Parallel CSE Applications

Abhinav Bhatel, Department of Computer Science
Announcements

- Assignment 6 (CUDA) for extra credit due on Dec 6 11:59 pm
  - Please name tarball as LastName-FirstName-assign6.tar.gz
Molecular Dynamics
Molecular Dynamics

• Calculate trajectories of atoms and molecules by solving Newton’s equations of motions
Molecular Dynamics

- Calculate trajectories of atoms and molecules by solving Newton’s equations of motions
- Force calculations
  - Bonded interactions: bonds, angles, dihedrals
  - Non-bonded interactions: van der Waal’s and electrostatic forces
Molecular Dynamics

- Calculate trajectories of atoms and molecules by solving Newton’s equations of motions
- Force calculations
  - Bonded interactions: bonds, angles, dihedrals
  - Non-bonded interactions: van der Waal’s and electrostatic forces
- Number of atoms: thousands to millions
Molecular Dynamics

- Calculate trajectories of atoms and molecules by solving Newton’s equations of motions
- Force calculations
  - Bonded interactions: bonds, angles, dihedrals
  - Non-bonded interactions: van der Waal’s and electrostatic forces
- Number of atoms: thousands to millions
- Simulation step: ~1 femtosecond ($10^{-15}$ s)
Sequential Algorithm

- At every step, calculate forces on each atom
  - Calculate bonded and short-range forces every step
  - Calculate long-range non-bonded forces every few time steps (using PME or P3M etc.)
- Particle mesh Ewald (PME) summation:
  - Calculate long-range interactions in Fourier space
- Calculate velocities and new positions
- Repeat …
Traditional approaches to parallelization
Traditional approaches to parallelization

- Atom decomposition:
  - Partition the atoms across processes
Traditional approaches to parallelization

- Atom decomposition:
  - Partition the atoms across processes

- Force decomposition:
  - Distribute the force matrix to processes
  - Matrix is sparse and non-uniform
Traditional approaches to parallelization

- Atom decomposition:
  - Partition the atoms across processes

- Force decomposition:
  - Distribute the force matrix to processes
  - Matrix is sparse and non-uniform

- Spatial decomposition:
  - Assign a region of the 3D simulation space to each process
Hybrid parallelization

- Hybrid of spatial and force decomposition
- Decouple assignment of data and work to processes
- Distribute both atoms and the force calculations to different processes
Neutral territory (NT) methods

- Desmond’s mid-point method

![Diagram of Neutral territory (NT) methods](image)
Particle mesh Ewald

- Replace direct force calculations by:
  - Calculate short-range forces in real space
  - Calculate long-range forces in Fourier space
- Create a 3D mesh/grid representing charge densities of atoms
  - Compute a 3D Fast Fourier Transform (FFT)
- FFT computes the discrete Fourier transform (DFT) or inverse DFT
  - Reduces the complexity from $O(N^2)$ to $O(N \log N)$
Parallelization of PME (3D FFT)
Parallelization of PME (3D FFT)

- Bring all the data to one process
Parallelization of PME (3D FFT)

- Bring all the data to one process
- 1D or slab decomposition

### Diagram Description

- **Proc 0**
  - Problem size: $L \times M \times N$
  - Perform 1D-FFT along y-dimension and 2nd 1D-FFT along z-dimension (a)
- **Proc 1**
  - Receive data from Proc 0
  - Perform 1D-FFT along x-dimension (b)
- **Proc 2**
  - Receive data from Proc 0
  - Perform 1D-FFT along y-dimension (c)
- **Proc 3**
  - Receive data from Proc 0
  - Perform 1D-FFT along z-dimension (d)

**ALL to ALL**

- Data exchange:
  - Proc 0 to Proc 1
  - Proc 0 to Proc 2
  - Proc 0 to Proc 3
  - Proc 1 to Proc 2
  - Proc 1 to Proc 3
  - Proc 2 to Proc 1
  - Proc 2 to Proc 3
  - Proc 3 to Proc 1
  - Proc 3 to Proc 2

**To get data over x-dimension locally**

- Local computations within each process
Parallelization of PME (3D FFT)

- 2D or pencil decomposition
Measles killed 200,000 in 2020 alone!

Predictions say that 1.66 million died of tuberculosis in 2020

https://www.nature.com/articles/d41586-020-01011-6

https://www.nature.com/articles/d41586-020-02497-w
Measles killed 200,000 in 2020 alone!

START RECORDING

Data from March 2020.
* Suspected, not yet officially reported to WHO.

For AIDS, the latest estimates of the number of deaths in 2019 must have been published by UNAIDS and available at http://www.unaids.org. For TB, the estimates for 2019 are those published in the United Tuberculosis Report 2019. Deaths from TB among people living with HIV are officially classified as deaths caused by HIV/AIDS in the International Classification of Diseases.


https://www.nature.com/articles/d41586-020-01011-6

Predictions say that 1.66 million died of tuberculosis in 2020

https://www.nature.com/articles/d41586-020-02497-w
Societal challenge

• Controlling the spread of infectious diseases is important

• Computational and mathematical modeling of epidemics important to assist governments in responding to outbreaks

• Made challenging due to:
  • increased and denser urbanization
  • increased local and global travel
  • increasingly immuno-comprised population
Approach: individual-based simulation

- Agent-based modeling to simulate epidemic diffusion
- Models agents (people) and interactions between them
- People interact when they visit the same location at the same time
- These “interactions” between pairs of people are represented as “visits” to locations
- Use a bi-partite graph of people and locations or a people-people interactivity graph
Serial algorithm

- At each timestep (typically a day):
  - Determine which people visit which locations
  - “Send” people to those locations
  - At each location “interactions” happen and transmission happens
  - Update people’s states at the end of the day and continue

- Interventions (vaccinations, school closures) can be added on certain days to change people’s susceptibility, movements etc.
Combination of network theory and discrete-event simulations

- Hybrid time-stepped and discrete-event simulation
Combination of network theory and discrete-event simulations

- Hybrid time-stepped and discrete-event simulation

Contact model at each location

Combination of network theory and discrete-event simulations

- Hybrid time-stepped and discrete-event simulation
Combination of network theory and discrete-event simulations

- Hybrid time-stepped and discrete-event simulation
Combination of network theory and discrete-event simulations

- Hybrid time-stepped and discrete-event simulation

```python
while d < num_days:
    for each person:
        Send visit messages to locations

for each location:
    Process all visit messages
    Run discrete event simulation
    Send interaction messages

for each person:
    Process interactions
    Update disease state
```

---

**Contact model at each location**

**Model for human transmission**

**Disease model for each person**

https://sites.hms.harvard.edu/fs/special-edition-on-infectious-disease/2014/an-introduction-to-infectious-disease
Parallel simulation is challenging

- Size and scale of the social contact network (6 billion agents for a global simulation)
  - Unstructured networks and complicated dependencies lead to high communication cost
- Individuals and their behaviors are not identical
- Co-evolving epidemics, public policies and agent behaviors make it impossible to apply standard model reduction techniques
Parallel implementation: Loimos

- All the people and locations are distributed among all processes
- DES computation can be done locally in parallel
- Communication when sending visit and infection messages
- Uses Charm++, a message-driven model
Application software stack

- Parallel programming model / runtime:
  - MPI, OpenMP, Charm++, CUDA, …

- Libraries
  - Data and visualization libraries (mesh management, simulation output)
  - I/O libraries
  - Math/numerical libraries
  - Graph partitioning, load balancing …
Why use libraries?

• No need to reinvent the wheel
• Significant effort to write and optimize code
Popular Libraries

- Data/visualization and I/O libraries
  - I/O: HDF5, pNetCDF, Adios

- Numerical libraries:
  - Fast Fourier transforms: FFTW
  - Dense linear algebra: BLAS, LAPACK, Intel MKL
  - Solvers for sparse systems: Hypre, PETSc, Trilinos

- Graph partitioning/load balancing:
  - METIS, Scotch, Zoltan, Chaco

https://events.prace-ri.eu/event/176/contributions/38/attachments/154/305/HPC_libraries.pdf
Domain-specific languages/frameworks

- Structured grids: SAMRAI, Chombo, AMReX
- Unstructured grids: MFEM, Quinoa
The $n$-body problem

- Simulate the motion of celestial objects interacting with one another due to gravitational forces
- Naive algorithm: $O(n^2)$
  - Every body calculates forces pair-wise with every other body (particle)

Data distribution in $n$-body problems

- Naive approach: Assign $n/k$ particles to each process
- Other approaches?
Data distribution in $n$-body problems

- Naive approach: Assign $n/k$ particles to each process
- Other approaches?

http://datagenetics.com/blog/march222013/
https://en.wikipedia.org/wiki/Z-order_curve

Space-filling curves
Data distribution in $n$-body problems

- Naive approach: Assign $n/k$ particles to each process
- Other approaches?

Space-filling curves

http://datagenetics.com/blog/march222013/
https://en.wikipedia.org/wiki/Z-order_curve
Data distribution in $n$-body problems

- Naive approach: Assign $n/k$ particles to each process
- Other approaches?

Space-filling curves

http://datagenetics.com/blog/march222013/
https://en.wikipedia.org/wiki/Z-order_curve

Data distribution in \( n \)-body problems

- Let us consider a two-dimensional space with bodies/particles in it
Data distribution in $n$-body problems

- Let us consider a two-dimensional space with bodies/particles in it
Data distribution in \( n \)-body problems

- Let us consider a two-dimensional space with bodies/particles in it
Different parallelization methods

- Tree codes: Barnes-Hut simulations
- Fast multipole methods (FMM): Greengard and Rokhlin
- Particle mesh methods
- Particle-particle particle-mesh (P³M) methods
Barnes-Hut simulation

- Represent the space containing the particles as an oct-tree
- Pairwise force calculations for nearby particles
- For tree nodes that are sufficiently far away, approximate the particles in the node by a single large particle at the center of mass

Fast multipole methods

- Use multipole expansion for distant particles
- Takes advantage of the fact that for nearby particles, multipole-expanded forces from distant particles are similar
- Reduces the time complexity further to $O(n)$
Particle-particle particle-mesh methods

• Explicit calculation of forces on nearby particles
• Fourier-based Ewald summation for calculating potentials on a grid
• Smoothed particle hydrodynamics