Harnessing GPU speed to accelerate LAMMPS particle simulations

Paul S. Crozier, W. Michael Brown, Peng Wang

pscrozi@sandia.gov, wmbrown@sandia.gov, penwang@nvidia.com

SC09, Portland, Oregon

November 18, 2009

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

The GPU-LAMMPS team

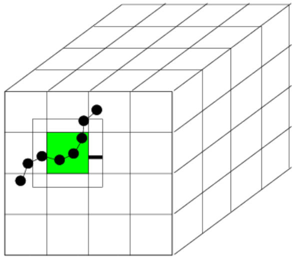
Marc Adams (Nvidia)
Pratul Agarwal (ORNL)
Sarah Anderson (Cray)
Mike Brown (Sandia)
Paul Crozier (Sandia)
Massimiliano Fatica (Nvidia)
Scott Hampton (ORNL)
Ricky Kendall (ORNL)
Hyesoon Kim (Ga Tech)
Axel Kohlmeyer (Temple)
Doug Kothe (ORNL)

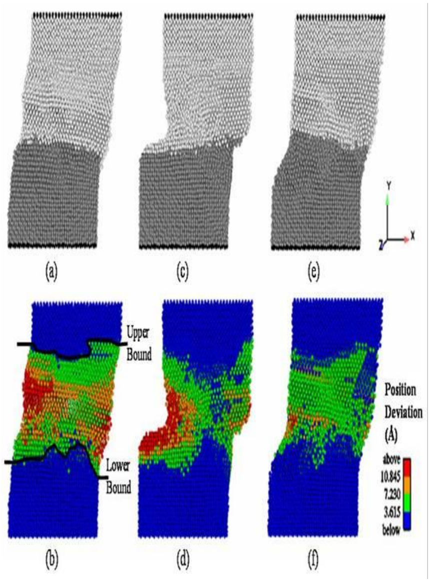
Scott LeGrand (Nvidia)
Ben Levine (Temple)
Steve Plimpton (Sandia)
Duncan Poole (Nvidia)
Steve Poole (ORNL)
Jason Sanchez (RPI)
Arnold Tharrington (ORNL)
John Turner (ORNL)
Peng Wang (Nvidia)
Andrew Zonenberg (RPI)

LAMMPS

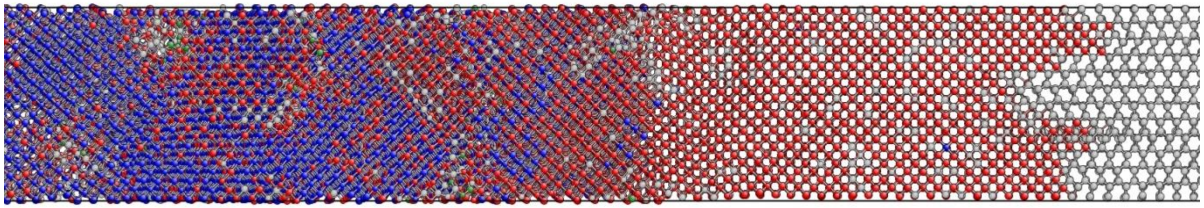
(Large-scale Atomic/Molecular Massively Parallel Simulator)

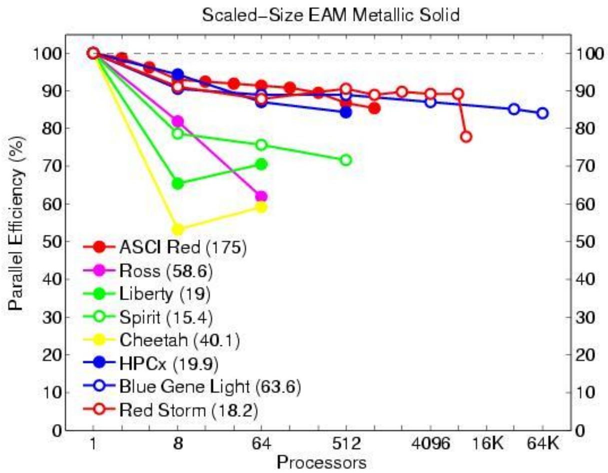
http://lammps.sandia.gov





- Classical MD code.
- Open source, highly portable C++.
- Freely available for download under GPL.
- Easy to download, install, and run.
- Well documented.
- Easy to modify or extend with new features and functionality.
- Active user’s e-mail list with over 300 subscribers.
- Since Sept. 2004: over 20k downloads, grown from 53 to 125 kloc.
- Spatial-decomposition of simulation domain for parallelism.
- Energy minimization via conjugate-gradient relaxation.
- Radiation damage and two temperature model (TTM) simulations.
- Atomistic, mesoscale, and coarse-grain simulations.
- Variety of potentials (including many-body and coarse-grain).
- Variety of boundary conditions, constraints, etc.





Why enable LAMMPS to run on GPUs?

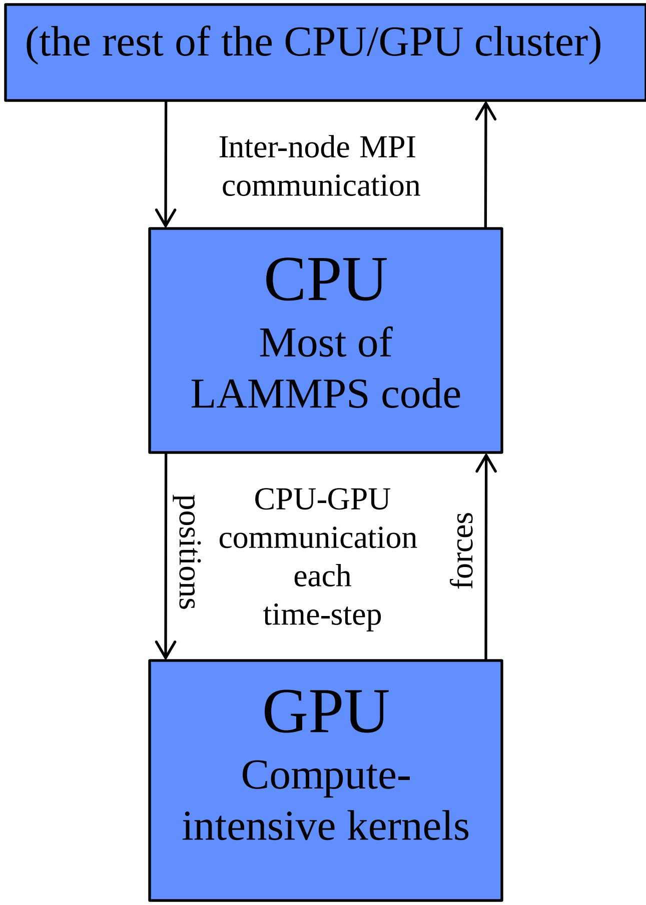
- **Other MD codes are running on GPUs and showing big speed-ups.**

- **The future is many-core, and GPU computing is leading the way.**

- **Next generation supercomputers will have accelerators, like LANL’s Road Runner, but the accelerators will be GPUs in many cases.**

GPU-LAMMPS strategy

- **Enable LAMMPS to run efficiently on future CPU-based clusters that have GPU accelerators.**
- **Not aiming for running on a single GPU.**
- **Not aiming to rewrite all of LAMMPS in CUDA.**
- **Rewrite the most compute-intensive LAMMPS kernels in CUDA.**
- **At each time-step, ship particle positions from CPU to GPU, compute forces on the GPU, and then ship forces back to the CPU.**



(the rest of the CPU/GPU cluster)
Inter-node MPI communication
CPU
Most of LAMMPS code
positions
CPU-GPU communication each time-step
forces
GPU
Compute-intensive kernels

Current status of the GPU-LAMMPS project

- GPU library and package available for download with LAMMPS: http://lammps.sandia.gov
- CUDA kernels written for two pairs styles: LJ and Gay-Berne
- Pre-release developer's version of GPU-LAMMPS available for subversion checkout at: http://code.google.com/p/gpulammps/
  - Faster CUDA kernel for LJ
  - Cell list built on the GPU to avoid reneighboring on the CPU
  - More LAMMPS pair styles coming soon
- To offer feedback, or join the GPU-LAMMPS team, send e-mail to: Paul Crozier (pscrozi@sandia.gov)

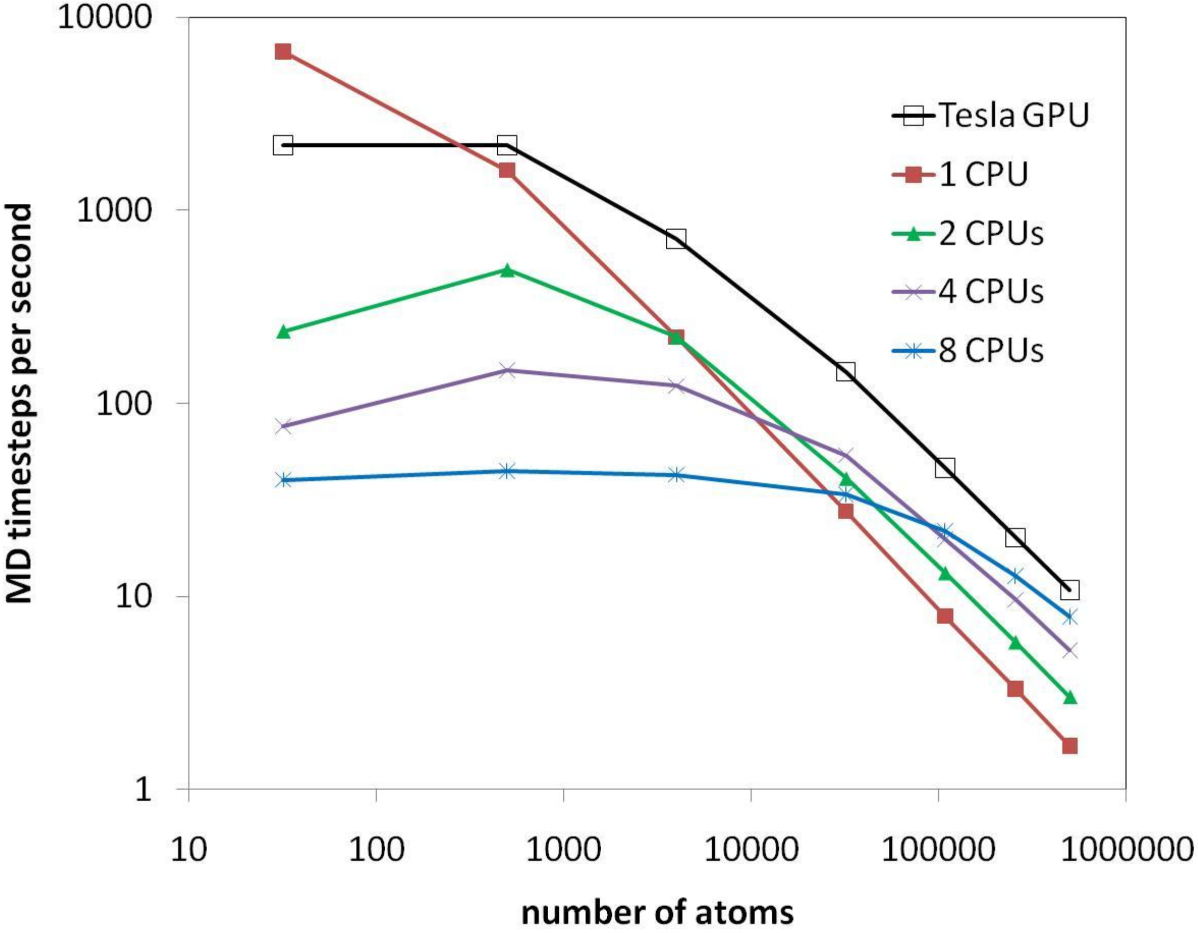
Results for LAMMPS LJ benchmark

- Dual Quad Core Intel® Xeon® Processors X5560 2.8GH

- 1 Tesla C1060 GPU

- Use of Tesla is almost always faster than not using it on this machine.

- Tesla 3.2x faster than Dual Quad Core for 4000 atom system.



Gay-Berne Potential for Liquid Crystal Simulation

- **Gay-Berne for dissimilar biaxial ellipsoids**
- **S is the shape matrix, A rotation, *h*$_{12}$ distance of closest approach**
- **The E matrix characterizes the relative well depths of side-to-side, face-to-face, and end-to end interactions**
- **~30 times the cost of an LJ interaction**

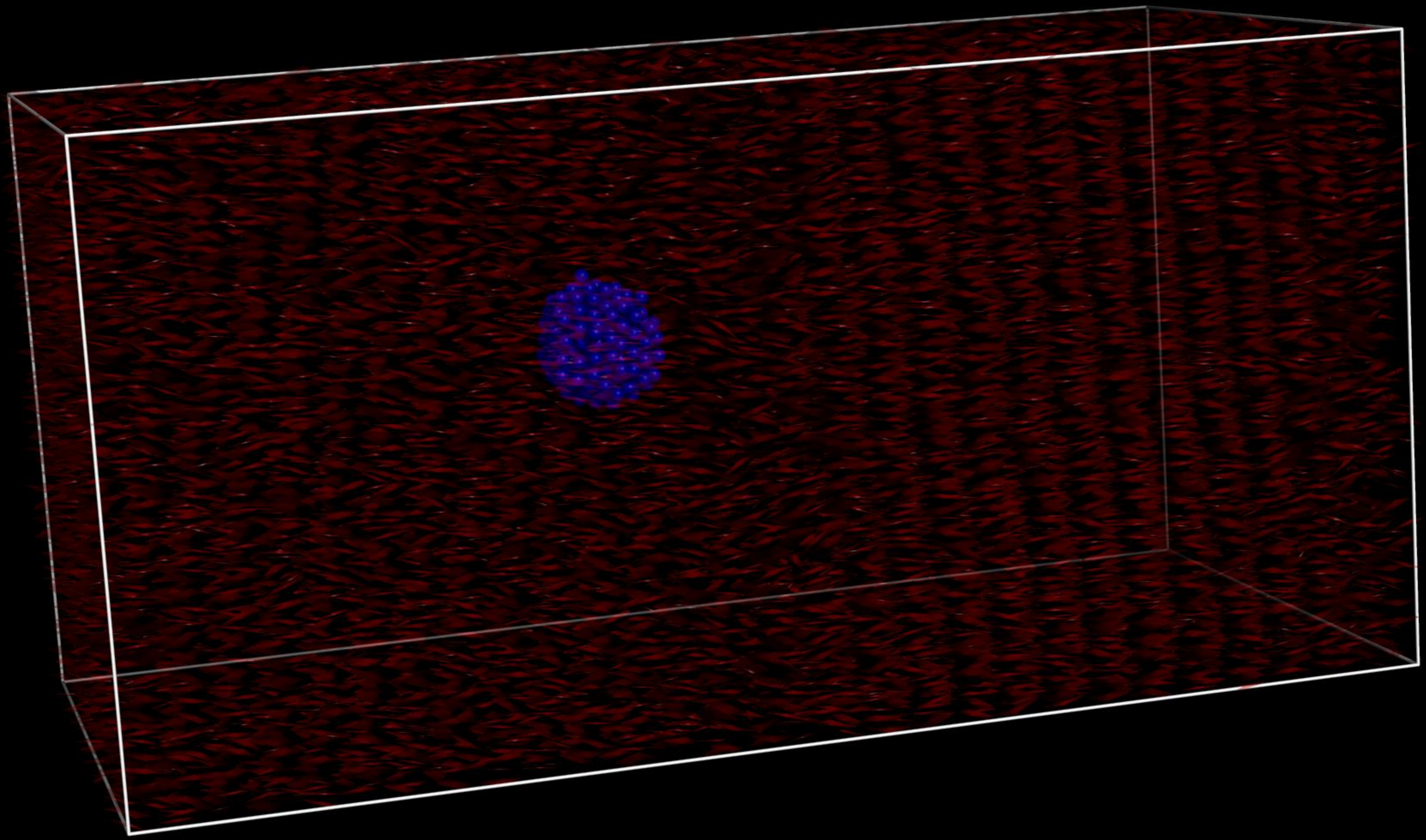
$$U = U_r\left(\mathbf{A}_1,\mathbf{A}_2,\mathbf{r}_{12}\right)\eta_{12}\left(\mathbf{A}_1,\mathbf{A}_2\right)\chi_{12}\left(\mathbf{A}_1,\mathbf{A}_2,\hat{r}_{12}\right)$$

$$U_r = 4\varepsilon\left[\left(\frac{\sigma}{h_{12}+\gamma\sigma}\right)^{12} - \left(\frac{\sigma}{h_{12}+\gamma\sigma}\right)^{6}\right]$$

$$\eta_{12} = \left[\frac{2s_1s_2}{\det\left[\mathbf{A}_1^T\mathbf{S}_1^2\mathbf{A}_1 + \mathbf{A}_2^T\mathbf{S}_2^2\mathbf{A}_2\right]}\right]^{\upsilon/2}$$

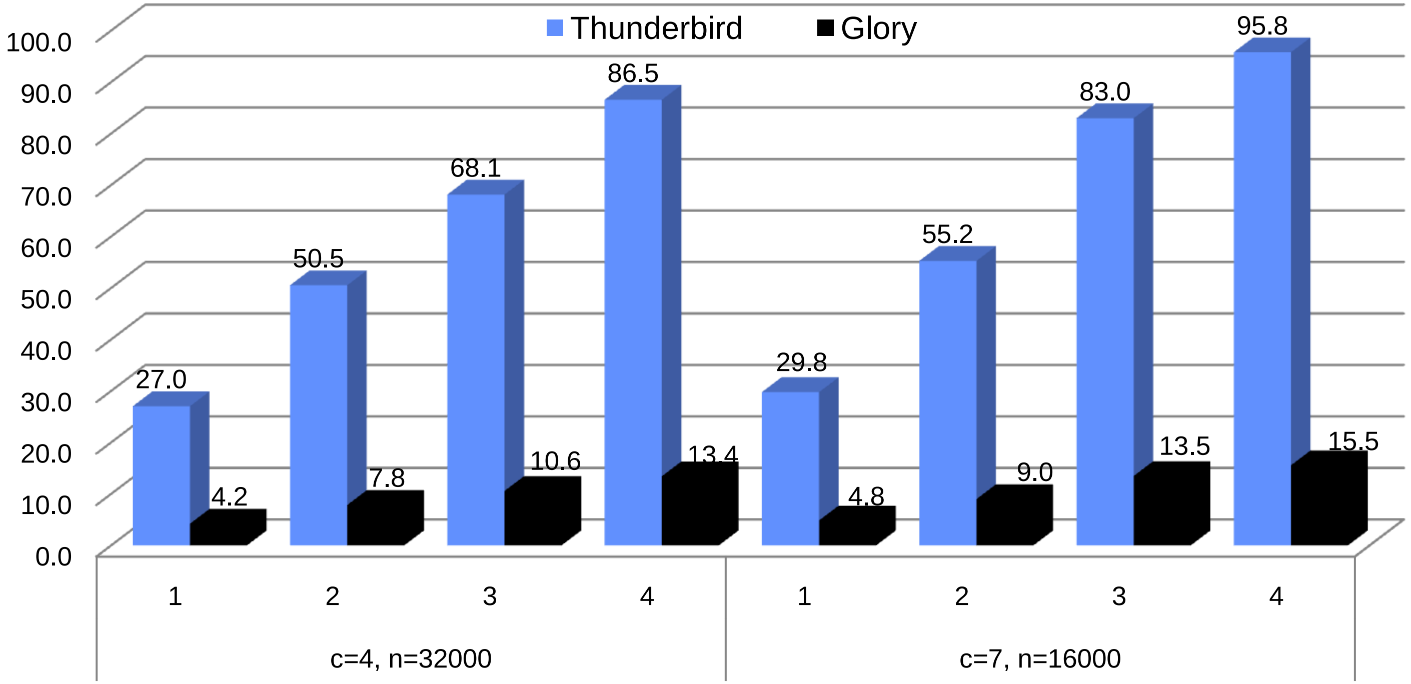
$$s = \left[a_ib_i + c_ic_i\right]\left[a_ib_i\right]^{1/2}$$

$$\chi_{12} = \left[2\hat{\mathbf{r}}_{12}^T\left(\mathbf{A}_1^T\mathbf{E}_1\mathbf{A}_1 + \mathbf{A}_2^T\mathbf{E}_2\mathbf{A}_2\right)^{-1}\hat{\mathbf{r}}_{12}\right]^{u}$$



GPU Times Speedup vs 1 Node

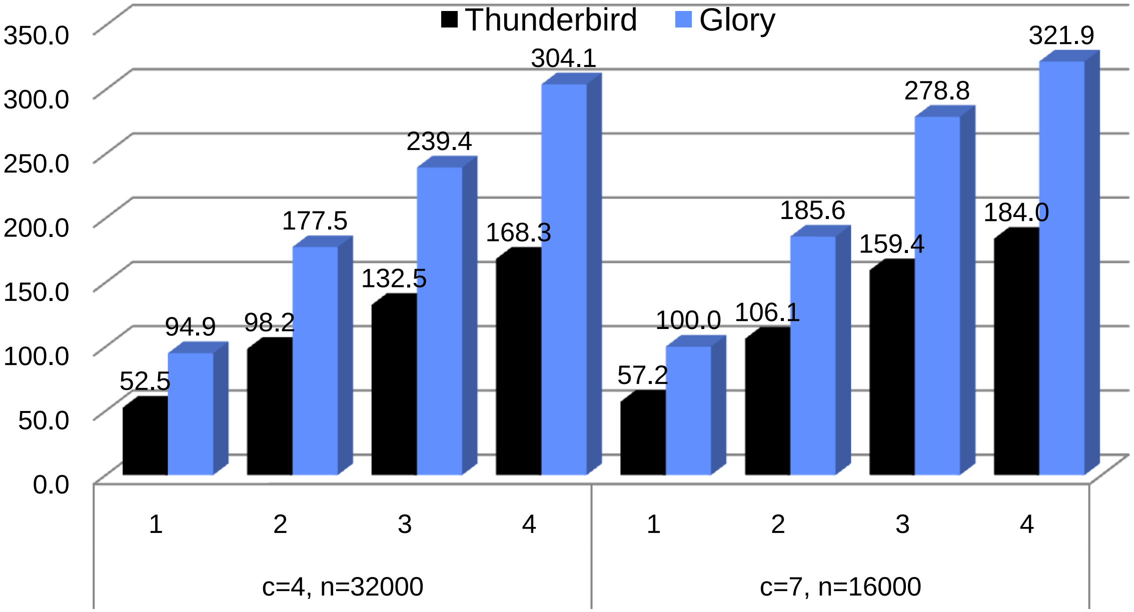
(*c*=cutoff, *n*=particles)



GPU: 1, 2, 3, or NVIDIA, 240 core, 1.3 GHz GPUs
Thunderbird: *2 procs, Dual 3.6 GHz Intel EM64T processors*
Glory: *16 procs, Quad Socket/Quad Core 2.2 GHz AMD*

GPU Times Speedup vs 1 Core

(*c*=cutoff, *n*=particles)



GPU: 1, 2, 3, or 4 NVIDIA, 240 core, 1.3 GHz GPUs
Thunderbird: *1 core of Dual 3.6 GHz Intel EM64T processors*
Glory: *1 core of Quad Socket/Quad Core 2.2 GHz AMD*

Plans for further GPU-LAMMPS work

Before SC10, we plan to:

- Focus on fast biomolecular simulations on CPU/GPU hybrid supercomputers.
- Do efficient 3D FFTs on CPU/GPU hybrid cluster for faster long-range electrostatics.
- Improve CUDA kernel efficiency of LJ and GB pair styles.
- Add more CUDA kernels for additional LAMMPS pair styles until most of LAMMPS's 40 or so pair styles are done.

Sandia
National
Laboratories