

Molecular Simulation of Reactive Systems.

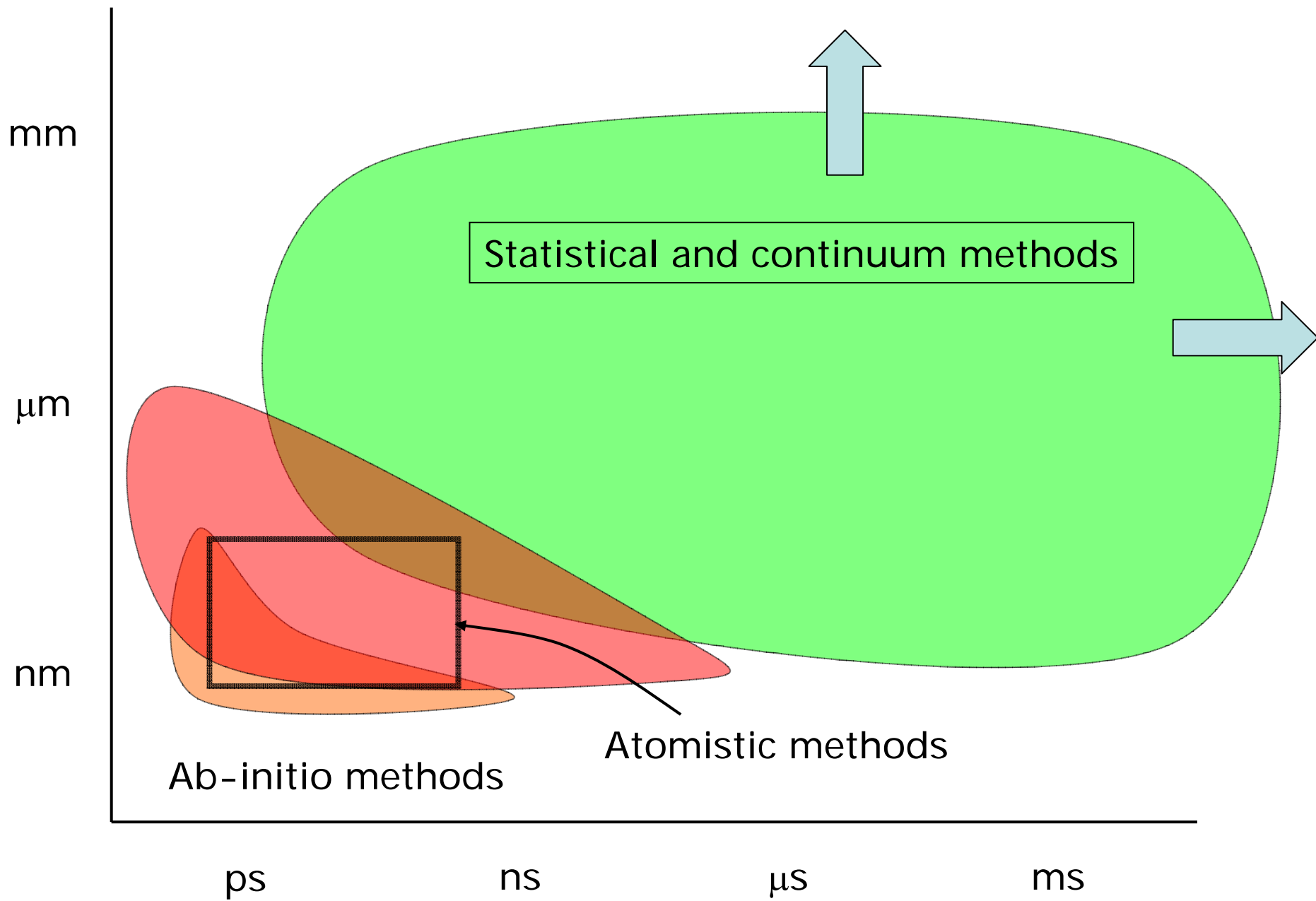
Sagar Pandit, Hasan Aktulga, Ananth Grama
Coordinated Systems Lab
Purdue University
ayg@cs.purdue.edu

--

Supported by the National Science Foundation and National Institutes of Health.

Molecular Simulation Methods

- Ab-initio methods (few approximations but slow)
 - DFT
 - CPMD
 - Electron and nuclei treated explicitly
- Classical atomistic methods (more approximations)
 - Classical molecular dynamics
 - Monte Carlo
 - Brownian dynamics
 - No electronic degrees of freedom. Electrons are approximated through fixed partial charges on atoms.
- Continuum methods (no atomistic details)

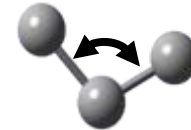


Simplified Interactions Used in Classical Simulations

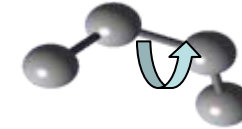
$$V_{bond} = \frac{1}{2} k_{ij} (r_{ij} - b_{ij})^2$$



$$V_{angle} = \frac{1}{2} k_{ijk}^{\theta} (\theta_{ijk} - \theta_{ijk}^0)^2$$



$$V_{dihedral} = \sum_n k_{ijkl}^{\phi} (1 + \cos(n\phi_{ijkl} - \phi_{ijkl}^{n_0}))$$



$$V_{LJ} = \sum_{i < j} \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^6}$$

$$V_{Electrostatics} = \sum_{i < j} f \frac{q_i q_j}{\epsilon_r r_{ij}}$$

$$V = V_{bond} + V_{angle} + V_{dihedral} + V_{LJ} + V_{Electrostatics}$$

$$F = -\nabla V$$

Implementation of Classical Interactions

- Molecular topologies are fixed, so bonded interactions are implemented as static neighbor lists
 - One time expense at the beginning
- Non-bonded interactions are implemented as dynamic neighbor lists
 - Usually not updated at every time step
 - Only two body interactions, so relatively easy to implement.

Reactive systems

- Chemical reactions are association and dissociation of chemical bonds
 - Classical simulations cannot simulate reactions
 - ab-initio methods calculate overlap of electron orbitals to investigate chemical reactions
- ReaX force field postulates a classical *bond order interaction* to mimic the association and dissociation of chemical bonds¹

¹ van Duin *et al* , J. Phys. Chem. A, **105**, 9396 (2001)

Bond order interaction

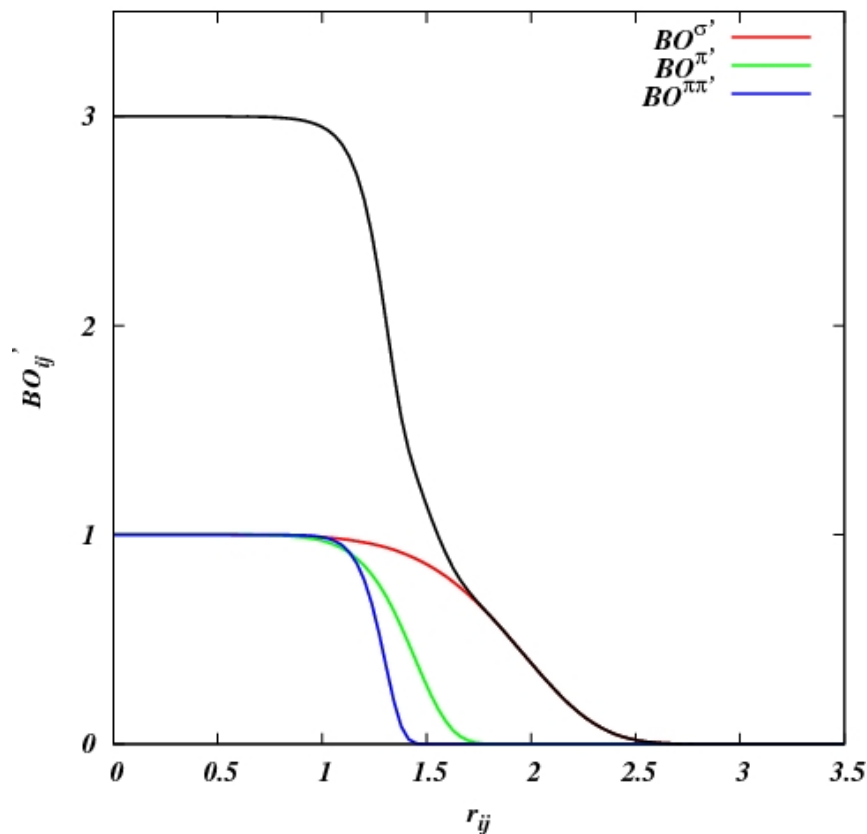
- Uncorrected bond order:

$$BO_{ij}^{\alpha'}(r_{ij}) = \exp \left[a_{\alpha} \left(\frac{r_{ij}}{r_{0\alpha}} \right)^{b_{\alpha}} \right]$$

Where α is for

σ - σ , π - σ , and π - π bonds

- The total uncorrected bond order is sum of three types of bonds
- Bond order requires correction to account for the correct valency



Bond order for C-C bond

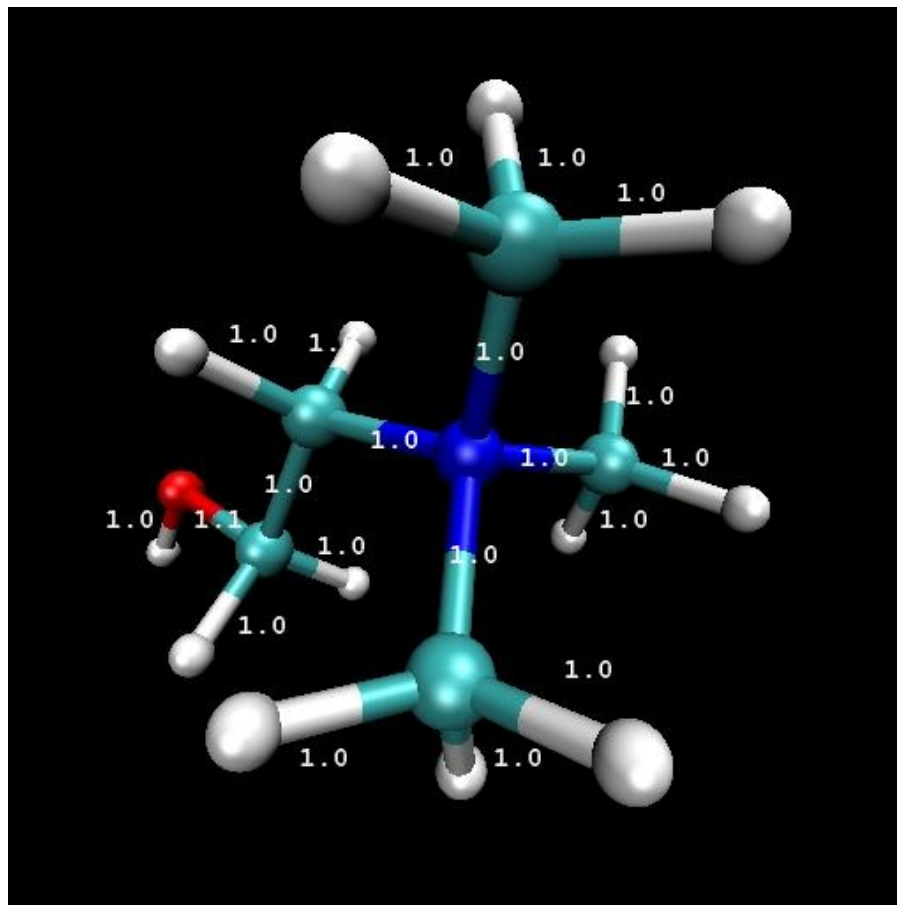
Bond Order Interaction

- After correction, the bond order between a pair of atoms depends on the uncorrected bond orders of the neighbors of each atoms
- The uncorrected bond orders are stored in a tree structure for efficient access.
- The bond orders rapidly decay to zero as a function of distance so it is reasonable to construct a neighbor list for efficient computation of bond orders

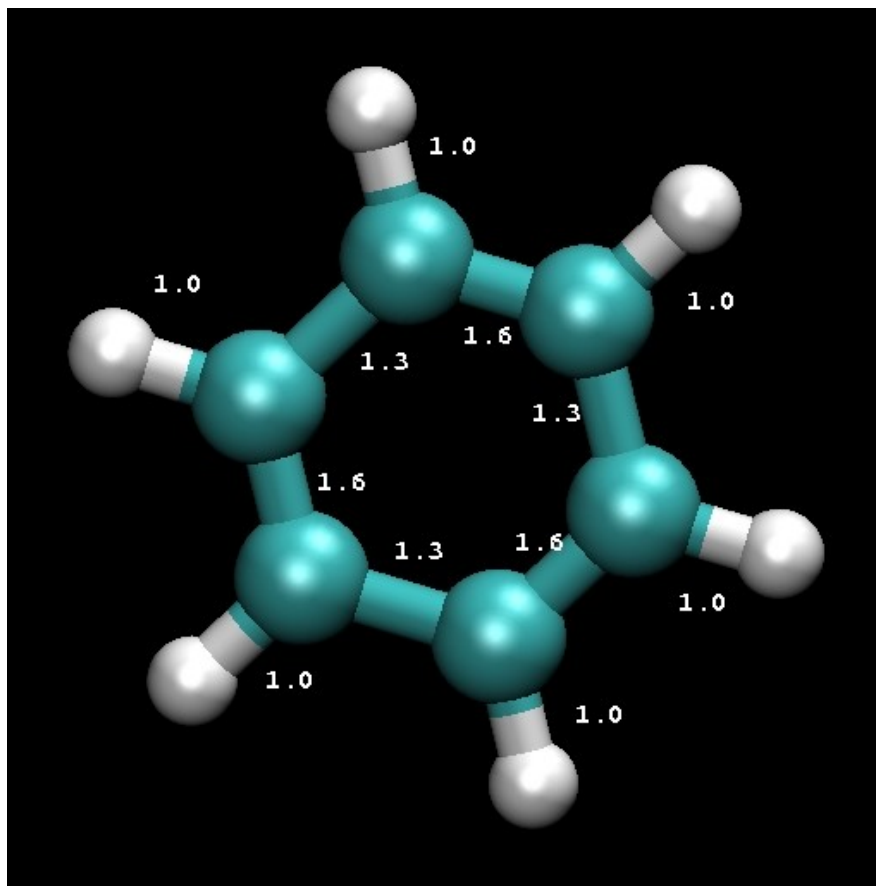
Neighbor Lists for Bond Order

- Efficient implementation critical for performance
- Implementation based on an oct-tree decomposition of the domain
- For each particle, we traverse down to neighboring octs and collect neighboring atoms
- Has implications for parallelism (issues identical to parallelizing multipole methods)

Bond Order : Choline



Bond Order : Benzene



Other Local Energy Terms

- Other interaction terms common to classical simulations, e.g., bond energy, valence angle and torsion, are appropriately modified and contribute to non-zero bond order pairs of atoms
- These terms also become many body interactions as bond order itself depends on the neighbors and neighbor's neighbors
- Due to variable bond structure there are other interaction terms, such as over/under coordination energy, lone pair interaction, 3 and 4 body conjugation, and three body penalty energy

Non Bonded van der Waals Interaction

- The van der Waals interactions are modeled using distance corrected Morse potential

$$V_{vdW}(r_{ij}) = D_{ij} \exp\left[\alpha_{ij} \left(1 - \frac{R(r_{ij})}{r_{vdW}}\right)\right] - 2D_{ij} \exp\left[\frac{1}{2}\alpha_{ij} \left(1 - \frac{R(r_{ij})}{r_{vdW}}\right)\right]$$

Where $R(r_{ij})$ is the shielded distance given by

$$R(r_{ij}) = \left(r_{ij}^\beta + \frac{1}{\lambda_{ij}^\beta}\right)^{1/\beta}$$

Electrostatics

- Shielded electrostatic interaction is used to account for orbital overlap of electrons at closer distances

$$V_{Ele}(r_{ij}) = f \frac{q_i q_j}{\left[r_{ij}^3 + \gamma_{ij}^{-3} \right]^{\frac{1}{3}}}$$

- Long range electrostatics interactions are handled using the Fast Multipole Method (FMM).

Charge Equilibration (QEq) Method

- The fixed partial charge model used in classical simulations is inadequate for reacting systems.
- One must compute the partial charges on atoms at each time step using an ab-initio method.
- We compute the partial charges on atoms at each time step using a simplified approach call the Qeq method.

Charge Equilibration (QEq) Method

- Expand electrostatic energy as a Taylor series in charge around neutral charge.
- Identify the term linear in charge as electronegativity of the atom and the quadratic term as electrostatic potential and self energy.
- Using these, solve for self-term of partial derivative of electrostatic energy.

Qeq Method

We need to minimize:

$$E_{ele} = \sum_i X_i q_i + \frac{1}{2} \sum_j H_{ij} q_i q_j$$

where

$$H_{ij} = J_i \delta_{ij} + \frac{1 - \delta_{ij}}{(r_{ij}^3 + \gamma_{ij}^{-3})^{1/3}}$$

subject to:

$$\sum_i q_i = 0$$

Qeq Method

$$E_u(\{q_i\}) = E_{ele}(\{q_i\}) - u \sum_i q_i$$

$$-\frac{\partial}{\partial q_i} E_u = -X_i + u - \sum_j H_{ij} q_j = 0$$

$$\sum_j H_{ij} q_j = -X_i + u$$

$$H \tilde{q} = -\tilde{X} + \tilde{u}$$

Qeq Method

$$q_i = \sum_k H^{-1}_{ik} (-X_k + u1_k)$$

$$\sum_i q_i = -\sum_i \sum_k H^{-1}_{ik} X_k + \sum_i \sum_k H^{-1}_{ik} u = 0$$

From charge neutrality, we get:

$$u = \frac{\sum_i \sum_k H^{-1}_{ik} X_k}{\sum_i \sum_k H^{-1}_{ik} 1_k}$$

Qeq Method

Let

$$u = \frac{\sum_i s_i}{\sum_i t_i}$$

where

$$s_i = -\sum_k H^{-1}_{ik} X_k$$

$$t_i = -\sum_k H^{-1}_{ik} 1_k$$

or

$$-X_k = \sum_i H_{ik} s_i$$

$$-1 = -\sum_i H_{ik} t_i$$

Qeq Method

- Substituting back, we get:

$$q_i = s_i - ut_i = s_i - \frac{\sum_i s_i}{\sum_i t_i} t_i$$

We need to solve $2n$ equations with kernel H for s_i and t_i .

Qeq Method

- Observations:
 - H is dense.
 - The diagonal term is J_i
 - The shielding term is short-range
 - Long range behavior of the kernel is $1/r$

Hierarchical methods to the rescue! Multipole-accelerated matrix-vector products combined with GMRES and a preconditioner.

Hierarchical Methods

- Matrix-vector product with $n \times n$ matrix – $O(n^2)$
- Faster matrix-vector product
 - *Matrix-free* approach
 - Appel's algorithm, Barnes-Hut method
 - Particle-cluster interactions – $O(n \lg n)$
 - Fast Multipole method
 - Cluster-cluster interactions – $O(n)$
- Hierarchical refinement of underlying domain
 - 2-D – quad-tree, 3-D – oct-tree
- Rely on decaying $1/r$ kernel functions
- Compute approximate matrix-vector product at the cost of accuracy

Hierarchical Methods ...

- **Fast Multipole Method (FMM)**
 - Divides the domain recursively into 8 sub-domain
 - Up-traversal
 - computes multipole coefficients **to give** the effects of all the points inside a node at a *far-way* point
 - Down-traversal
 - computes local coefficients **to get** the effect of all *far-away* points inside a node
 - Direct interactions – for *near by* points
- Computation complexity – $O((d+1)^4 n)$
 - d – multipole degree

Hierarchical Methods ...

- **Hierarchical Multipole Method (HMM)**
 - Augmented Barnes-Hut method or variant of FMM
 - Up-traversal
 - Same as FMM
 - For each particle
 - *Multipole-acceptance-criteria (MAC)* - ratio of distance of the particle from the center of the box to the dimension of the box
 - use MAC to determine if multipole coefficients should be used **to get** the effect of all *far-away* points or not
 - Direct interactions – for *near by* points
- Computation complexity – $O((d+1)^2 n \lg n)$

Qeq: Parallel Implementation

- Key element is the parallel matrix-vector (multipole) operation
- Spatial decomposition using space-filling curves
- Domain is generally regular since domains are typically dense
- Data addressing handled by function shipping
- Preconditioning via truncated kernel
- GMRES never got to restart of 10!

Qeq Parallel Performance

Size	Iterations	$P=1$	$P=4$	$P=32$
43,051	7	104	32 (3.3)	4.95 (21.0)
108,092	9	316	87 (3.6)	13 (24.3)
255,391	9	718	187 (3.8)	27 (26.5)

Size corresponds to number of atoms; all times in seconds, speedups in parentheses. All runtimes on a cluster of Pentium Xeons connected over a Gb Ethernet.

Qeq Parallel Performance

Size	$P=1$	$P=4$	$P=16$
43,051	73	21 (3.5)	3.2 (22.8)
108,092	228	61 (3.7)	9.2 (24.8)
255,391	508	132 (3.8)	19.1 (26.6)

Size corresponds to number of atoms; all times in seconds, speedups in parentheses. All runtimes on an IBM P590.

Parallel ReaX Performance

- ReaX potentials are near-field.
- Primary parallel overhead is in multipole operations.
- Excellent performance obtained over rest of the code.
- Comprehensive integration and resulting (integrated) speedups being evaluated.

Ongoing Work

- Comprehensive validation of parallel ReaX code
- System validation of code – from simple systems (small hydrocarbons) to complex molecules (larger proteins)
- Parametrization and tuning force fields.