CMSC 714
Lecture 23
Anton and the Virtual Microscope

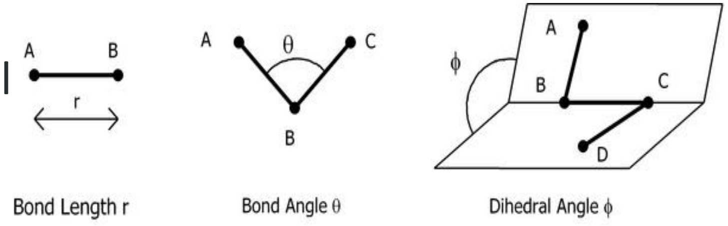
Alan Sussman

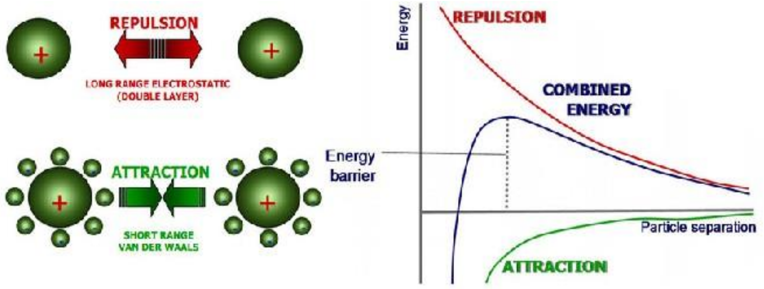
Notes

- Group Project presentations scheduled for next week on May 5 and 5 – date for each group posted on Readings web page
  - final report due Monday, May 11
- Course evaluation (Student Feedback on Course Experiences) web site open
  - https://www.courseexp.umd.edu

Anton & Molecular Dynamics

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



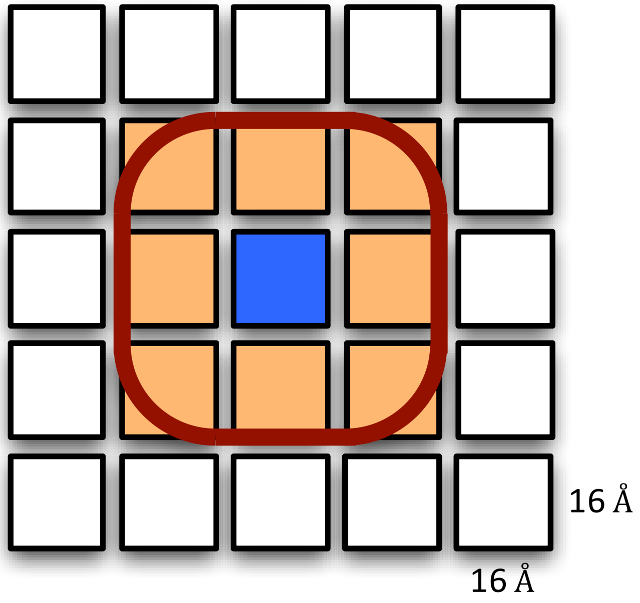


Sequential Algorithm

- 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

- 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

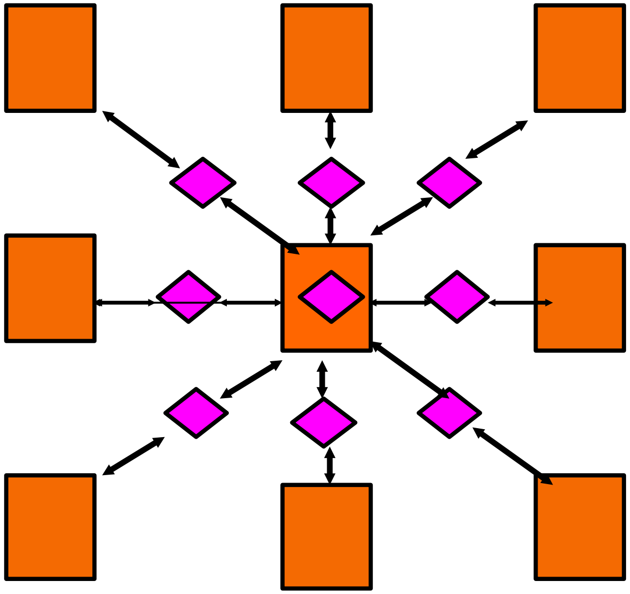


Hybrid parallelization

- Hybrid of spatial and force decomposition

- Decouple assignment of data and work to processes

- Distribute both atoms and the force calculations to different processes



Anton

- Special purpose machine built for molecular dynamics simulations, built at D.E. Shaw Research
  - to simulate biological processes that occur on very small time scales (10^{-15} sec), such as protein folding, interaction between proteins, etc.
  - and simulate those processes for a long time
- Molecular dynamics
  - force calculation followed by integration step to move particles
  - biomolecular forces have 3 parts
    - bonded forces – small atomic groups with covalent bonds
    - van der Waals forces – all pairs of atoms, but fall off quickly with distance (so only need close ones)
    - electrostatic forces – all pairs of atoms, fall off slowly with distance – divide into 2 parts to avoid all pairs computation

Anton

- Anton machine
  - up to 512 nodes in 8x8x8 torus
  - each node has 2 parts on 1 chip
    - high throughput interaction subsystem (HTIS) for range-limited interactions , using 32 hardwired pairwise point interaction pipelines (PPIPs)
    - flexible subsystem with 8 programmable geometry cores (GCs) for less structured part of MD computation, 4 Tensilica processors, 4 data transfer engines
    - plus DRAM controllers, 6 network interfaces, and host interface for I/O
- Most of computational time mapped to PPIPs, which run those computations maybe 100x faster than standard microprocessor core
- And computations spatially decomposed across nodes, with some twists to deal with communication as particles move between spatial domains
- Uses fixed-point arithmetic, with various bit widths, for several reasons:
  - performance – fixed-point hardware fast and small
  - fixed point arithmetic is truly associative
  - gain determinism – run same simulation again get exact same results bit-for-bit (doesn’t really help, since MD is a chaotic system, so need ensemble)
  - computations are reversible

Anton

- Performance results show can run a large chemical system at much higher rates than any previous system
  - can run multiple microseconds of simulation time per day of wall clock time
    - maybe 500 times faster than 512 node Intel Xeon cluster
    - and have run simulated systems up to over 1000 microseconds, which showed interesting behavior of the molecules
  - and results are validated very well
    - both against “known” results and using statistical error tests
- Two later generations of Anton
  - Anton 2 in 2014, Anton 3 in 2022
  - Bigger, better, faster …

Virtual Microscope

- Software emulation of a light microscope, to view and manipulate very large slide images, built at UMD and Johns Hopkins med school
  - for viewing and processing images captured from standard pathology specimens (need special purpose hardware for high throughput data capture)
  - problem is very large data sizes
    - a slide is maybe 30K pixels on a side at high resolution, so one focal plane is maybe 10GB uncompressed
    - and need multiple focal planes for some samples
    - and JHU hospital produces >400K slides per year (in 1998!)
- Client/server system design
  - client runs on user desktop machine – Java GUI
  - server stores, retrieves, processes slide image data on parallel machine or workstation cluster
    - implemented both with Active Data Repository OO framework and with DataCutter component framework

Virtual Microscope

- Client provides drag/zoom interface to browse through a slide
  - use thumbnail to keep track of where you are on a slide
  - standalone client can cache image data for improved response time – using both memory and disk on client machine
- Server basic computation is map-reduce
  - map one or more input pixels at highest resolution to desired output resolution, and aggregate if multiple pixels map to same output pixel
- Active Data Repository
  - user defined functions used for map and reduce, framework orchestrates parallel execution across data stored on multiple nodes of a cluster or parallel machine
  - data blocks distributed across disks for parallel access and are indexed for fast retrieval (more important for more complex map functions)
  - images also need to be decompressed from stored JPEG form before map and reduce steps, and clipped to query window
  - experiments show that ADR implementation scales well, to handle multiple clients, with low overhead

Virtual Microscope

- DataCutter
  - component framework for processing large datasets in a distributed environment
  - filter-stream programming model
    - each filter is a component, and filters connected via streams, which deliver data buffers between filters
  - supports flexible placement of filters, filter replication for load balancing (transparent copies)
  - VM filter pipeline is: read-data, decompress, clip, zoom, view
- Performance results show that DataCutter implementation deals better than ADR with load balance issues, but ADR can process large queries faster from parallel execution of a single query
  - for DataCutter, filter placement matters – communication between filters adds latency if on different hosts
- Overall performance results for VM show that can achieve interactive response times for real slide data, on not-too-large server system configurations