

StreamMD

Molecular Dynamics

Eric Darve

MD of water molecules

- Cutoff is used to truncate electrostatic potential

$$U(r) = 0, \quad \text{for } |r| > r_{cutoff}$$

- Gridding technique: water molecules are grouped in clusters of size r_{cutoff} .
- Problem: what data structure do we use to store this array of clusters?

- Option: 3D array of streams.
- Needs to be updated at each time step (or every 10 time steps) to account for molecules that enter or leave a cluster.
- Q: how do I move data from one stream to another?
- Q: how does the compiler take care of the load balancing?
- Data locality defined by geometric distance between clusters, i.e.
If two clusters are neighbors, they should be physically located in neighbor processors.

Current implementation

- After position has been updated:
 - Loop over all clusters $C[i][j][k]$:
 - We push in $C[i][j][k]$ if water molcl is in same cluster.
 - We push in C_{tmp} otherwise.
 - C_{tmp} is then concatenated in a large stream Mol which contains all the molecules which have exited their cell.
 - Kernel is exec. to push molecules in Mol to $C[][][]$.

MD of proteins

- Three possible types of potentials:
Bond stretch: two atoms and 1 bond.
Bond angle: three atoms and 2 bonds.
Torsion angle: four atoms and 3 bonds.
- Typical organization of the computation:
loop over bond stretch, bond angle and torsion potentials.
- Brook needs some information to map the data in an efficient manner.

- Information is static: connectivity between atoms does not change throughout the run.
- However the information is not accessible to the compiler: run-time information.
- Protein backbone: sequence of amino-acids. Typically 10-20 atoms each.
- User can provide following information: for each atom, give residue number.

- “Distance” between two data can then be defined as the difference between residue number.
- Run-time optimization approach:
all atoms with same residue number are on the same processor.
2 processors with nearby residue numbers should be physically close to each other.

Load balancing

- Typical size of protein: 10 residues for test proteins to hundreds for real life problem.
- Size is relatively small but computationally very expensive.
Well optimized load balancing and distribution of data is critical.
- Example: Duan and Kollman '98. 36-residue protein + 3000 water molecules.
1/2 billion time steps: four months on 256-processor computer.
Complete fold would have required: x 10, x 100.