SAND2013-6383C

LAMMPS implementation of the Multilevel Summation Method (MSM) for long-range electrostatics calculations

Stan Moore and Paul Crozier

August 7, 2013

LAMMPS Users' Workshop and Symposium

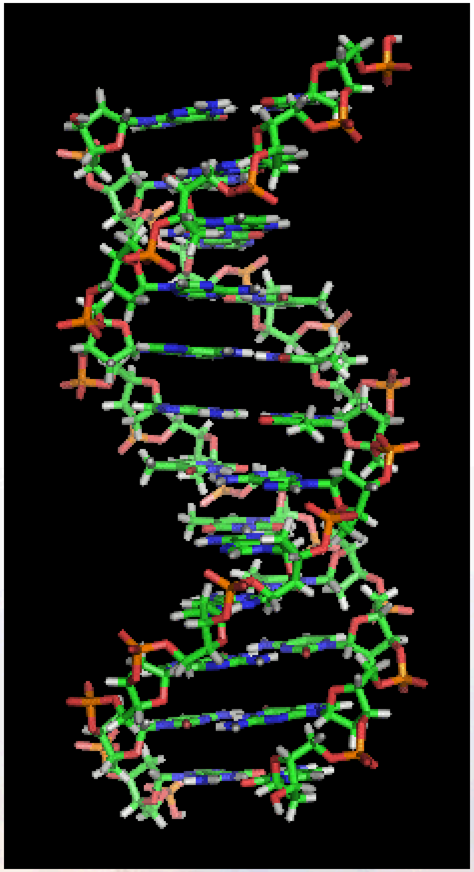
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Why do we care about LRE?

LRE = long-range electrostatics

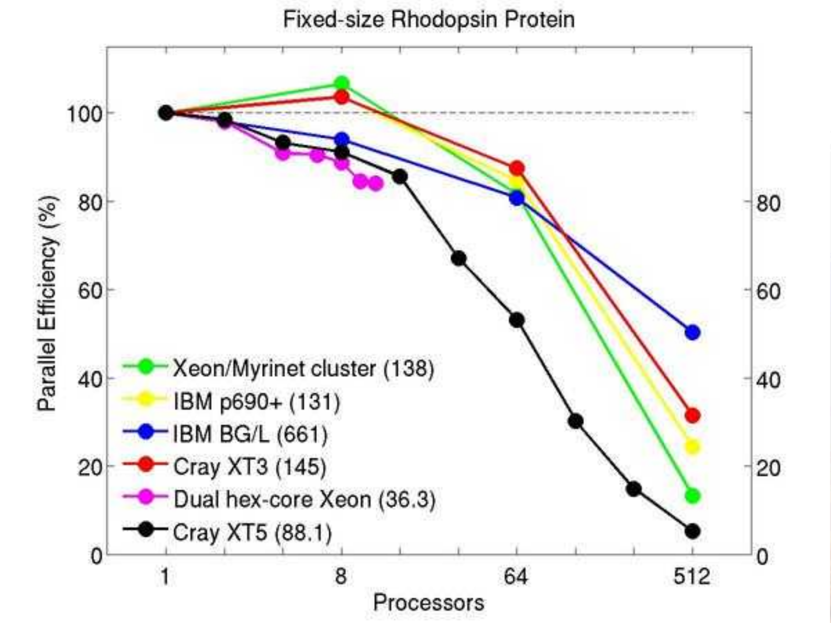
- Electrostatics are extremely important in many atom-level (and coarser) models.
- Long-range part usually cannot be neglected in molecular simulations (simple cutoff can lead to artifacts).
- Large fraction of compute cycles are used in LRE calculations.
- LRE calculations represent a scaling bottleneck in many MD calculations.



The motivation: FFTs don't scale very well

(and HPC core counts are growing quickly)

- LAMMPS originally had two methods for computing long-range electrostatics: Ewald and particle-particle/particle-mesh (PPPM)
- Ewald summation is fastest for small systems (or very high accuracy), but expensive for large systems
- PPPM relies on FFTs, which don't scale well on many processors:



http://lammps.sandia.gov/bench/rhodo.fixed.jpg

Multilevel-summation method (MSM) background

- **Multi-grid method (but not iterative); split potential and approximate the slowly varying part on a hierarchy of grids**
- **No FFTs are required, so the communication cost of MSM is expected to scale better than PPPM on large core counts**
- **PPPM scales with number of atoms as O(N log N) while MSM scales as O(N)**
- **MSM may be faster for large problems running on large core counts**

D. J. Hardy, Ph.D. thesis, University of Illinois at Urbana-Champaign (2006).

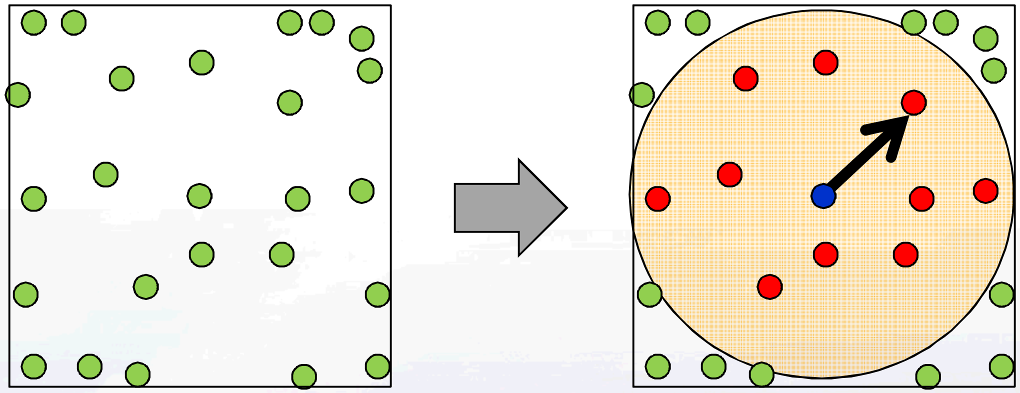
D. J. Hardy, J. E. Stone, and K. Schulten, Parallel Comput. **35**, 164 (2009).

MSM Algorithm pieces

1. Short-range part
2. Anterpolation
3. Direct sum
4. Restriction
5. Prolongation
6. Interpolation

Algorithm pieces

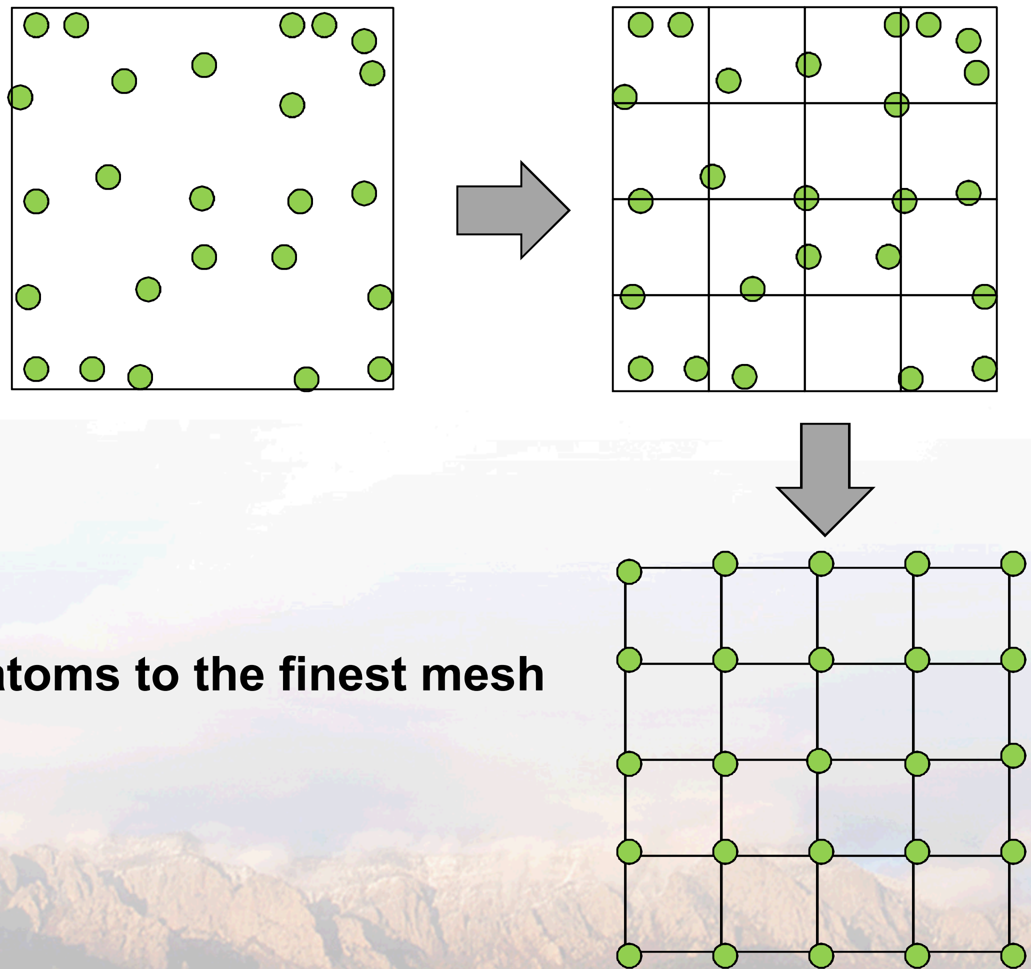
1. **Short-range part**
2. Anterpolation
3. Direct sum
4. Restriction
5. Prolongation
6. Interpolation



Particles interact within a spherical cutoff to get short-range forces, energy, and pressure

Algorithm pieces

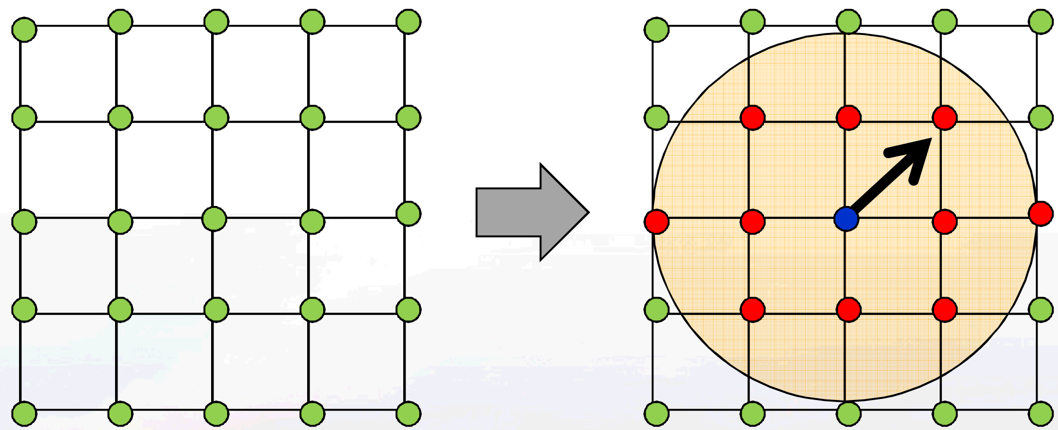
1. Short-range part
2. **Anterpolation**
3. Direct sum
4. Restriction
5. Prolongation
6. Interpolation



Interpolate charges from atoms to the finest mesh

Algorithm pieces

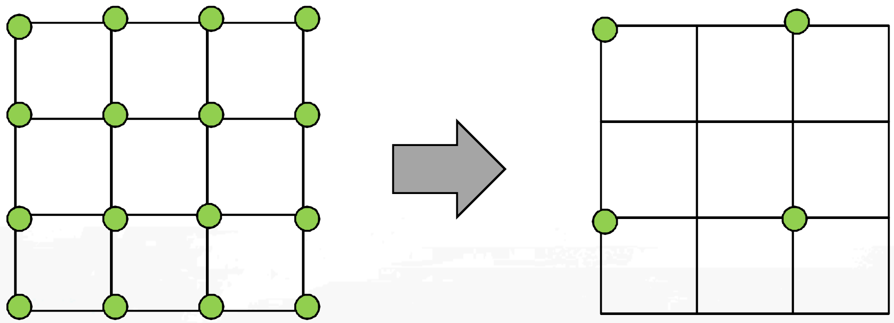
1. Short-range part
2. Anterpolation
3. **Direct sum**
4. Restriction
5. Prolongation
6. Interpolation



Mesh points interact within a spherical cutoff to get long-range electric field, energy, and pressure

Algorithm pieces

1. Short-range part
2. Anterpolation
3. Direct sum
4. **Restriction**
5. Prolongation
6. Interpolation

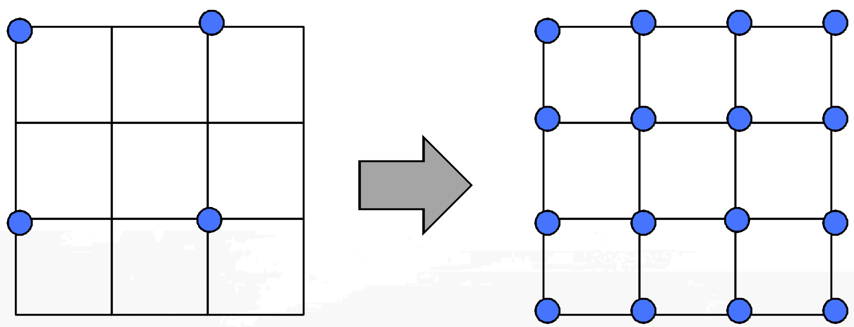


Interpolate charge from finer mesh to coarser mesh

Repeat steps 3 and 4 until finished on coarsest mesh

Algorithm pieces

1. Short-range part
2. Anterpolation
3. Direct sum
4. Restriction
5. **Prolongation**
6. Interpolation

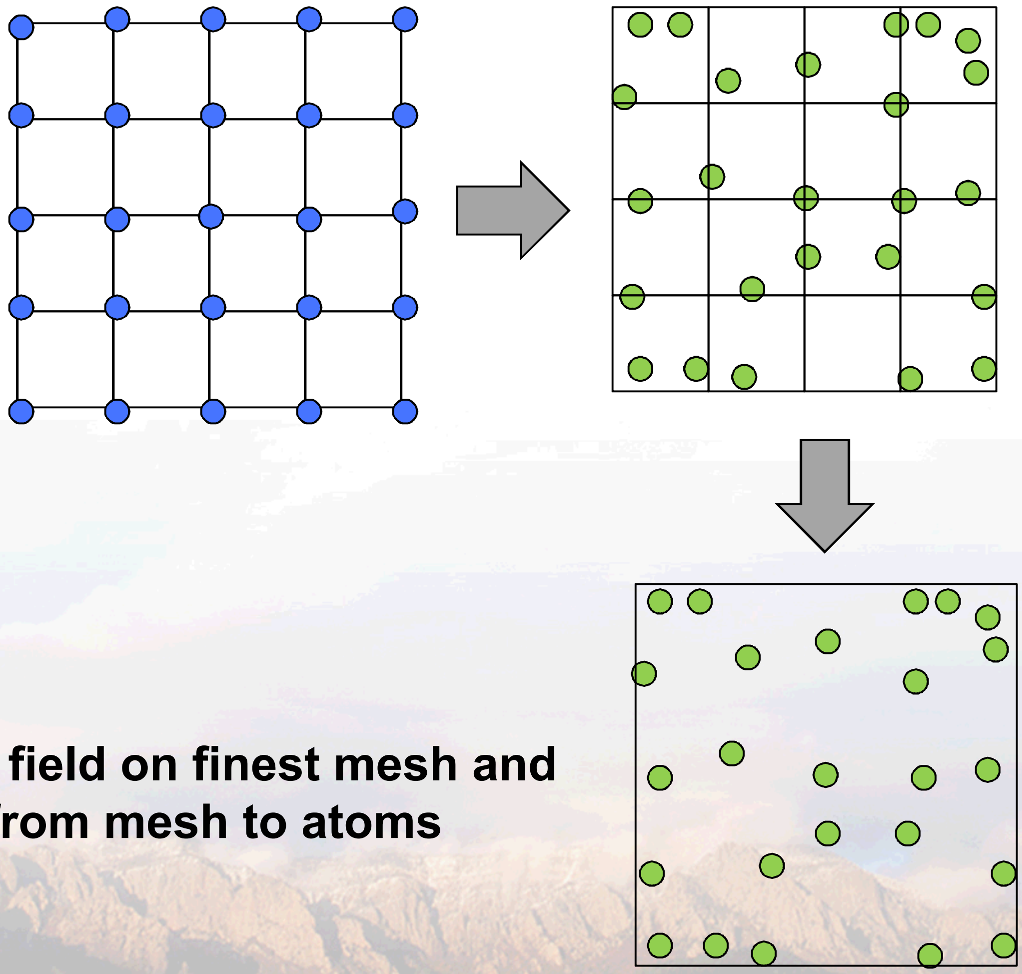


Interpolate electric field from coarser mesh to finer mesh

Repeat step 5 until the finest mesh is reached

Algorithm pieces

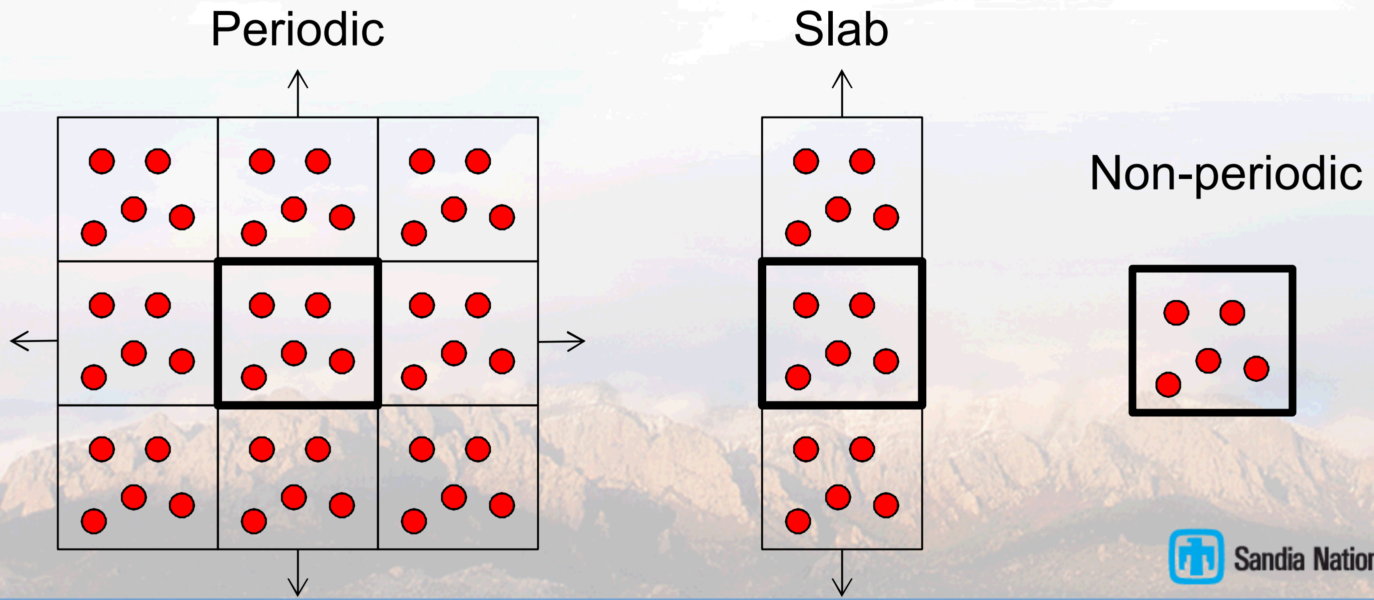
1. Short-range part
2. Anterpolation
3. Direct sum
4. Restriction
5. Prolongation
6. **Interpolation**



Compute force from electric field on finest mesh and back-interpolate force from mesh to atoms

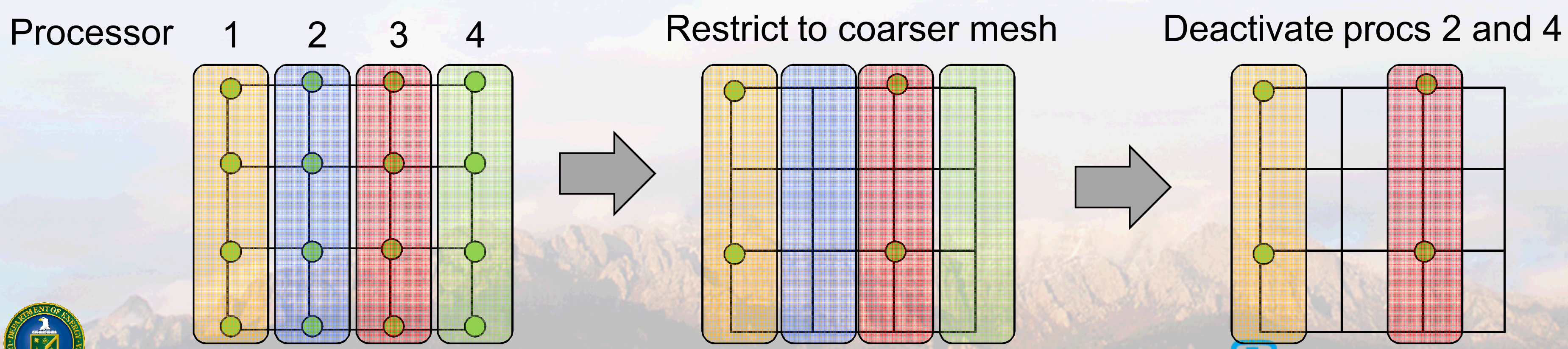
Non-Periodic BCs

- **MSM works for any combination of periodic, non-periodic, or shrink-wrapped boundary conditions**
- **Ewald and PPPM only work for periodic or slab (periodic in *x* and *y* and non-periodic in *z*) boundary conditions**



Parallelization Strategy

- Challenge: lots of work on finest grid, very little work on coarsest grid
- Use same domain-decomposition layout on all levels (simple)
- Inactive processors don’t participate in MPI communication routines
- Use neighbor point-to-point communication for fine grids
- Use MPI AllReduce for coarse grids

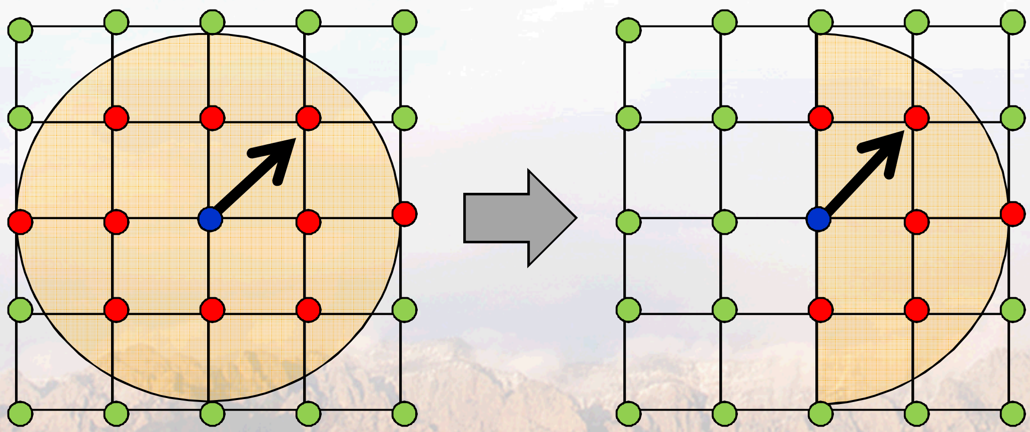


Enhancements to MSM

- Error estimator (important for comparing to other methods like PPPM)
- Pressure calculation
- Added heuristic to estimate optimal parameters, including automatic adjustment of Coulombic cutoff (based on work by Hardy)
- Per-atom energy/virial
- Fast scalar pressure
- OMP threaded version of MSM (Axel Kohlmeyer)

Improving Single-Core Performance

- Use hemisphere (instead of full sphere) for direct sum interactions to avoid double computations
- Using a hemisphere can also (sometimes) reduce the amount of communication needed
- Added various other code optimizations
- Compared to Hardy's NAMD-lite code, LAMMPS MSM was 60% faster for periodic and 25% faster for non-periodic (1 processor, two point-charges in a box, order 4)



[David J. Hardy, *NAMD-Lite*, http://www.ks.uiuc.edu/Development/MDTools/namdlite/, University of Illinois at Urbana-Champaign, 2007.]

Fast (Scalar) Pressure Calculation

- Calculation of the 6-component pressure tensor is expensive with MSM (increases cost by ~2x)
- Often only scalar pressure [i.e. 1/3*(P_{xx} + P_{yy} + P_{zz})] is needed
- For Coulombic systems, can use a virial “trick” to relate energy to scalar pressure (much cheaper)
- For SPC/E system, reduces overall cost by 20% (short-range part has some overhead)
- Can use scalar pressure to run isotropic barostat

*Should be released soon

LRE speed and scalability tests

Chama

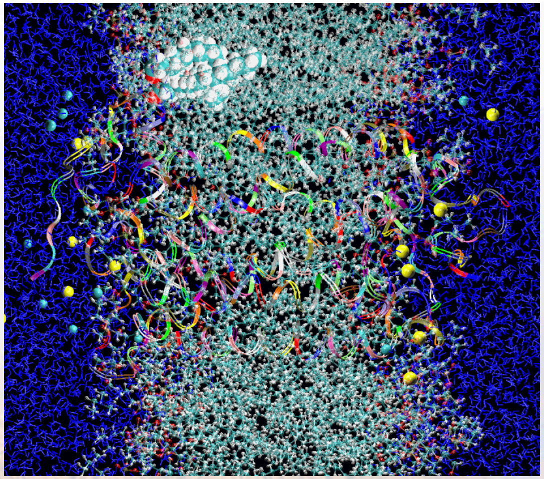
- Sandy-Bridge (2.6 GHz)
- 2012
- 16 cores/node
- 1,232 nodes
- 19,712 cores
- Infiniband 4X QDR, Fat Tree, Qlogic

Redsky

- Nehalem (2.93 GHz)
- 2009
- 8 cores/node
- 2,816 nodes
- 22,528 cores
- Infiniband 4X QDR, 3D Torus, Mellanox

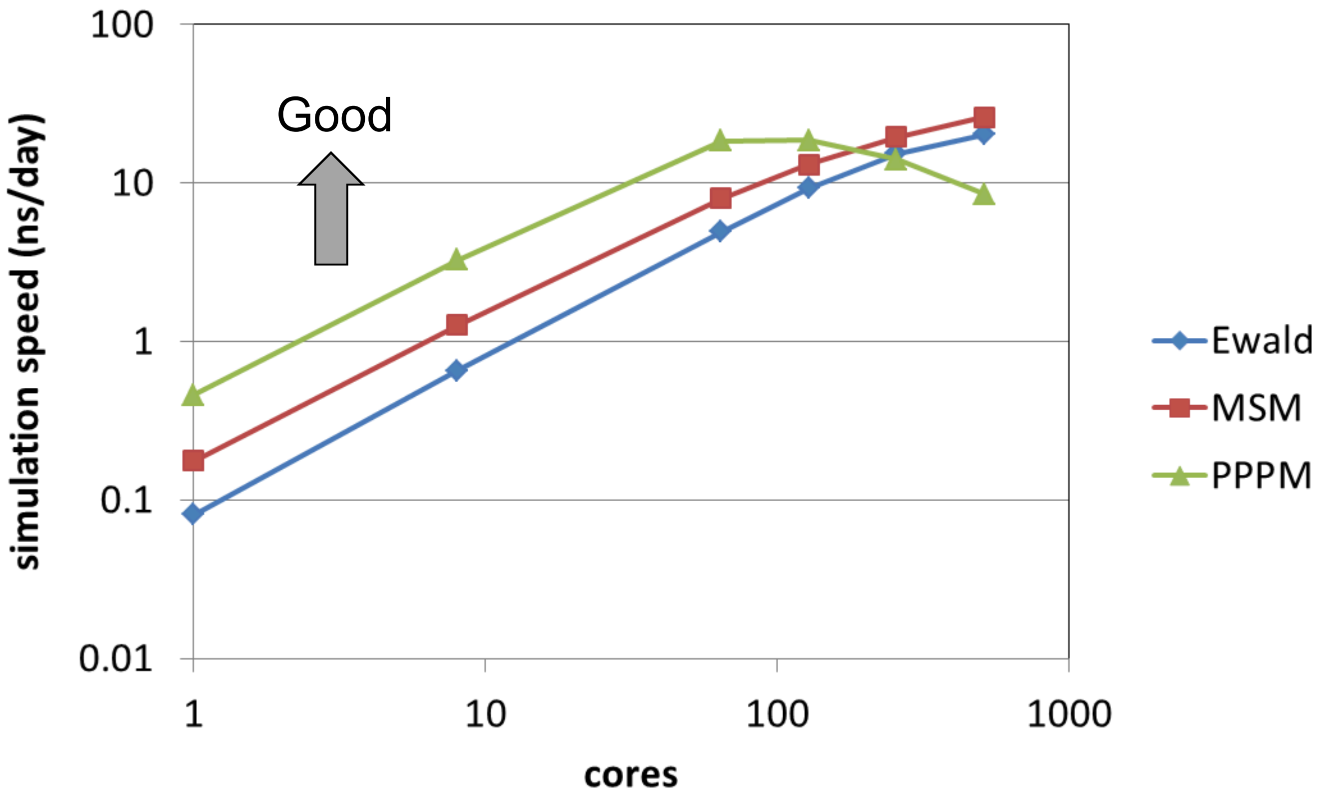
Rhodopsin benchmark

- NVT dynamics
- 1e-4 accuracy
- 32k atoms, replicable
- 2 fs timestep size



Redsky, 32k atoms, 10^{-4} accuracy

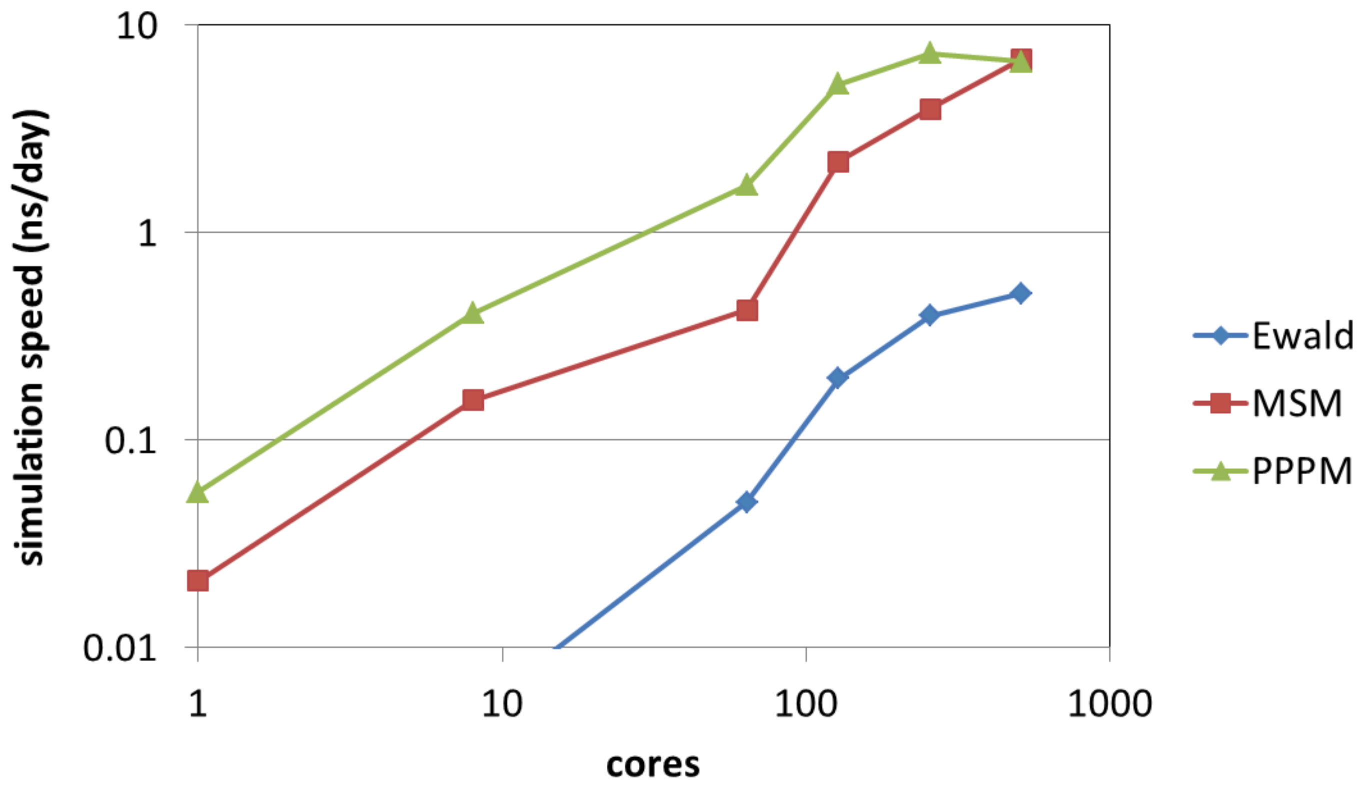
- **PPPM is fastest at low core count**
- **MSM is fastest at high core count**
- **MSM scales better than PPPM since it doesn't rely on FFTs**
- **Maxes out at 26 ns/day**



10 Angstrom cutoff, default parameters, MSM order 10

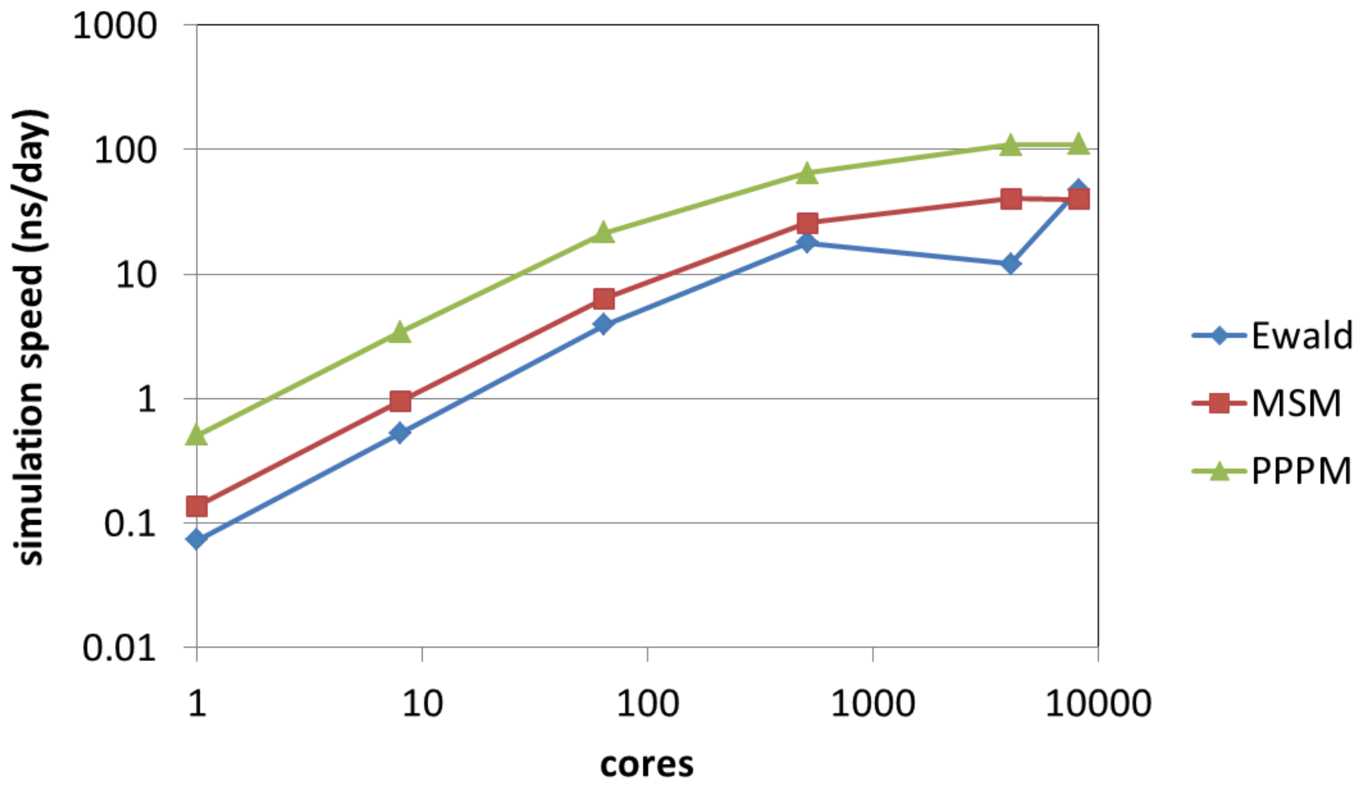
Redsky, 256k atoms, 10^{-4} accuracy

- 8x as many atoms as before
- Ewald chokes
- PPPM FFT bottleneck pushed out to larger core count
- Scaling: MSM O(N), PPPM O(N log(N)), Ewald $O(N^{1.5})$



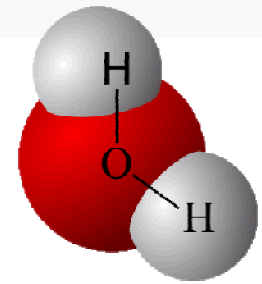
Chama, 32k atoms, 10^{-4} accuracy

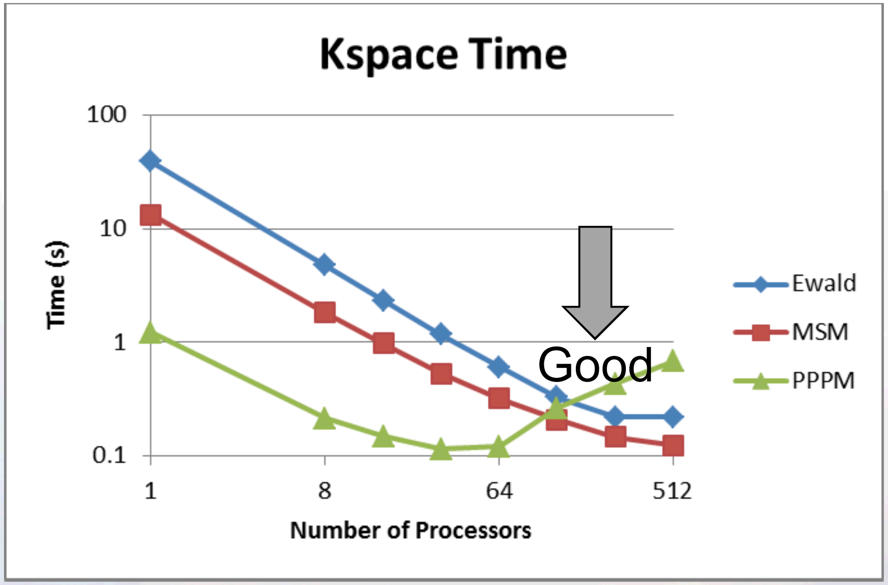
- Chama gives better overall performance
- PPPM wins, but we can't expect more scaling
- Maxes at 110 ns/day

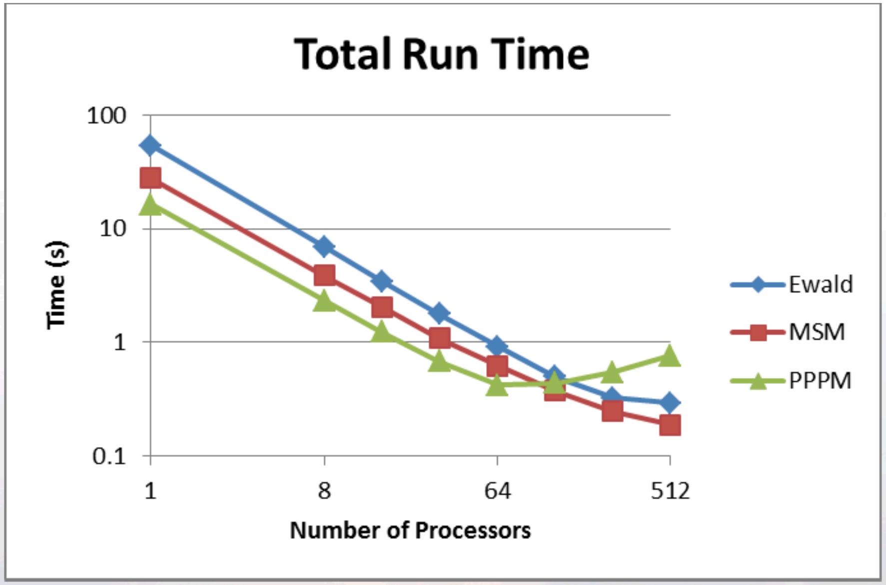


SPC/E Water Benchmark

- 36,000 atoms (strong scaling on Redsky)
- NVT, pressure computed every 50 timesteps
- 1e-3 accuracy

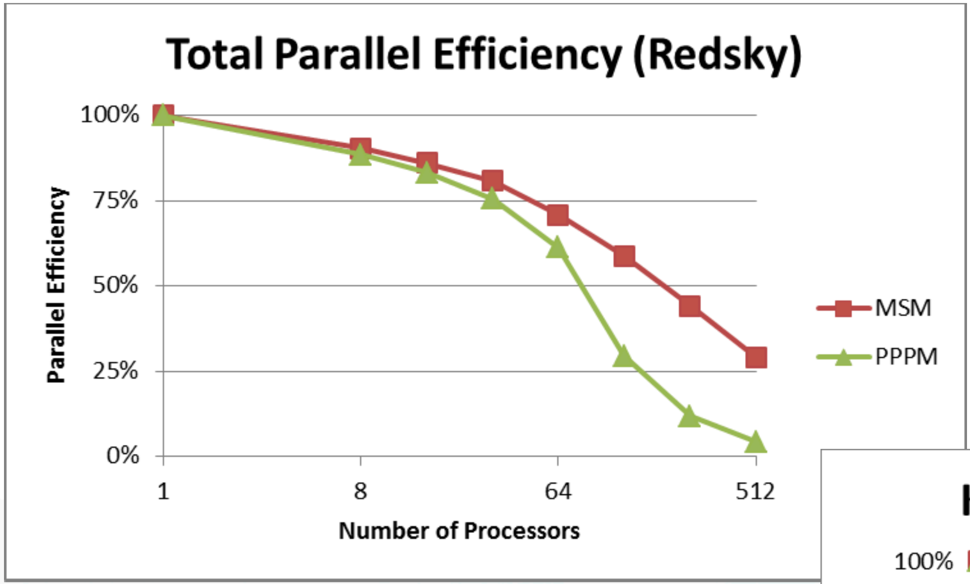


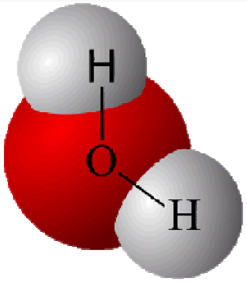


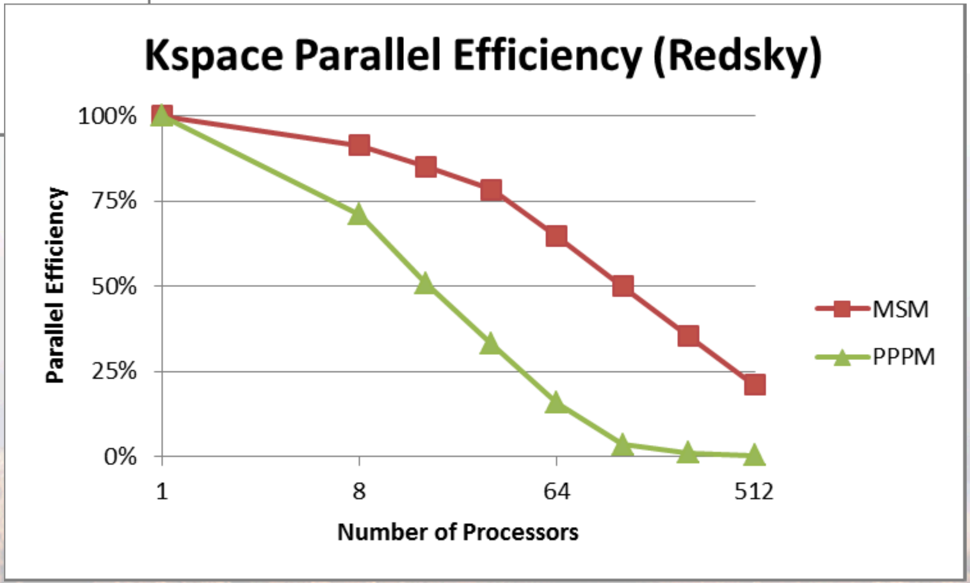


10 Angstrom cutoff, default parameters, MSM order 8

SPC/E Water Benchmark







Other Enhancements to LRE in LAMMPS

- Per-atom energy/virial for kspace
- Compute group/group for PPPM and Ewald
- Triclinic for kspace
- Ewald/disp for point-dipoles
- Staggered PPPM (up to 4x faster for high accuracy)
- Fix tune/kspace

Good for PPPM on large core counts:

- Fix verlet/split
- 2 FFT PPPM

- Additional requests?

Automatically adjust LRE parameters for optimal performance

- There are now a **myriad of LRE styles/parameters** in LAMMPS to choose from: style (Ewald, PPPM, or MSM), cutoff, grid points in each direction, order, gewald, cores, other

- New “fix tune/kspace” option helps users **find the best LRE parameters**.

- **Tailored specifically** to user’s machine, MPI ranks, use of threading or accelerators, simulated system, and simulation details.

- Still being developed (by Paul Crozier). Try it out and report any bugs or suggestions. ☺



Conclusions

- **MSM works well for:**
  - large core counts where all-to-all communication is expensive
  - non-periodic BCs
  - lower accuracy (1e-4 and below)
  - large numbers of atoms (at least in theory due to better O(N) scaling)
- **Ewald works well for small systems**
- **PPPM is good for many practical systems and is hard to beat!**

Thank You

Questions or Comments?