ALGORITHM IMPROVEMENTS AND HPC

Basic MD algorithm

1. Initialize atoms' status and the LJ potential table;
2. set parameters controlling the simulation; O(N)

For all time steps do

1. Update positions of all atoms; O(N)

If there are atoms who have moved too much, do

2. Update the neighbour list, including all atom pairs that are within a distance range (half distance matrix computation); **$O(N^2)$**

End if

3. Make use of the neighbour list to compute forces acted on all atoms; O(N)
4. Update velocities of all atoms; O(N)
5. Update the displace list, which contains the displacements of atoms; O(N)
6. Accumulate and output target statistics of each time step; O(N)

End for

Some general thoughs

- MD basic algorithm has not changed the last 40 years
- Most major MD codes use similar approaches for optimization
- All trivial tricks were already applied when computers were slow!!
- No major breakthrough except perhaps GPU's and Anton
  - MD improvement is largely ligated to the improvement of hardware resources
- Can we use efficiently a present supercomputer for MD?

Improvement directions

- Floating-point calculations
  - Precision, Custom Maths.
  - Accurancy *vs.* time. MD is a stochastic process
- Non-Bonded interactions
  - PBC, Common interactions
- Stream processing & GPUs!!
- Integration steps
  - Multiple time steps, Constrains, Virtual interaction sites
- Parallelization. Spatial decomposition
- Asynchronous parallelization or even distributed computing
  - Replica exchange, Markov state models

P. Larsson, B. Hess & E. Lindahl. WIRE Comp. Mol. Sci. 2011, 1, 93-108

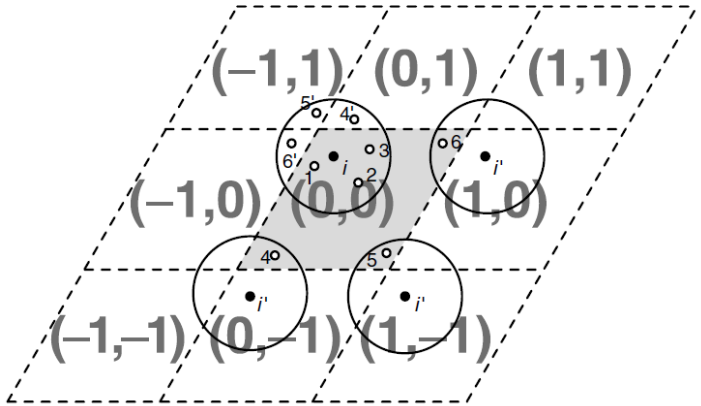
Floating point calculations

- Move to simple precision
  - Problem for accumulated numerical error or for too small position displacements
  - However, force calculations require only a relative error < 10^{-3}
  - Small effect in the calculation itself in modern processors
  - Improves use cache, SIMD registers
  - Math routines in single precision are more efficient

- Custom Math routines
  - MD does not require Maths to be IEEE complain!!
  - Some internal checks can be avoided (ex. positive parameter in sqrt)

Non-bonded interactions

- Avoid **O(N²)**
  - Maintain list of NB pairs within a distance cutoff
    - Updated at larger time steps
    - Caution, too short cutoff distance give wrong behavior
  - Domain decomposition (more later)
  - PBC requires to check for the closest image
    - Simple approach with 6 conditionals kill modern processors
    - Cut-off sphere partitioning method
      - Maintains information about all 26 copies of the particle, updated with the NB list.
  - This simple trick could give 10-50x speedup



Specialized interactions

- Some interactions are more abundant than others
  - WAT-WAT dominate NB interactions list
  - Water models
    - SPC, TIP3P, TIP4P
    - Rigid, only VdW in O, reduced set of charges
  - The number of interactions to check is largely reduced
  - Only oxygens should be considered in the NB Lists

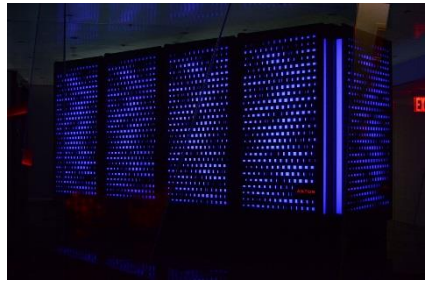
- A Charmm implementation uses a combined Coulomb + Lennard-Jones potential for Wat-Wat interactions

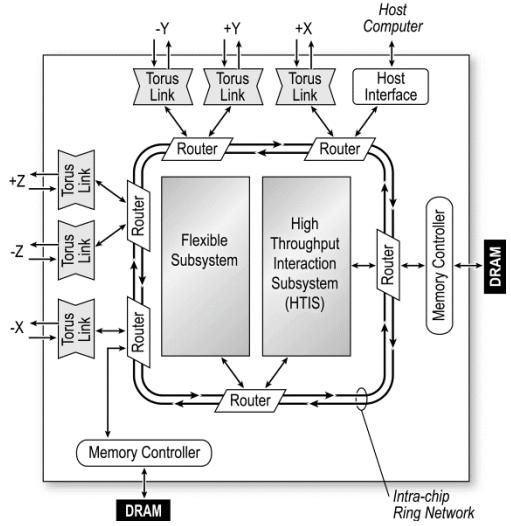
HW optimization: Streaming, GPUs, ASICs

- GPUs have become popular in simulation field.
  - One order of magnitude faster that comparable CPUs
  - Requires reformulate algorithms
  - Usually the strategy is not compatible with others
  - SIMD approach
  - No random access to memory, no persistent data

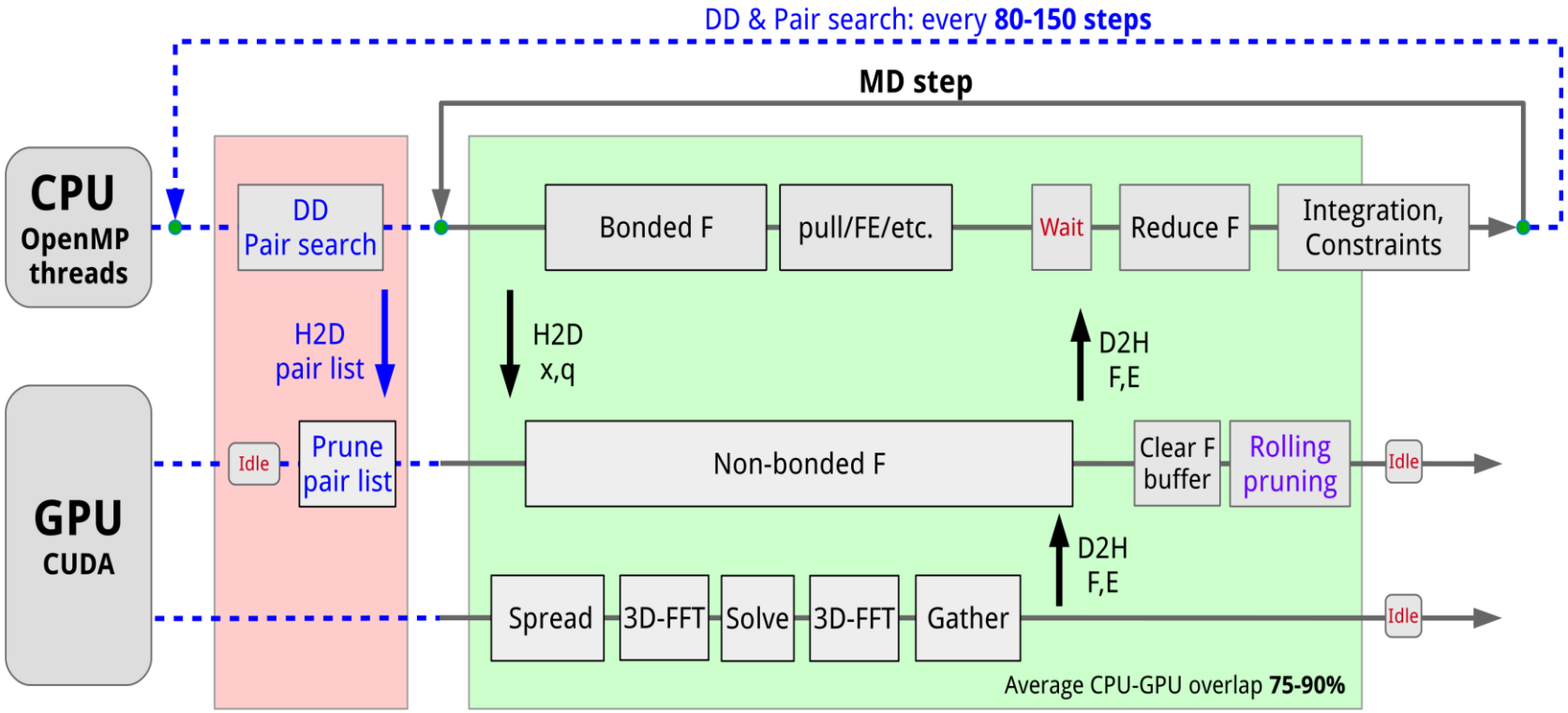
- CPU is responsible of the main algorithm, GPU performs selected parts
  - Update the NB list, Evaluate interactions, Update coordinates & velocities
  - However, AMBER new versions move everything to GPU!

- ASIC (application specific processors)
  - ANTON (DE Shaw Research)
    - The fastest ever MD computer (3 orders of magnitude above)





Heterogenous parallelism in GROMACS

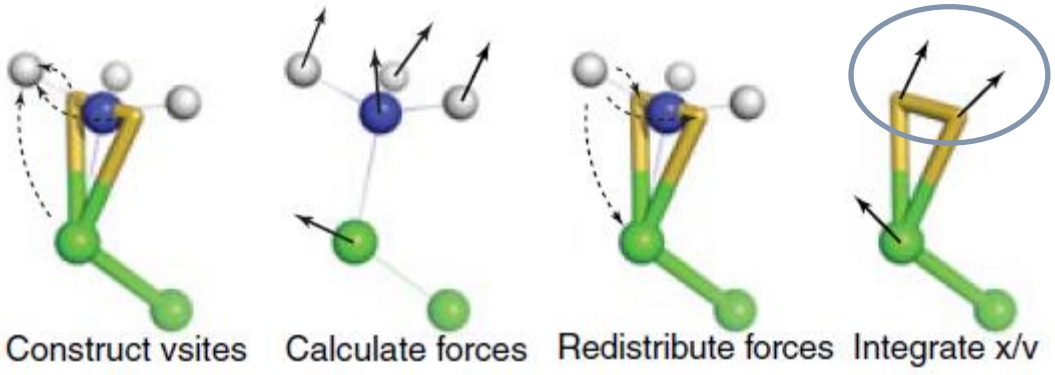


Integration steps

- Multiple time steps
  - Numerical integrations require that the time step is kept below the fastest movement in the system
    - Normally 1 femtosecond (10^9 iterations to reach biologically meaningful time scales)
  - Most parts of the calculation do not change significatively in 1 fs.
    - GROMACS style:
      - 1fs bond vibrations, 2fs NB interactions, 4fs PME (long range interactions)
      - Tenths fs (NB list update, temperature and pressure update)
    - Increasing the granularity of the minimal calculation (ex. in coarse-grained) will allow longer time steps

Integration steps. Constrains

- Constrains
  - Some movements are not relevant for global flexibility
  - Algorithms like SHAKE, RATTLE, M-SHAKE, LINCS, update bond distances by direct minimization
- Constrains are very efective but limit parallelization
  - Usually restricted to bonds with hydrogen atoms
- Virtual interaction sites
  - Hydrogen atoms are replaced by virtual atoms
  - Possible to reach 4-5 fs

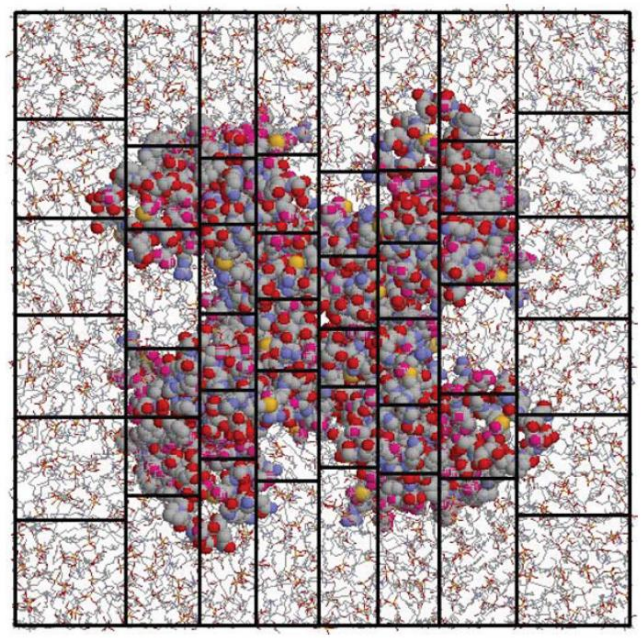


Optimization tricks: Parallelization

- MD parallelization uses MPI
  - Heavily limited by communication bandwidth
  - Modern multicore processors will allow to combine with openMP

- Simplest strategy is Atoms/Force decomposition
  - Atoms are distributed among processing units
  - Bad and unpredictable load balance
  - Requires all-to-all update of coordinates
  - Partially improved by distribution NB list instead of atoms
  - Scales only up to 8-16 processors

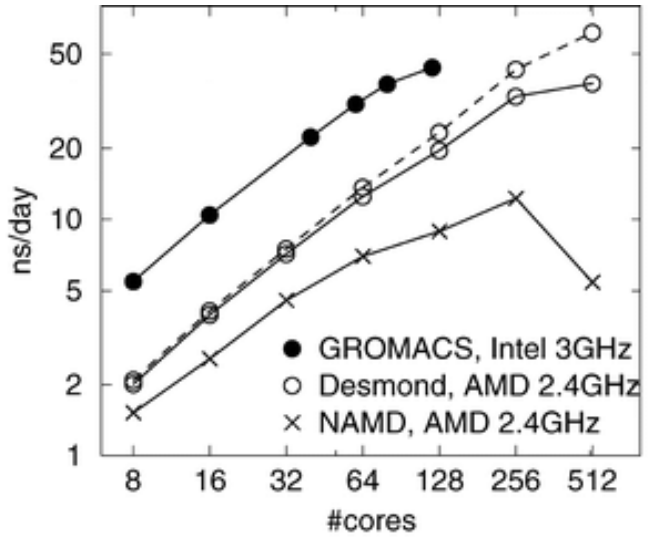
- Distribution should be dynamic: Spatial decomposition (the *Air traffic controller* way)
  - Space regions (not atoms) are distributed among processing units
  - Assignment of atoms is dynamic, depending on their position in space
  - Load Balancing is the major bottleneck

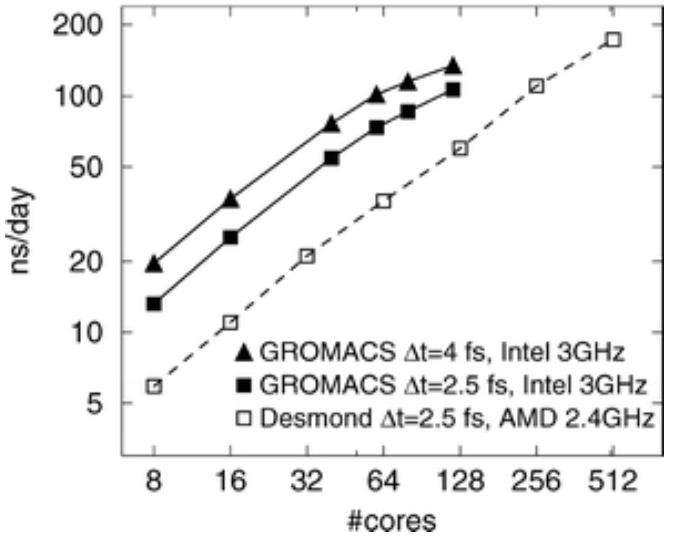


Asyncronous parallelization

- Any parallelization method become useless with less of ca. 100 atoms per processing unit
  - Present computers can have thousands of cores but most usual simulations have around 100,000 atoms !!!

- A single long simulation can be replaced by a series of shorter ones.
  - Same statistics
  - Sampling is improved as initial structures could be different
  - In most cases simulations are completely independent (embarrisingly parallel)
  - A single supercomputer is not really needed
    - Distributed computing
    - Grid
    - Folding@home





Conclusions

- Major MD codes uses a combination of optimization methods
- Current routine tends to be GPU based calculations, but combinations MPI + openMP are being considered.
  - GPU + MPI + … approaches are being considered but are highly dependent on hardware evolution.
- No parallelization schema will survive next generation of supercomputers (i.e. no “exascale md”)

MD & HPC future?

"An ideal approach would for instance be to run a simulation of an ion channel system over **1000 cores** in parallel, employing, e.g., **1000 independent such simulation**s for kinetic clustering, and finally perhaps use this to screen **thousands of ligands or mutations** in parallel (each setup using 1,000,000 cores)"

P. Larsson, B. Hess & E. Lindahl. WIRE Comp. Mol. Sci. 2011, 1, 93-108