Molecular Dynamics on GPUs

Ian Buck, Vishal Vaidyanathan, Vidya Rangasayee, Vijay Pande, Eric Darve
Computer Science Dept, Mechanical Engineering Dept, Chemistry Dept
Stanford University

Background
This project attempts to use the massive compute power available in today's GPUs to model the folding of human proteins. Learning more about this process will have a tremendous impact on biology and medicine and will help medical researchers understand diseases and find cures.

One of the leading software packages used to simulate protein folding is Gromacs. Up to 80-90% of the computation time in simulating a molecular system is computing the non-bonded or intermolecular forces, usually consisting entirely of the water molecules surrounding a protein. In this work, we focus on this force calculation for implementation on the GPU as well as exploring additional algorithms deployed in Gromacs.

Numerical ODE in Molecular Dynamics
• Gromacs simulates molecular dynamics as an N-body problem
  - Newton's equations of motion
    \[ m_i \ddot{x}_i = -\nabla U(x_1, \ldots, x_n) = F(x_1) \]
  - Potential energy \( U \) is defined as the sum of several terms:
    ➢ Non-bonded Interactions
    - Coulomb electrostatics
      \[ \frac{q_i q_j}{4\pi\varepsilon_0 |r_i - r_j|^2} \]
    - Lennard-Jones vdW
      \[ \frac{1}{|r_i - r_j|^6} \]
    ➢ Bonded Interactions
    - Bond stretching
      \[ \frac{1}{2} k_i \Delta r_i^2 \]
    - Bond bending
      \[ \frac{1}{2} k_i \theta_i^2 \]
    - Torsion
      \[ 1 + \cos(n\theta) \]
  ➢ The Ordinary Differential Equations (ODE) can be solved with the Leap-frog scheme.
    \[ \ddot{x}_i(t + \Delta t) = \ddot{x}_i(t) + \Delta t \dot{x}_i(t) + \frac{\Delta t^2}{m_i} F(x_i(t)) \]
  ➢ Using this method, it is possible to simulate the complex trajectories of atoms and molecules for very long periods of time. However, only the advent of more powerful supercomputers will allow studying molecules over time scales which are biologically and experimentally relevant.

Gridding Technique
• The most expensive part of the simulation is the computation of the forces \( \vec{F} (\vec{r}_i) \).
• A cutoff is applied so that all particles which are at a distance greater than a cutoff radius do not interact.

Force Calculation Algorithm
1. Load initial input data into streams. Because of hardware constraints, all neighbor lists must be of the same length for the code to run efficiently. Consequently, molecules with extra neighbors have multiple neighbor lists. Molecules with too few neighbors are given dummy neighbors.
   A. Molecule list:
   
   B. Neighbor list:
   
   2. Compute the forces between each molecule and the molecules in its neighbor list(s).
   Store the forces in a new stream.

   3. Reduce forces to compute total molecule force.
   C. Partial Forces:
   
   4. Total Forces:
   
   D. Total Forces:

Maintaining Bond Constraints
After the force update, we must apply the bond constraints to connected atoms. Gromacs uses the SHAKE algorithm, an iterative constraint solver.

Updating atom positions:

<table>
<thead>
<tr>
<th>Pass</th>
<th>Force Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,2) \rightarrow (1',2')</td>
</tr>
<tr>
<td>2</td>
<td>(2,6) \rightarrow (2'',6')</td>
</tr>
<tr>
<td>3</td>
<td>(2',3) \rightarrow (2'',3')</td>
</tr>
<tr>
<td>4</td>
<td>(3',4) \rightarrow (3'',4')</td>
</tr>
<tr>
<td>5</td>
<td>(4',5) \rightarrow (4'',5')</td>
</tr>
</tbody>
</table>

SHAKE for GPUs operates on disjoint atoms in parallel:

Preliminary Results

Brook Implementation
Brook is an extension of standard ANSI C and is designed to incorporate the ideas of data parallel computing and arithmetic intensity into a familiar and efficient language. Brooke provides a familiar C-like programming environment for GPU-based computing.

• Brook files can co-exist within existing C source. This allows us to port only the compute intensive portions of GROMACS to Brook. The compiled Brook files link with the rest of Gromacs code.
• Brooke simplifies GPU programming permitting developers with no graphics programming experience to benefit from GPU-based computing.

• Brook supports both Linux and Windows, NVIDIA and ATI, OpenGL and DirectX. This allows us to search for the fastest compute platform for molecular dynamics.

Current Status
• Our goal is to have all key components of Gromacs executing on the GPU.
  ➢ We are also looking at other aspects of biocomputing:
    - Protein matching: Comparing the sequence of a target protein against known protein sequence databases.
    - Protein docking: Exploring the fit between two proteins to predict potential active sites.

Brook kernel execution on GPUs