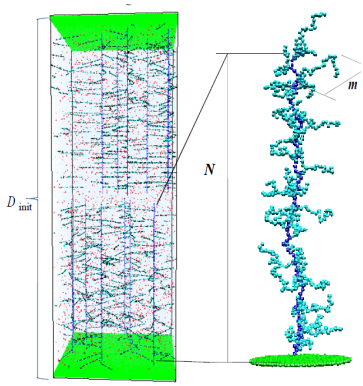
Kepler II vs Fermi +

- More cores
  - Efficiently handle larger problems, better performance for complex models
- Improved thread performance
  - Faster time to solution for same problem (increased clock, etc.)
- Larger register file
  - Better performance for complex models such as the liquid crystal case
- Warp shuffle
  - Reduced shared memory usage for reductions across threads
- Hyper-Q
  - MPI processes sharing the GPU can share a context
    - Reduced memory overhead per process
    - Concurrent kernel execution from multiple processes
      - Better strong-scaling performance (performance with smaller number of particles per GPU)

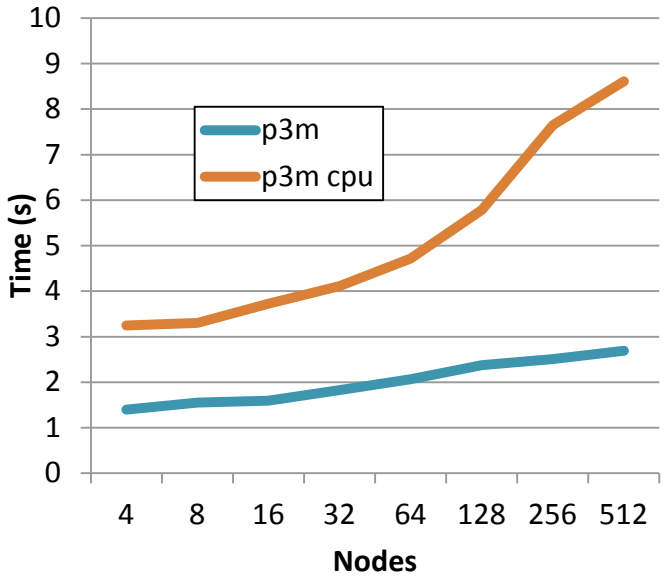
The “Porting Wall”

- Although you can get performance improvements by porting existing models/algorithms and running simulations using traditional parameterizations…
  - There is a limit to the amount of parallelism that can be exposed to decrease the time-to-solution
  - Increasingly desirable to re-evaluate computational physics methods and models with an eye towards approaches that allow for increased concurrency and data locality
    - Parameterize simulations to shift work towards routines well-suited for the accelerator
    - Methods/models with increased computational requirements can perform better if they can increase concurrency, allow for larger time-steps, etc.
- Computational scientists will play a critical role in exploiting the performance of current and next-generation architectures
- Some very basic examples…

Electrostatics Example with Polyelectrolyte Brushes



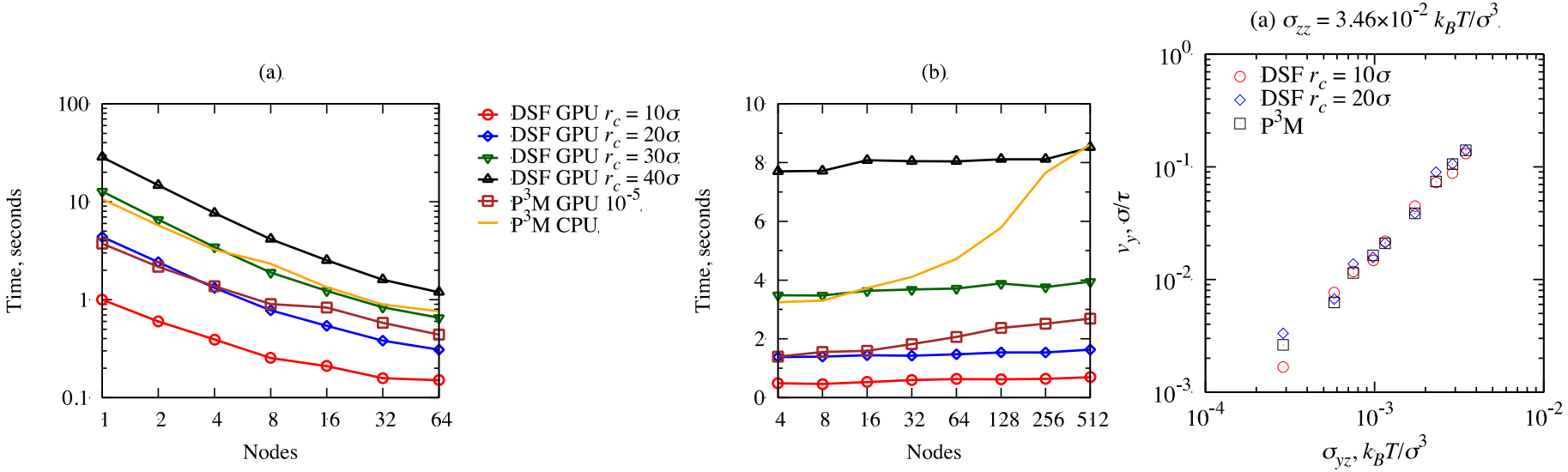
- Electrostatics are typically solved by splitting the Coulomb potential into a short-range potential that decays to zero at the cutoff and a long-range potential that converges rapidly in k-space
  - Long-range component is typically solved with discretization on a mesh
    - Poisson solve with 3D FFTs
    - Communications bottleneck
  - The traditional parameterizations that work well on older high-performance computers are not necessarily optimal for many simulations
  - Shift the work towards the short-range component
    - Disproportionate performance improvements on accelerator with larger short-range cutoffs
    - Benefits from acceleration **improve** at larger node counts!



Nguyen, T.D., Carrillo, J.-M., Dobrynin, A.V., Brown, W.M. **A Case Study of Truncated Electrostatics for Simulation of Polyelectrolyte Brushes on GPU Accelerators**. *To Appear.*

Electrostatics Example with Polyelectrolyte Brushes

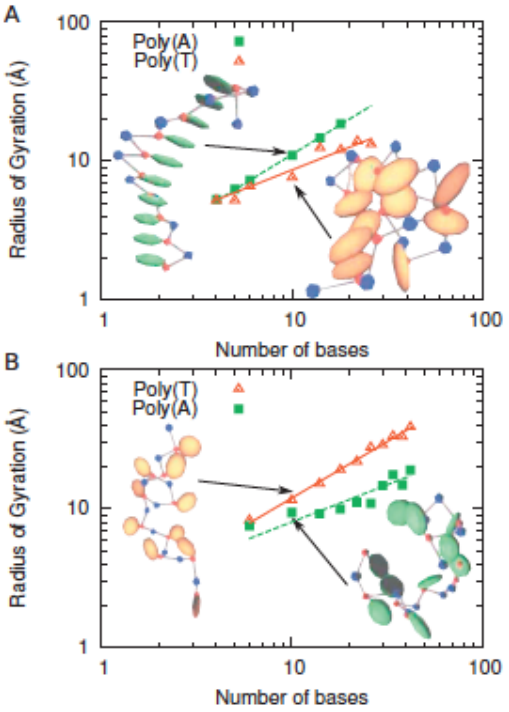
- Eliminate the long-range component?
  - In condensed phase systems, charges can be effectively short-range due to screening from other charged particles in the system
  - Use a large short range cutoff, correct for issues due to non-neutral spherical cutoff regions, and modify cell list calculation to be efficient for large cutoffs in parallel simulations
  - Accurate results with faster time-to-solution for some simulations



Nguyen, T.D., Carrillo, J.-M., Dobrynin, A.V., Brown, W.M. **A Case Study of Truncated Electrostatics for Simulation of Polyelectrolyte Brushes on GPU Accelerators**. *To Appear.*

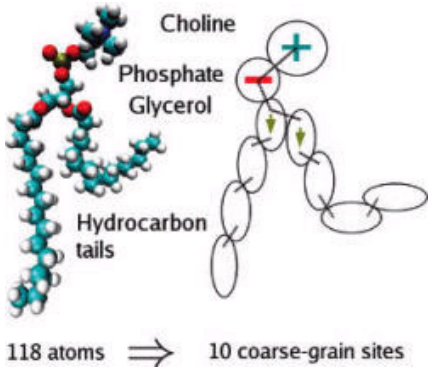
Alternative Potential Energy Models?

- Implementation of new models that can exploit high peak accelerator performance to improve accuracy and sampling
  - Computational cost of a single ellipsoid-ellipsoid interaction can be 15x that for Lennard-Jones on the CPU
  - With GPU acceleration, it is more competitive
    - Higher arithmetic intensity, so better speedup when compared to LJ acceleration
    - Better parallel efficiency *relative* to the CPU
      - Still get performance with fewer threads



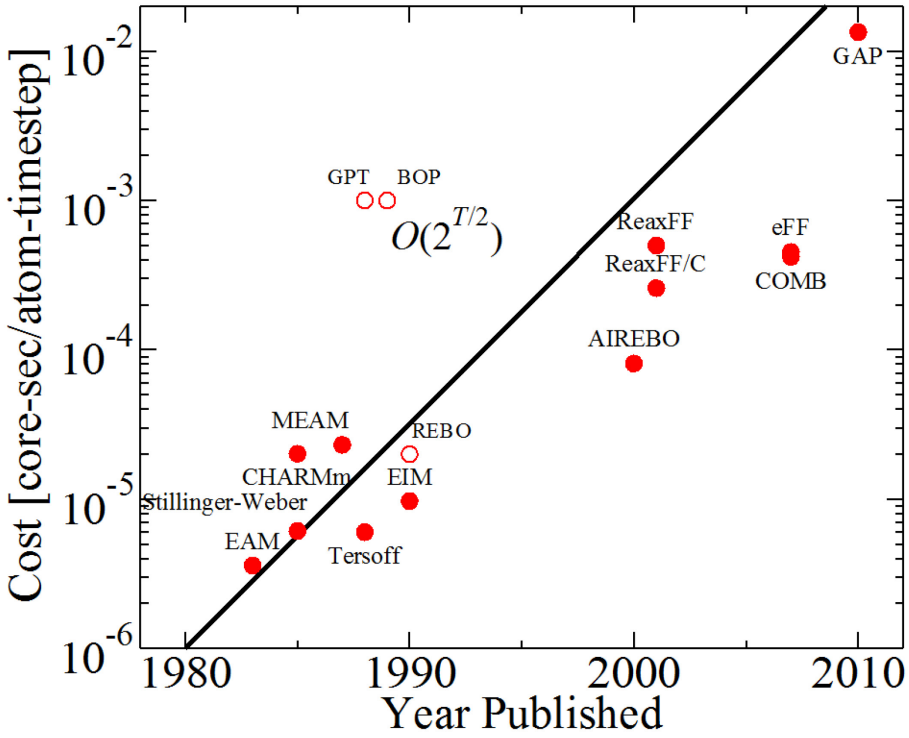
Morriss-Andrews, A., Rottler J., Plotkin S. S., JCP 132, 035105, 2010.

Mergell, B., Ejtehadi, M.R., Everaers, R., PRE, 68, 021911 (2003)



Orsi, M. et al, J. Phys. Chem. B, Vol. 112, No. 3, 2008

Moore's Law for Potentials



More creative examples…

- Advanced time integrators, time parallelism, etc…

Ongoing Work

- Implementation and evaluation of alternative algorithms for long-range electrostatics.
  - Multigrid(-like) Methods for Poisson Solve
    - O(N), no FFTs
- Implementation of complex models well suited for accelerators
- Improvements driven by specific science problems

Publications

- Brown, W.M., Wang, P. Plimpton, S.J., Tharrington, A.N. **Implementing Molecular Dynamics on Hybrid High Performance Computers – Short Range Forces.** *Computer Physics Communications.* 2011. 182: p. 898-911.
- Brown, W.M., Kohlmeyer, A., Plimpton, S.J., Tharrington, A.N. **Implementing Molecular Dynamics on Hybrid High Performance Computers – Particle-Particle Particle-Mesh.** *Computer Physics Communications.* 2012. 183: p. 449-459.
- Brown, W. M., Nguyen, T.D., Fuentes-Cabrera, M., Fowlkes, J. D., Rack, P. D., Berger, M., Bland, A. S. A**n Evaluation of Molecular Dynamics Performance on the Hybrid Cray XK6 Supercomputer**. *Proceedings of the ICCS Workshop on Large Scale Computational Physics*. 2012. Published in *Procedia Computer Science*, 2012. 9 p. 186-195.
- Nguyen, T.D., Carrillo, J.-M., Dobrynin, A.V., Brown, W.M. **A Case Study of Truncated Electrostatics for Simulation of Polyelectrolyte Brushes on GPU Accelerators**. *To Appear.*

Science Applications

- Carrillo, J.-M., Brown, W.M., Dobrynin, A.V. **Explicit Solvent Simulations of Friction between Brush Layers of Charged and Neutral Bottle-Brush Macromolecules**. *Submitted.*
- Nguyen TD, Fuentes-Cabrera M, Fowlkes JD, Diez JA, Gonzalez AG, Kondic L, Rack PD, **Competition between collapse and breakup in nanometer-sized thin rings using molecular dynamics and continuum modeling**, *Langmuir* 28, 13960-13967, 2012.

Acknowledgements

- LAMMPS
  - Steve Plimpton (SNL) and many others
- LAMMPS Accelerator Library
  - W. Michael Brown (ORNL), Trung Dac Nguyen (ORNL), Peng Wang (NVIDIA), Axel Kohlmeyer (Temple), Steve Plimpton (SNL), Inderaj Bains (NVIDIA)
- Geryon Library
  - W. Michael Brown (ORNL), Manuel Rodriguez Rodriguez (ORNL), Axel Kohlmeyer (Temple)
- K-Space Partitions
  - Yuxing Peng and Chris Knight (University of Chicago)
- Metallic Nanoparticle Dewetting
  - Miguel Fuentes-Cabrera (ORNL), Trung Dac Nguyen (ORNL), W. Michael Brown (ORNL), Jason Fowlkes (ORNL), Philip Rack (UT, ORNL)
- Lipid Vesicle Science and Simulations
  - Vincenzo Carnevale (Temple), Axel Kohlmeyer (Temple), Michael Klein (Temple), et. al.
- Enhanced Truncation/Bottlebrush Simulations
  - Trung Dac Nguyen (ORNL), Jan-Michael Carrillo (UC), Andrew Dobrynin (UC), W. Michael Brown (ORNL)
- Multilevel Summation
  - Arnold Tharrington (ORNL)