N-Body simulation with MPI

CS484: Final Project

Himanshi Garg[1] Nick Christensen[2] Thilina Ratnayaka[2]

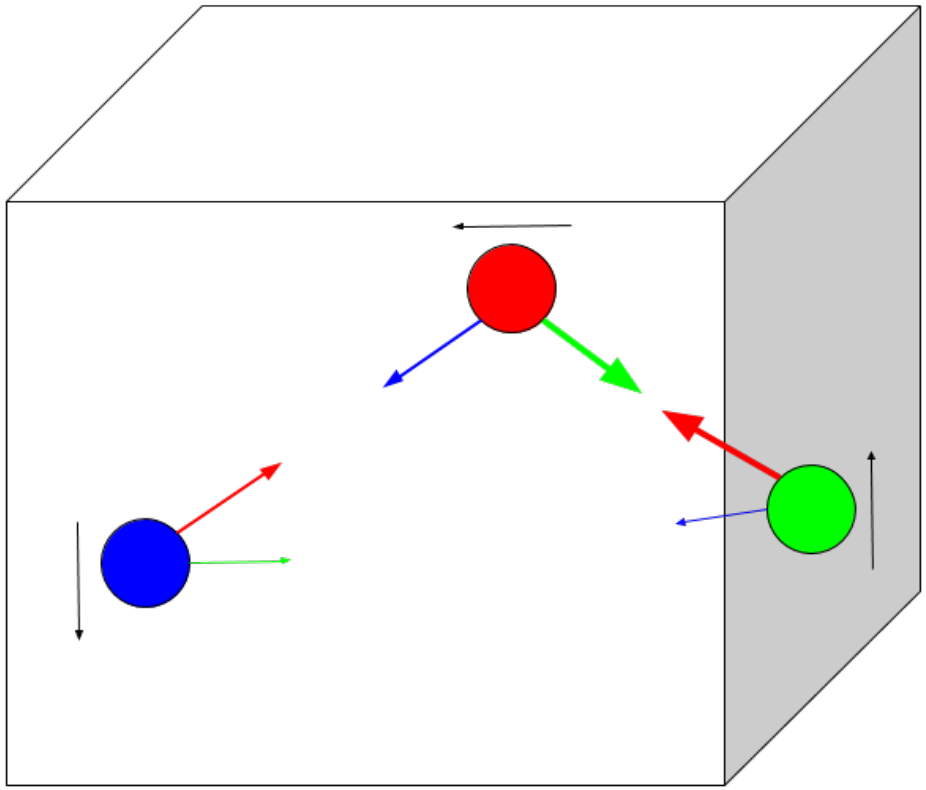
[1]Department of ECE

[2]Department of Computer Science

April 28, 2017

Outline

What is N-Body problem?



What is N-Body problem?

- Simulates the evolution of a system of N bodies where the force each body is due to attraction to/repulsion from other bodies in the system
- Bodies have masses, initial velocities, and initial positions. Determine the subsequent motion of bodies by numerically integrating equations of motion
- Applicable to astrophysics, molecular dynamics and plasma physics and many other fields

What are we doing?

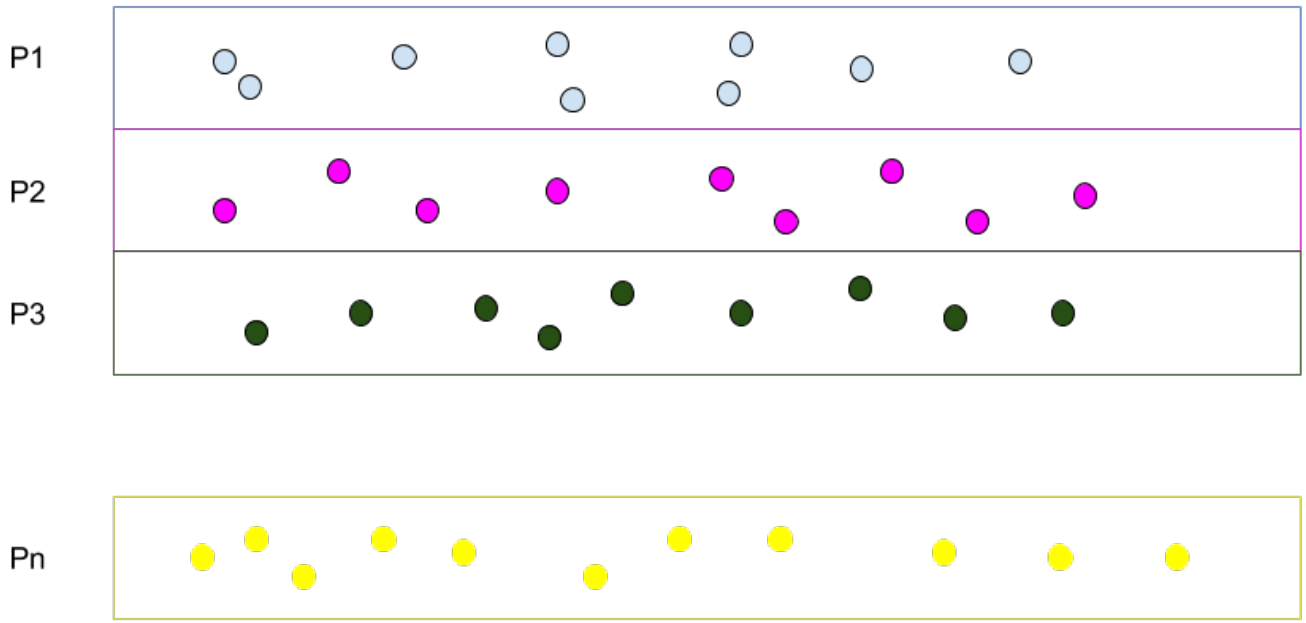
- Simulating inert gas particles
- Lennard-Jones potential to calculate the forces between pairs of neutral atoms is $4\epsilon\left(\left(\frac{\sigma}{\mathbf{r}}\right)^{12} - \left(\frac{\sigma}{\mathbf{r}}\right)^{6}\right)$
- Discretize time, calculate force at each time step, use force to update velocity and velocity to update position
- Approximates continuously changing force with constant force on a small time interval Δt

What is the simplest way to do this?

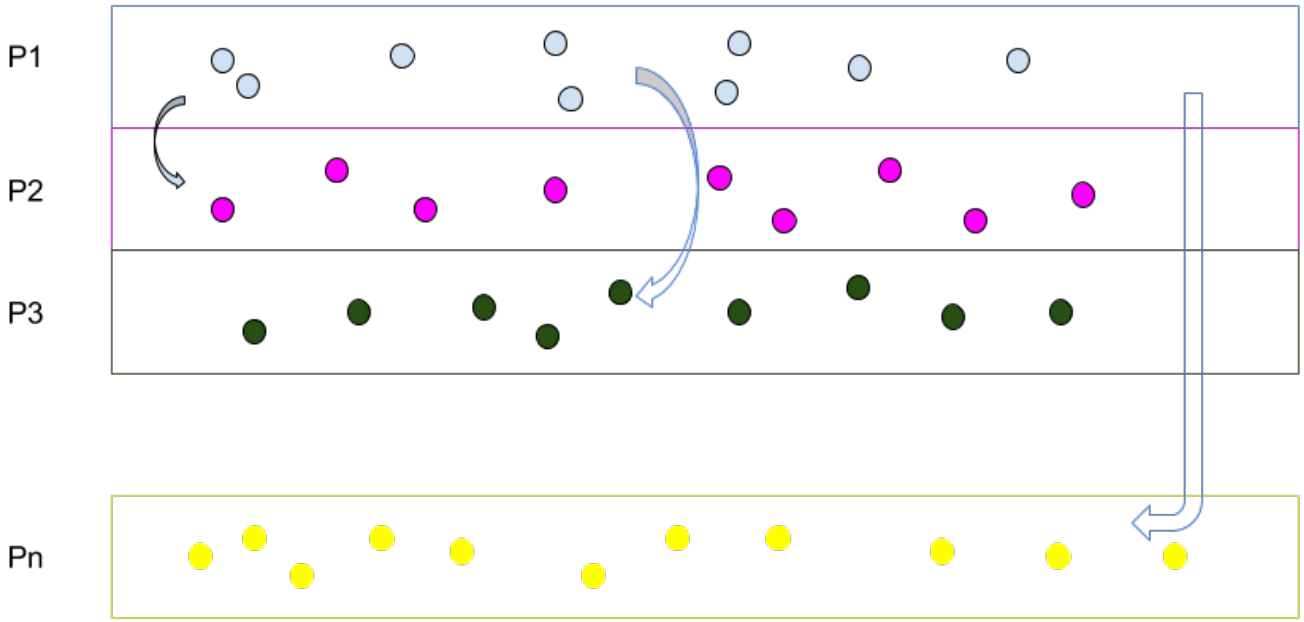
- Evaluate all pair-wise interactions among the N bodies
- Total force on each particle is the superposition of the forces imparted by each of the other particles
  - Advantage: More accurate results
  - Disadvantage: Computational complexity: $\mathcal{O}\left(N^2\right)$ for N particles $\implies$ Not scalable
- Optimization: Far away bodies don't add much to net force, ignore their force contributions
  - Still have to check position of every other particle, inaccurate if far away bodies exert substantial force in aggregate

Thinking in parallel ...

- Each processor calculates the net force on a subset of particles



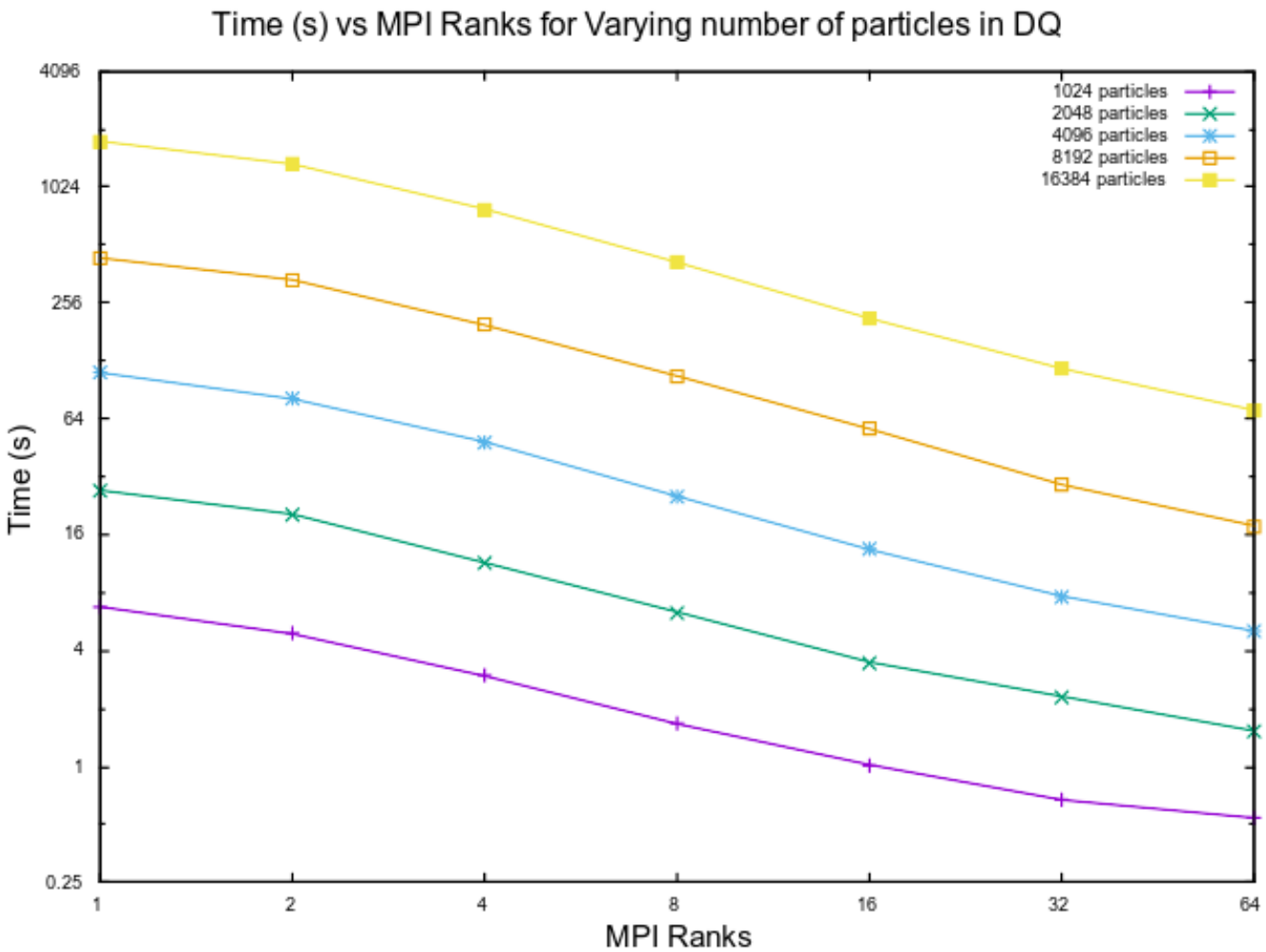
Thinking in parallel ...



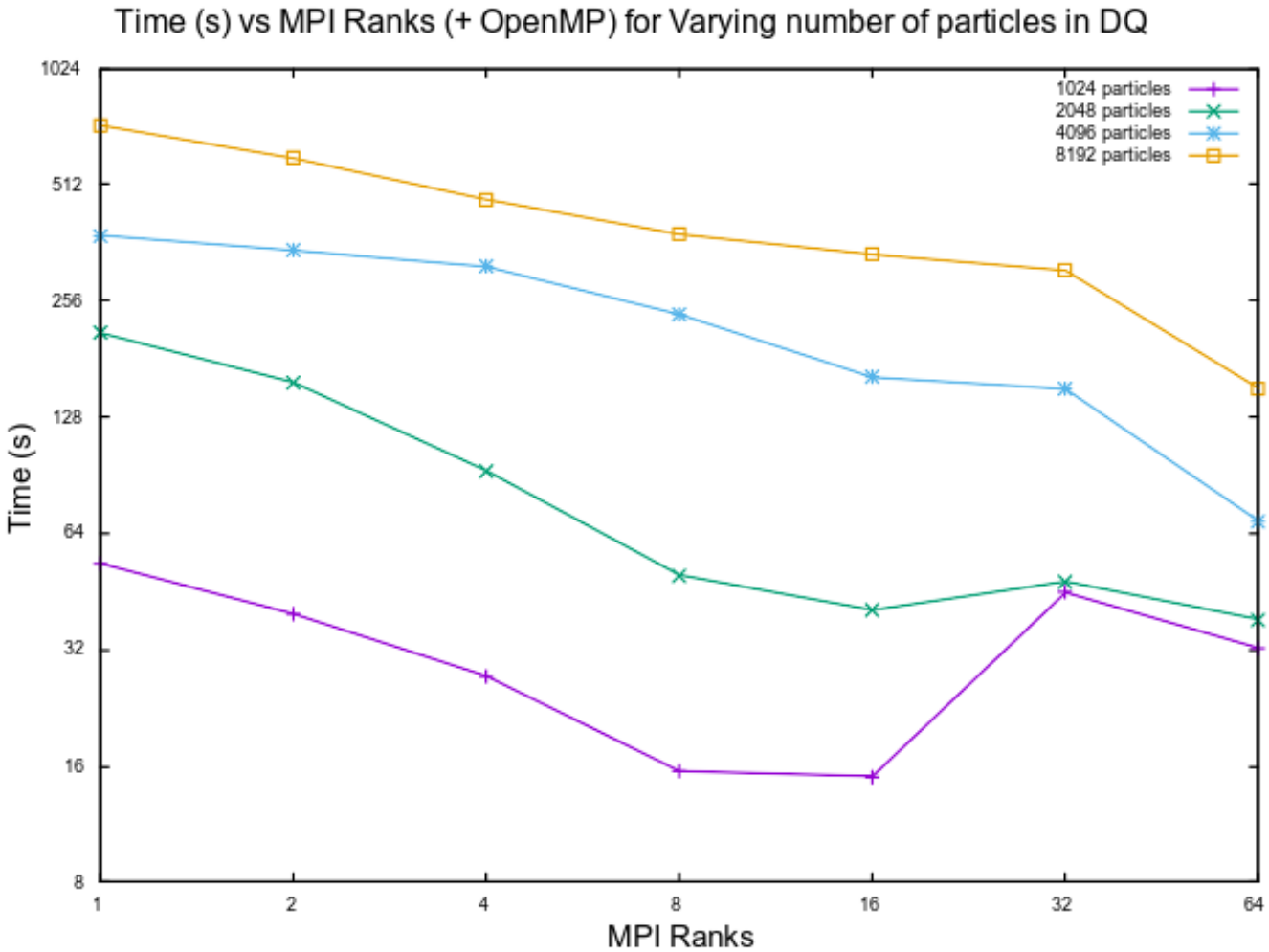
Thinking in parallel ...

- Then calculate acceleration, velocity and the position for the particle chunk for current time step
- Finally share the updated position data of the chunk of particles with all other processors

Results for MPI



Results for MPI + OpenMP



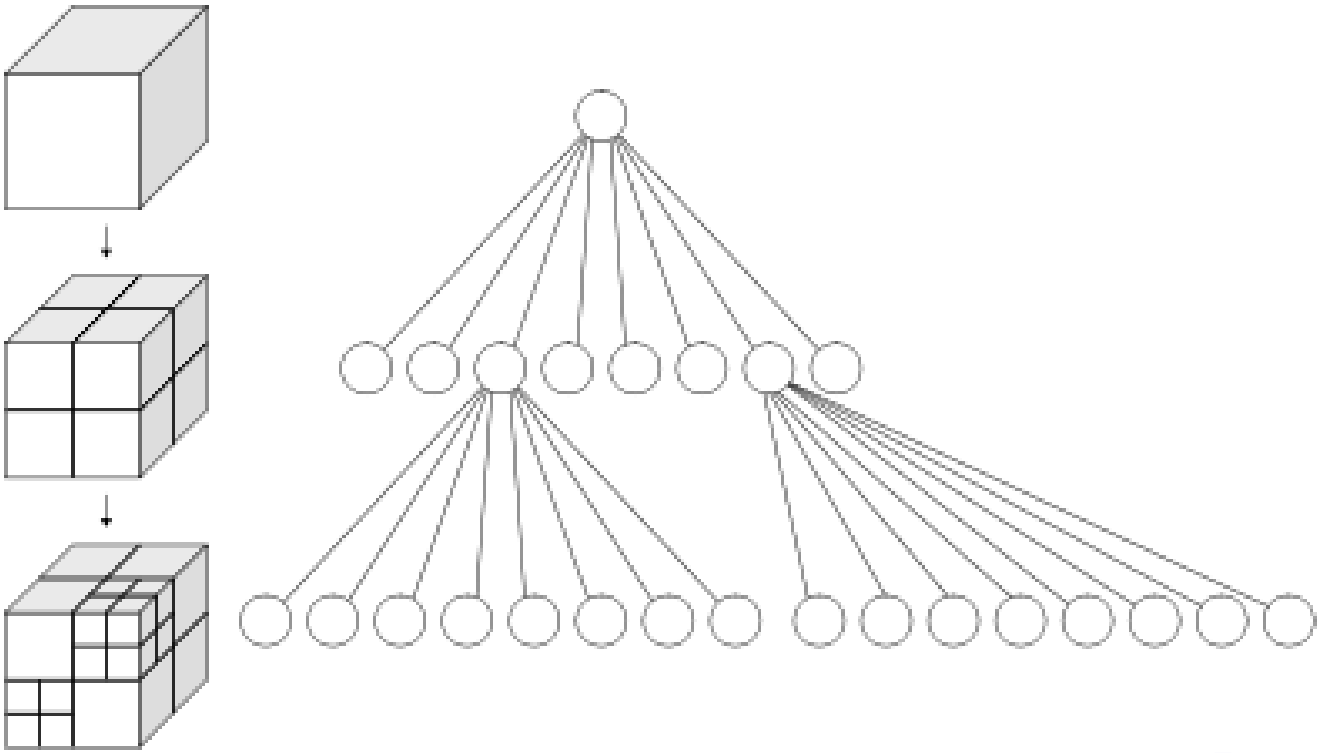
Barnes-Hut

- Force between particles scales as $\mathcal{O}\left(\frac{1}{r^2}\right)$
- Bodies that are far away from each other individually have less impact on each other
- Idea - Lump bodies that are far away beyond a threshold from a given particle

Combining particles

- Recursively divide the set of bodies into groups by storing them in an oct-tree
- An oct-tree is similar to a binary tree, except that each node has 8 children

- Space is recursively subdivided into octants until each subdivision contains 0 or 1 bodies
- Each node aggregates the group of bodies beneath it
- Stores the center-of-mass and the total mass of all bodies within its boundaries



Force calculation

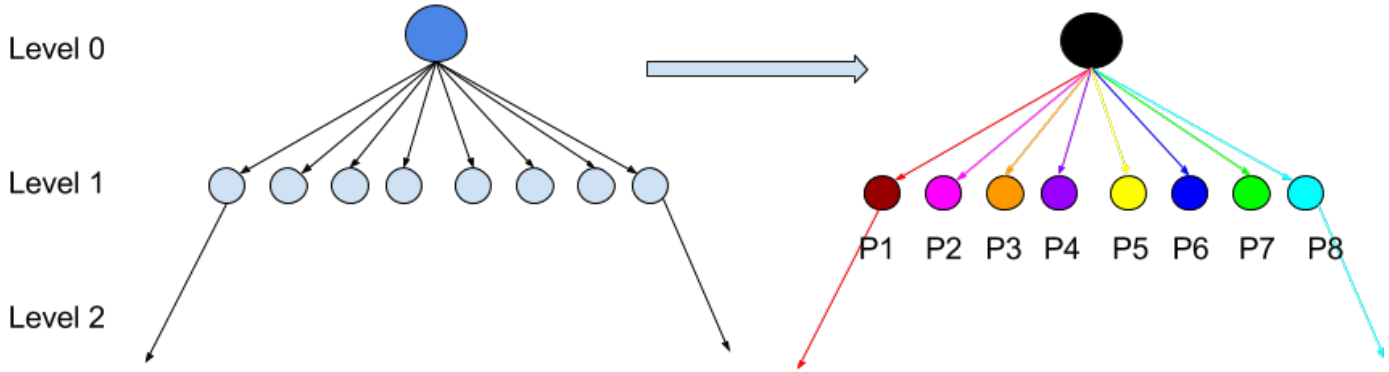
- If the group is sufficiently far, approximate its force contribution using a virtual body at the group's center of mass
- If the node is not sufficiently far from the body, then recursively traverse each of its subtrees.
- How to define what is far?

Determining closeness

- Calculate a ratio $\frac{s}{d}$
  - s is the mass of the region represented by the internal node
  - d is the distance between the body and the nodes center-of-mass
- If $\frac{s}{d} \leq \Theta$, then the node is far away.
- Θ can be adjusted for accuracy and speed

Octree in parallel

- All processors generate root node
- Processors separately generate child trees
- Merge child trees to create full tree
- Force calculation is embarrassingly parallel



Barnes-Hut results

...TBD

Thank You!