CMSC416: Introduction to Parallel Computing

Topic: Parallel CSE Applications
Date: April 30th, 2024

Parallel CSE Applications

Molecular Dynamics

Parallel computing is used in various domains, from modeling atoms and molecules, to computational astronomy. There is a huge diversity in fields: trading on Wall Street, computation epidemiology, climate modeling, and much more.

Molecular dynamics is related to calculating the trajectories of atoms and molecules often used for drug design (pharma companies looking to design more effective drugs). This involves large teams of computer scientists using computer simulations to figure out if drugs are effective. Though we are not at the level of simulating a whole human body's worth of atoms, the scale of modeling has grown over time.

Materials design is another big field in which molecular dynamics is used. It is used to figure out what new materials we can use to design things like chips, optical cables, and many other things.

A **molecular system** is defined in a three-dimensional simulation box: we have some number of atoms, molecules and we are trying to figure what forces are applied to the atoms. After figuring out the forces, we then apply Newton's equations of motion to calculate new positions of atoms.

Q: What are some forces that apply?
A: Atoms are connected by 3 types of **bonds**. These include bonds which are 2 atoms connected, angles which 3 atoms connected to create an angle, and dihedrals which are 4 atoms connected together. There are also 2 types of **non-bonded interactions**, which include electrostatic forces (related to electrical charges) and Van der Waal's forces anything not related to electrical charges.

There are thousands to millions of atoms in these simulations. In order to simulate the life of a biomolecule, we split it up into **very small time steps**, and at each time step we calculate forces. A single simulation step can be ~1 femtosecond (10^-15 seconds) and simulations often span around 1 millisecond. The more atoms we are simulating, the more timesteps we have to simulate.

Molecular Dynamics - Sequential Algorithm

The **sequential algorithm** for molecular dynamics is outlined below:

1. At every time step, we are **calculating forces** on each atom in the system.
2. We calculate bonded and short-range forces every step.
3. We calculate long-range non-bonded forces every few time steps
    a. It would be **O(N^2)** every time step!
    b. Instead of calculating this every time step, every few time steps we calculate a Particle mesh Ewald (PME) summation which approximates the long-range interactions in Fourier space
    c. This helps our simulation be more scalable
4. We then calculate velocities, accelerations, new positions -> update new positions
5. Repeat…

Molecular Dynamics - Parallel Algorithms

Let's first look at parallelizing the calculation of short range forces.

The simplest approach is called **atom decomposition.** The idea is to partition the atoms across process **(# of atoms / # of processes)**. We are not taking the 3D space into account, we are approaching the atoms as a linear list. Some drawbacks with this method is load imbalance and large communication overhead as the atoms are assigned without taking 3D space into account.

The second approach is called **force decomposition**. This approach is to create a **force matrix** which is essentially a sparse, non-uniform 2D matrix of atoms. The matrix represents what calculations need to be done: a particular cell represents the work that needs to be done between two atoms. We try to assign an equal amount of atoms to each process, which in turn helps us with load imbalance but this might not solve communication issues.

A much more common approach nowadays is a **spatial decomposition**. The goal of this method is to have nearby atoms on the same processor.

Here are some steps of spatial decomposition:

1. We create smaller sub-boxes and assign each sub-box to a process.
2. We only calculate forces in the cutoff-radius of 16 angstrom (unit of length).
3. We make boxes of 16 angstroms x 16 angstroms. This makes sure that any sub-box in the system will only need atoms in the 26 neighboring boxes/processes.
4. Each sub-box computes the forces on the atoms in its box.

5. It will need to communicate with boxes around it to get other atoms within its cutoff-radius.
6. If atoms should be calculated with atoms from other boxes, they will be sent to other boxes.

Q: Any issues with this approach?
A: Once you have the size of the simulation and the size of the cutoff-radius, the number of processes is restricted. If we have more processes than boxes, we can reduce one of the axis sizes (16x16 -> 16x8), however this creates more communication.

The last approach is **hybrid parallelization** which combines force and space decomposition. We still take our 3D boxes and divide it into boxes and assign the boxes to processors. **However,** instead of having a certain process handle the atoms and force calculations, there will be **separate processes** in charge of force calculations.

In addition, we will **create an object** every time we need to compute forces between atoms in different boxes and then independently assign these objects to processes who will compute the forces.

There may be more communication, but it is much more scalable. We first do a spatial decomposition by assigning atoms to processes, but similar to the force decomposition we are assigning calculations between different boxes to independent processes.

The last approach is the **Neutral Territory** method. This is very similar to hybrid decomposition, but the creators came up with a different way to divide the simulation box.

The idea is to split the box into: towers and plates around each tower, representing a box. When doing force calculations: if the atoms are in the **same sub-box**, then the midpoint is in that box and that process calculates the forces. If the atoms are in **separate sub-boxes**, then whichever box contains the midpoint, the process associated with that box will calculate the forces.

Molecular Dynamics - Long Range Forces

Let's look at long range forces again! The **Particle mesh Ewald** (PME) is a method for calculating the long-range forces.

This involves long-range forces which are calculated in **Fourier space** using **fast-fourier transforms**. After they are calculated, we then bring back the results to real space. We can then create a 3D **charge grid** which represents the charge densities of all the atoms.

We can compute a **3D Fast Fourier Transform** on this charge grid. This FFT can either compute a discrete Fourier transform (DFT) or inverse DFT. Ultimately, this process reduces the time complexity of calculating long-range forces from O(N^2) to **O(NlogN)**.

Now, let's look at the **parallelization of PME** using a 3D FFT! The process involves bringing all the data to one process and then assigning 1D slabs of data to each process. We can also do a 2D decomposition of the charge grid called "pencils", and assign each "pencil" to a process.

<u>Computational Epidemiology - Intro</u>

Epidemiology is the study of how diseases occur and spread among human populations. This is an important societal challenge as it is critical to control the spread of infectious diseases. It is also important to study how different interventions (i.e vaccines) affect these diseases.

Some **challenges** of modeling epidemiology are the sheer number of people that need to be simulated. In addition there has been a vast increase in density of urbanization, local and global travel, and the immuno-compromised population.

One approach to modeling computational epidemiology is **individual-based simulation**. We can create mathematical equations that take in parameters and compare simulated curve to real curve

Another method is **agent-based modeling**. This involves creating computational simulations which simulate how people move and how people interact. We can create a **bi-partite graph** of people and locations including each person's schedules. We can then look at when people come into contact and how that causes diseases to spread.