6.189 Multicore Programming Primer, January (IAP) 2007

6.189 IAP 2007

Student Project Presentation

Molecular Dynamics

Molecular Dynamics on the Playstation 3

Greg Pintilie

Overview

- Molecular Dynamics
- Algorithm
- Parallelization Approaches

Molecular Dynamics

- Potential Energy:

$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$

Molecular Dynamics

- Potential Energy:

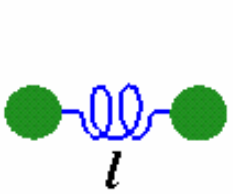
$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$

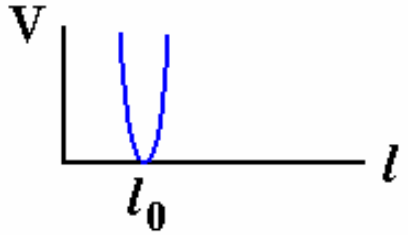
$$E_{bonded} = E_{bonds} + E_{angles} + E_{dih,imp}$$

Molecular Dynamics

- Potential Energy:

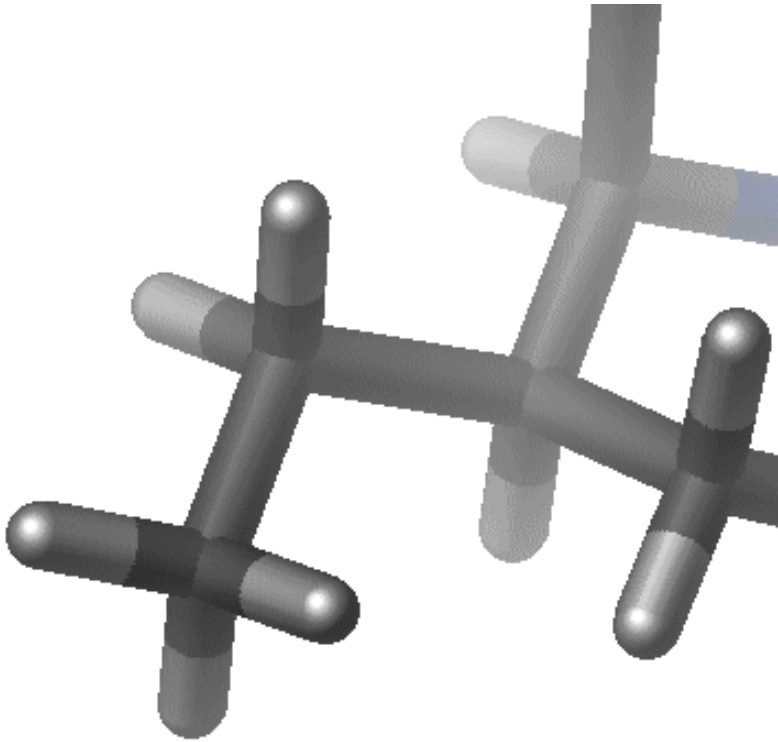
$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$





$$E_{bonded} = E_{bonds} + E_{angles} + E_{dih,imp}$$

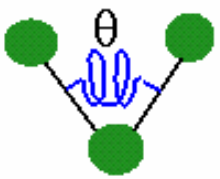
$$E_{bonds} = \sum_{bonds} k_b (l - l_0)^2$$

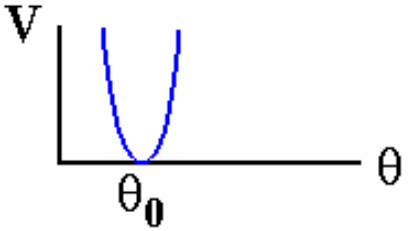


Molecular Dynamics

- Potential Energy:

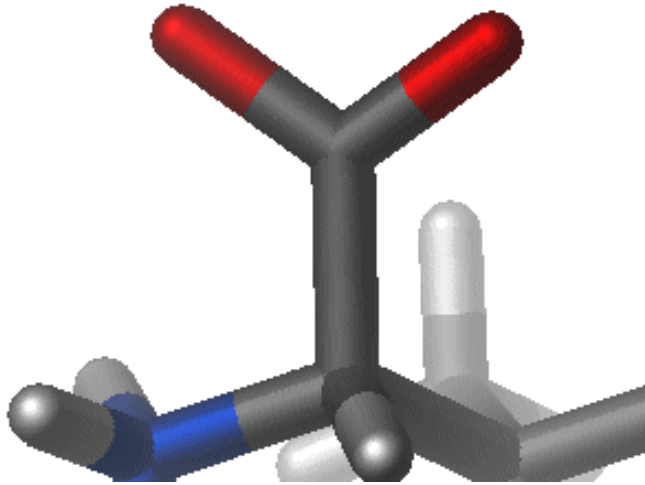
$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$





$$E_{bonded} = E_{bonds} + E_{angles} + E_{dih,imp}$$

$$E_{angles} = \sum_{angles} k_\theta (\theta - \theta_0)^2$$

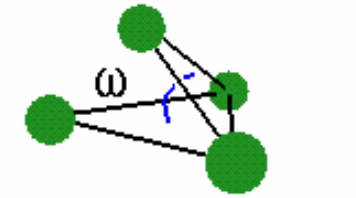


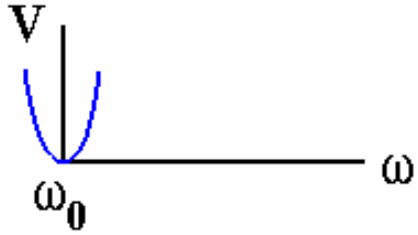
Molecular Dynamics

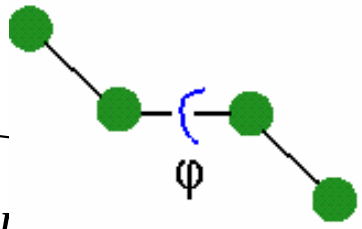
- Potential Energy:

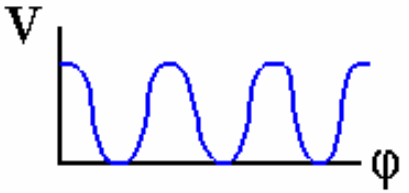
$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$

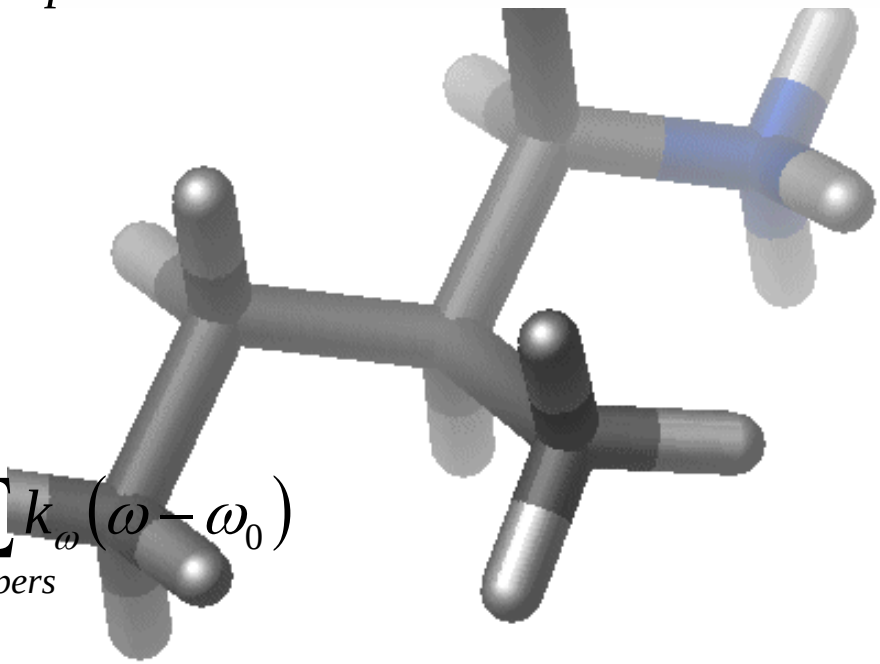
$$E_{bonded} = E_{bonds} + E_{angles} + E_{dih,imp}$$











$$F_{dih,imp} = \sum_{dihedrals} k_\phi(1-\cos(n\phi)) + k_\theta(\theta-\theta_0) + \sum_{impropers} k_\omega(\omega-\omega_0)$$

Molecular Dynamics

- Potential Energy:

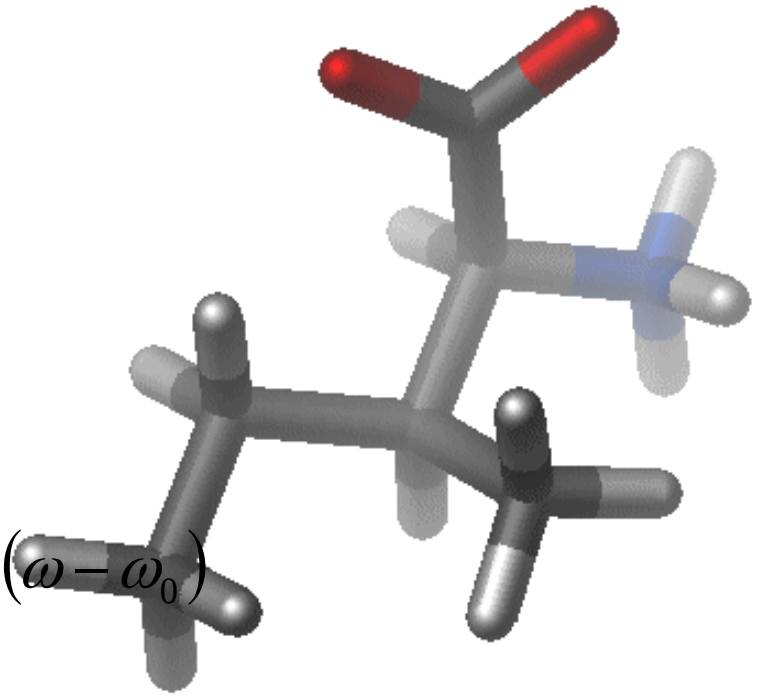
$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$

$$E_{bonded} = E_{bonds} + E_{angles} + E_{dih,imp}$$

$$E_{bonds} = \sum_{bonds} k_b (b - b_0)^2$$

$$E_{angles} = \sum_{angles} k_\theta (\theta - \theta_0)^2$$

$$F_{dih,imp} = \sum_{dihedrals} k_\phi (1 - \cos(n\phi)) + k_\theta (\theta - \theta_0) + \sum_{impropers} k_\omega (\omega - \omega_0)$$



Molecular Dynamics

- Potential Energy:

$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$

$$E_{non-bonded} = E_{van-der-Waals} + E_{electrostatic}$$

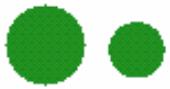
Molecular Dynamics

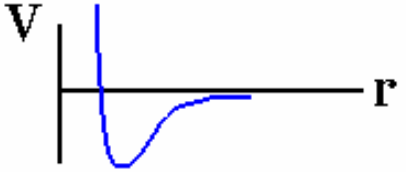
- Potential Energy:

$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$

$$E_{non-bonded} = E_{van-der-Waals} + E_{electrostatic}$$

$$E_{van-der-Waals} = \sum_{atoms\_i,k} \left(\frac{A_{ik}}{r_{ik}^{12}} - \frac{C_{ik}}{r_{ik}^{6}} \right)$$





Molecular Dynamics

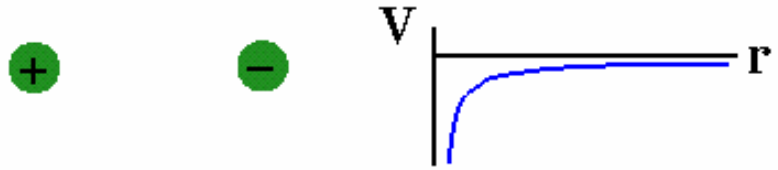
- Potential Energy:

$$E_p(\vec{x}) = E_{bonded} + E_{non-bonded}$$

$$E_{non-bonded} = E_{van-der-Waals} + E_{electrostatic}$$

$$E_{van-der-Waals} = \sum_{atoms\_i,k} \left(\frac{A_{ik}}{r_{ik}^{12}} - \frac{C_{ik}}{r_{ik}^{6}} \right)$$

$$E_{electrostatic} = \sum_{atoms\_i,k} \frac{q_i q_k}{D r_{ik}}$$



Molecular Dynamics

- Compute forces :

$$\vec{f}(t) = M\vec{a}(t) = -\frac{\partial}{\partial \vec{x}} E_p(\vec{x})$$

- Integrate to obtain velocity, position:

$$\vec{v}\left(t + \frac{1}{2}\Delta t\right) = v\left(t - \frac{1}{2}\Delta t\right) + \Delta t \frac{\vec{f}(t)}{m}$$

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \Delta t \cdot \vec{v}\left(t + \frac{1}{2}\Delta t\right)$$

Kinetic Energy

- Kinetic Energy/Temperature:
  - from classical equipartition theory, each degree of freedom has, at thermal equilibrium, this much energy: $\frac{1}{2}k_B T$

$$\langle E_k \rangle = \frac{1}{2}\sum_{i=1}^{3N} m_i v_i^2 = \frac{1}{2} N_F k_B T$$

Langevin Dynamics

- Account for collisions with imaginary molecules (heat bath)
- e.g. in solvent such as water

$$F = M\vec{a} = -\nabla E_p(\vec{x}) - \gamma M v + R(t)$$

$$\langle R(t) \rangle = 0 \qquad \langle R(t), R(t')^T \rangle = 2\gamma k_B T M \delta(t - t')$$

Solvation in Dielectric Material

- Molecules that are polar/ionic ‘shield’ electrostatic forces
- Water:
  - distance-dependent dielectric:

$$E_{electrostatic} = \sum_{atoms\_i,k} \frac{q_i q_k}{D r_{ik}}$$

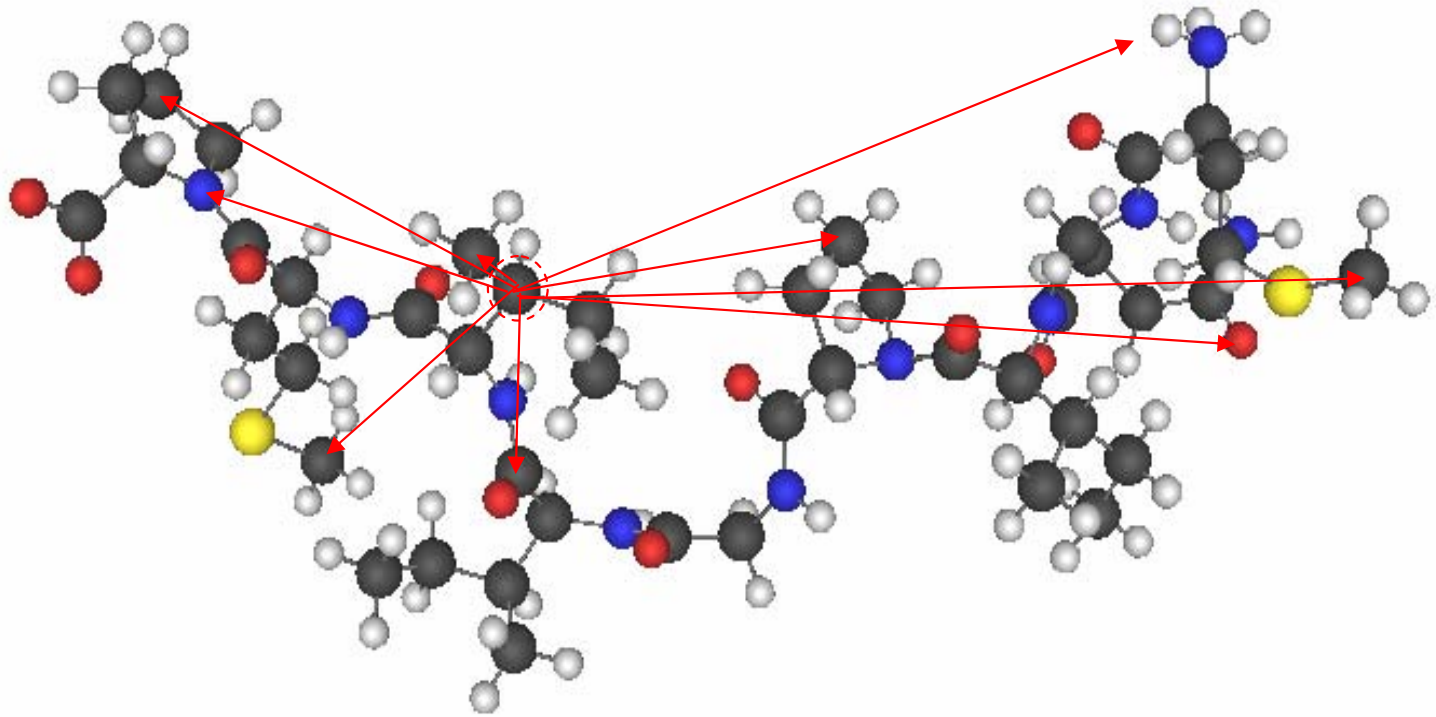
$$D = r$$

$$E_{electrostatic} = \sum_{atoms\_i,k} \frac{q_i q_k}{r_{ik}^{2}}$$

Non-bonded Cut-offs

$$E_{non-bonded} = \sum_{atoms\_i,k} \left\{ \left(\frac{A_{ik}}{r_{ik}^{12}} - \frac{C_{ik}}{r_{ik}^{6}} \right) + \frac{q_i q_k}{D r_{ik}} \right\}$$

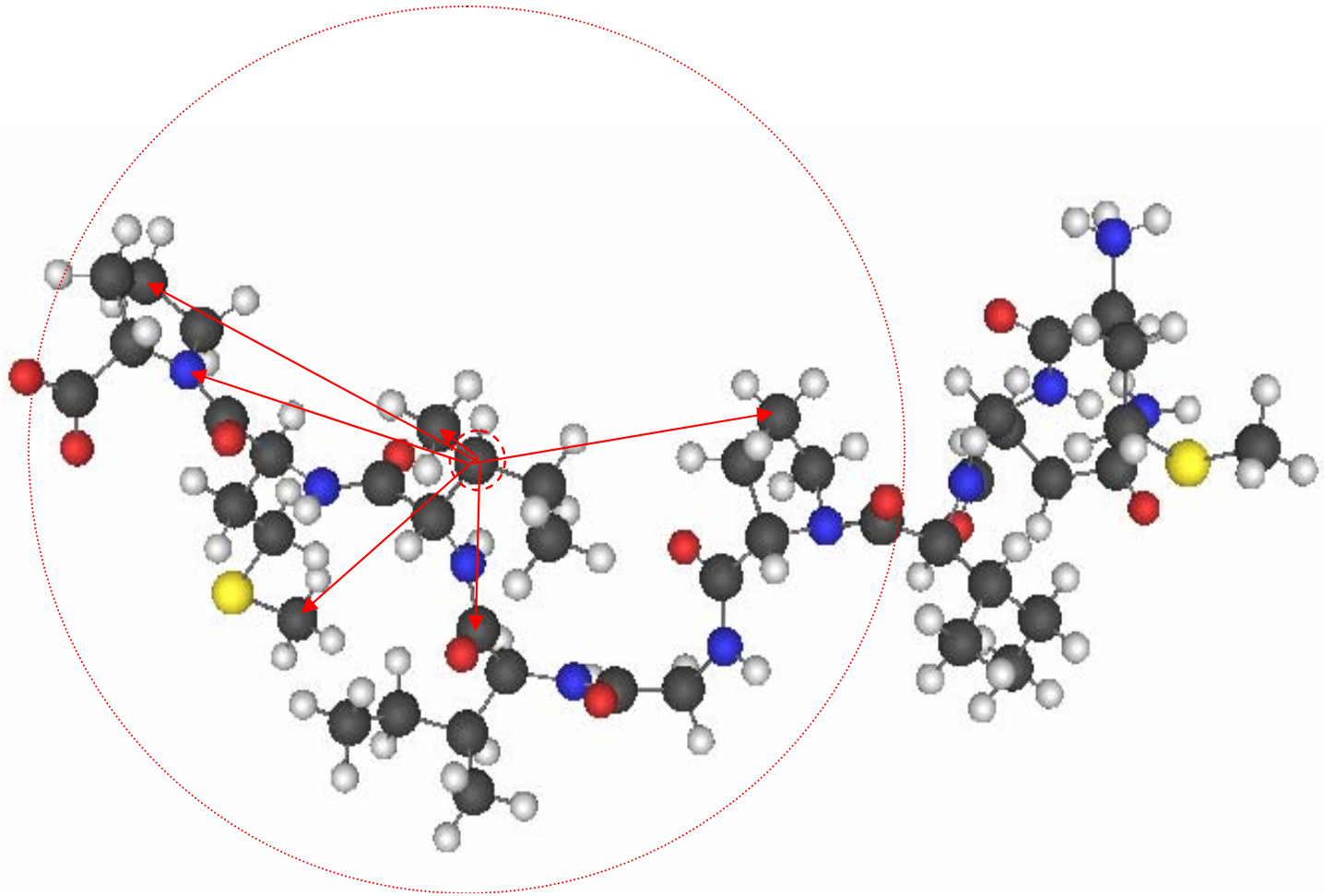
- Cut-off ~12A



Non-bonded Cut-offs

$$E_{non-bonded} = \sum_{atoms\_i,k} \left\{ \left(\frac{A_{ik}}{r_{ik}^{12}} - \frac{C_{ik}}{r_{ik}^{6}} \right) + \frac{q_i q_k}{D r_{ik}} \right\}$$

- Cut-off ~12A



Basic MD Algorithm

```
For i=0 to numsteps

    if i % ap_freq == 0
        find atom pairs

    Compute ‘bonded’ forces
        bonds, angles, dihedrals,
        impropers

    Compute ‘non-bonded’ forces
        atom pairs

    Integrate
```

Data Structures

Vector
double x, y, z

Atom
Vector pos, vel, force
double Mass
double Charge
double Rmin
double Eps

Molecule
list Atom atoms
list Bond bonds
list Angle angles
list Improper impropers
list Dihedrals dihedrals
list AtomPair atompairs

Bonded Forces

Bond
Atom *a1, *a2
double k, b0

Angle
Atom *a1, *a2, *a3
double k, t0

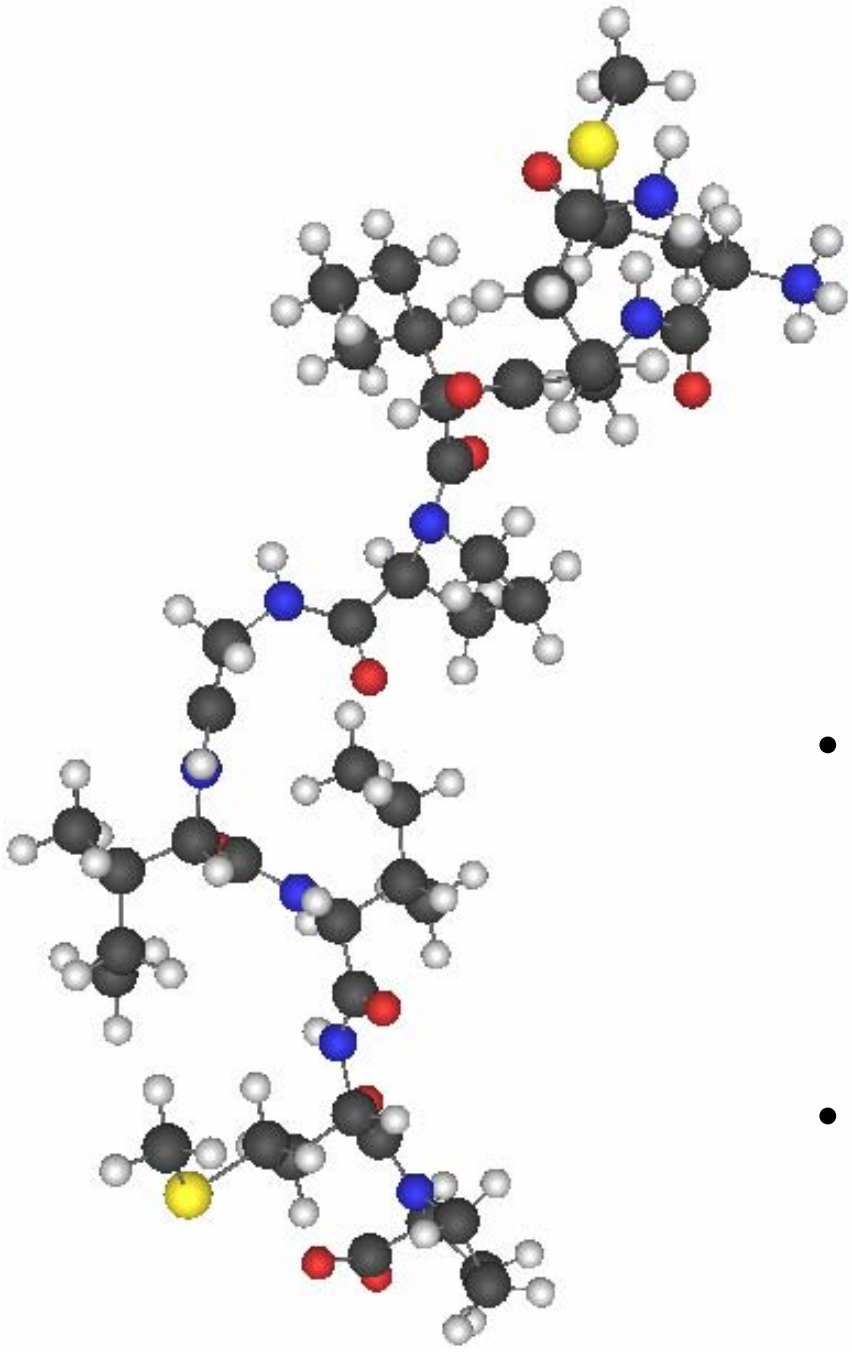
Improper
Atom *a1, *a2, *a3, *a4
double k, t0

Dihedral
Atom *a1, *a2, *a3, *a4
list DihedralValue vals

DihedralValue
double k, phase
int n

Non-Bonded Forces

Atom Pair
Atom *a1, *a2
double eij



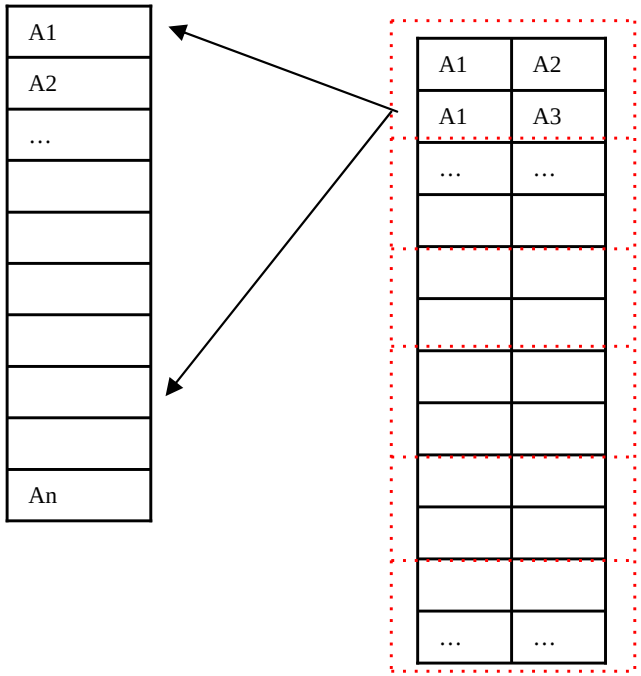
- A8m
  - ‘Bonded’ - total 41,652
    - 146 atoms x 104 bytes = 15,184
    - 147 bonds x 24 bytes= 3,528
    - 275 angles x 28 bytes = 7,700
    - 393 dihedrals x 16 bytes +
      414 dihedral values x 20 bytes = 14,568
    - 21 impropers 32 bytes = 672
  - ‘Non-bonded’ – total 176,000
    - 11,000 Atom Pairs x 16 bytes
      - (1-3 bonded atoms excluded)
- 10 x A8m
  - ‘Bonded’ – total 416,520
  - ‘Non-bonded’ – total 16,976,000
- 20 x A8m
  - ‘Bonded’ – total 833,040
  - ‘Non-bonded’ – total 68,064,000

Sequential Algorithm

`For i=0 to numsteps`	No cutoff	With cutoff
`if i%ap_freq==0` `find atom pairs`	0 ms	Bf / Kd 6,090 / 880 ms 24,440 / 1,750 ms
`‘bonded’ forces` `bonds, angles, dihedrals, impropers`	50ms	50ms
`‘non-bonded’ forces` `atom pairs`	1,480ms 1,060,850 pairs	150ms 116,434 pairs
`integrate`	10ms	10ms

Parallelization Approaches

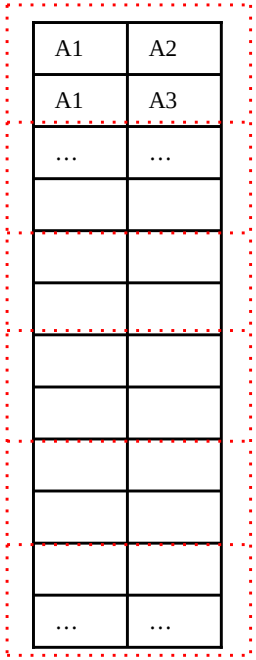
- Force Decomposition



- force operation includes both atom positions, returns the force on both atoms
- scales well with system size and #processors

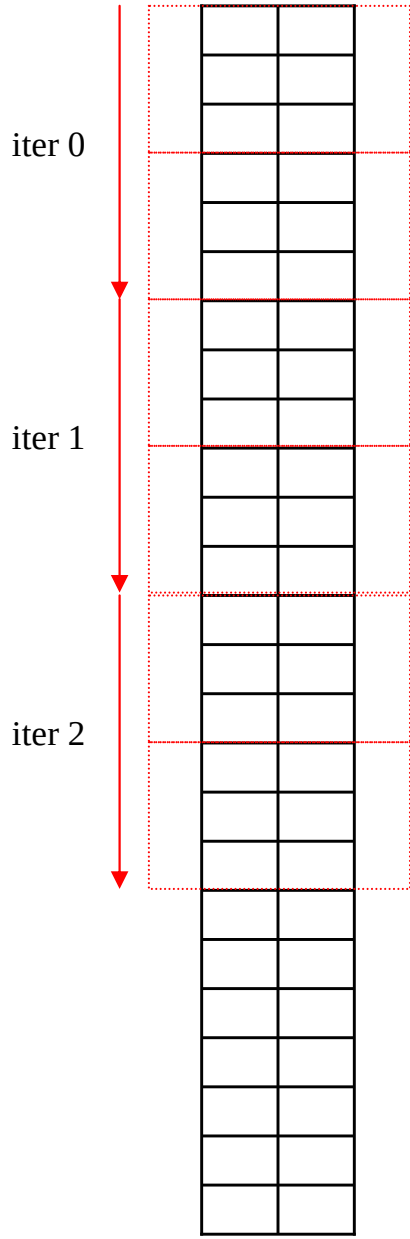
Parallelization Approaches

- Force Decomposition



PPU

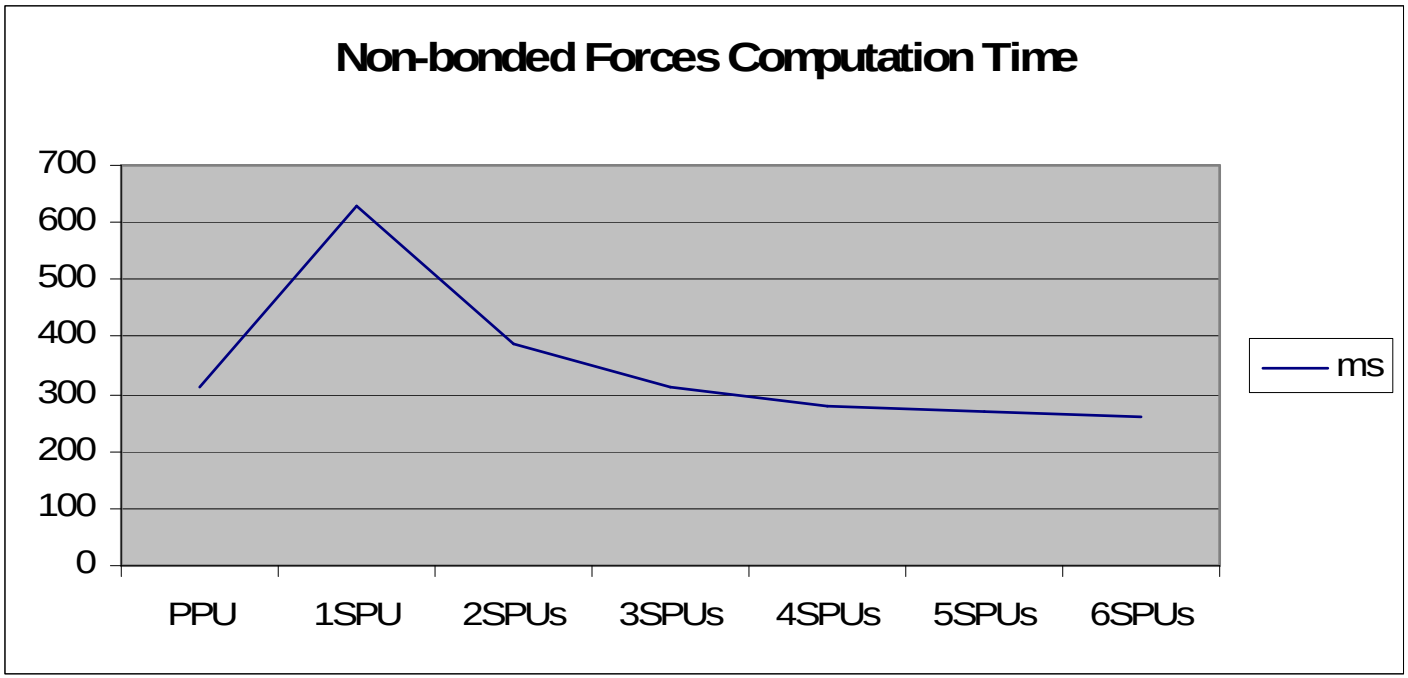
```
for j=0 to #SPUs
     •send control block to SPU-j
for step i=0 to num steps
     •compute bonded forces
     •compute non-bonded forces
          • while non-bonded operations remaining
               • for j=0 to #SPUs
                    • create block with force-operations (200)
                    • send control block with #ops to SPU-j
                    • tell SPU-j to start processing
               • for j=0 to #SPUs
                    • if SPU-j finished, add forces to atoms
               • break if all SPUs finished
     • integrate forces
```



Parallelization Approaches

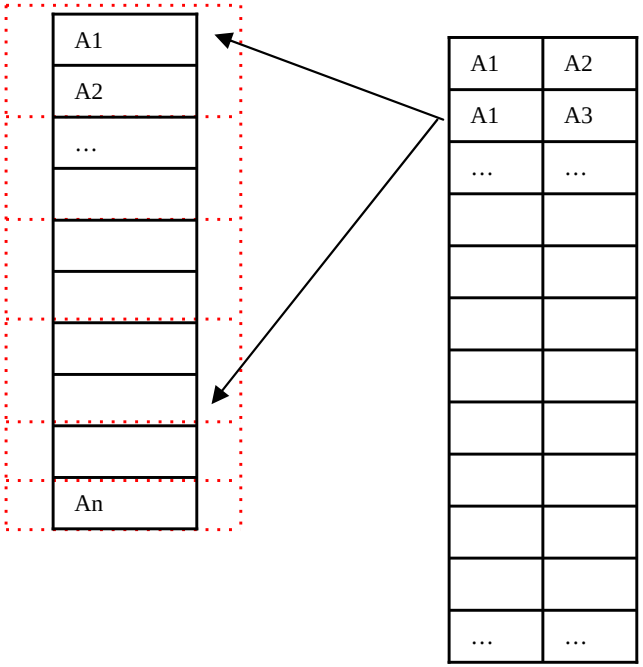
- Force Decomposition - performance

PPU	1SPU	2SPUs	3SPUs	4SPUs	5SPUs	6SPUs
310	630	390	310	280	270	260



Parallelization Approaches

- Atomic Decomposition

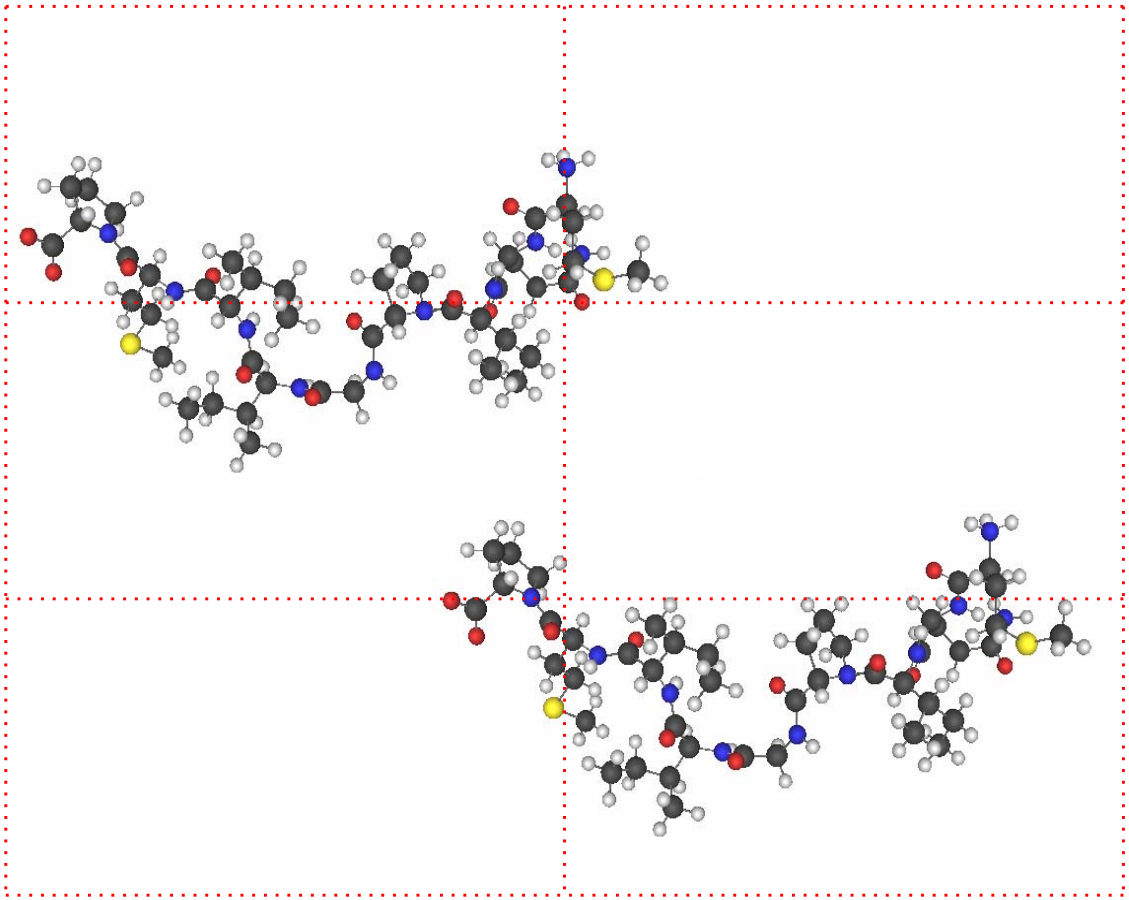


• atoms and forces stored independently

• doesn’t scale as easily with system size

Parallelization Approaches

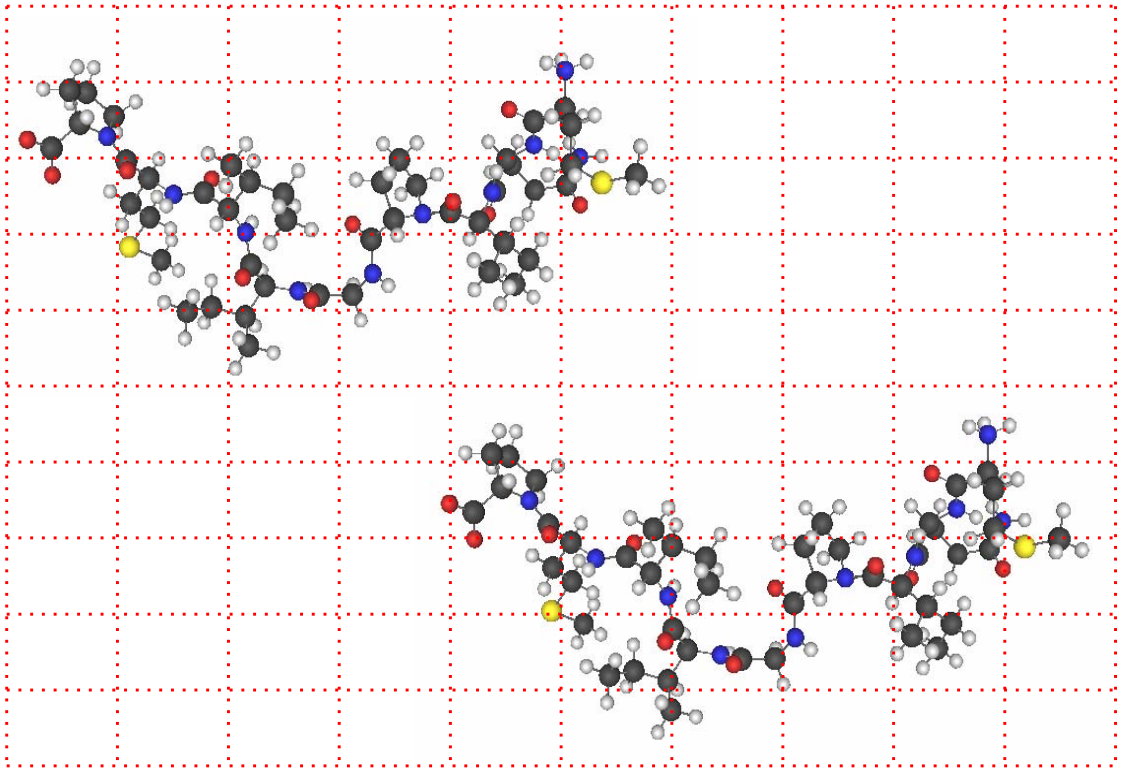
- Spatial Decomposition

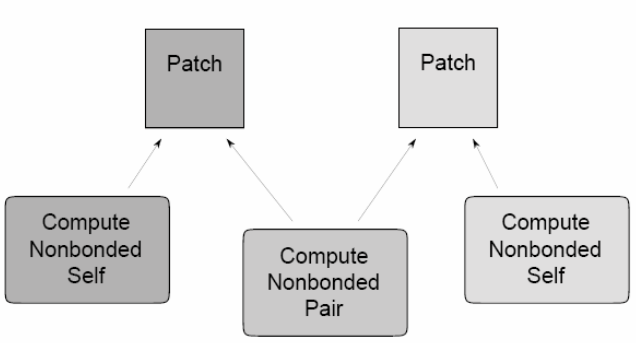


- not load-balanced
- atom positions must be communicated between processors
- periodically re-assign atoms

State of the Art - NAMD

- force-spatial decomposition





Images removed due to copyright restrictions.