Scientific Computing II

Molecular Dynamics Simulation (3)

Michael Bader – SCCS
Summer Term 2013
MD – Implementational Aspects
  Verlet Neighbour Lists
  Linked-Cell Algorithm
  Linked-Cell – Data Structures

MD – Parallelisation
  Shared Memory Parallelisation
  Force Decomposition
  Spatial Decomposition
  Domain Decomposition
Verlet Neighbour Lists

- every molecule stores its neighbours for a distance $r_{\text{max}} > r_c$
- every $n_{\text{upd}}$ time steps (dep. on $r_{\text{max}}$), the lists are updated
- the ”buffer” has to be larger than the covered distance of a molecule for that time:

$$r_{\text{max}} - r_c > n_{\text{upd}} \Delta t v_m$$
Classical Linked-Cell Algorithm

molecules arranged in a lattice of cubic cells of side length $\sim r_c$

- hash table with "geometrically motivated" hash function
- "Binning" resp. "Bucketing"-techniques from "Computational Geometry"
- direct volume representation (voxel) of the influence region
Classical Linked-Cell Algorithm

- molecules arranged in a lattice of cubic cells of side length $\sim r_c$
- runtime: $\mathcal{O}(n)$
- only half (point symmetry) of the neighbour cells are explicitly traversed (Newton’s 3rd law)
- erase and generate the data structure in each time step
Variable Linked-Cell Algorithm

- lattice might be built up from cells of side length $\frac{r_c}{t}$ with $t \in \mathbb{R}^+$
- integer numbers are preferable for the divisor $t \in \mathbb{N}$
- for $t \rightarrow \infty$, the examined influence volume will converge to the (optimal) sphere
Linked-Cell Algorithm – Data Structure I

- cells are stored as a one-dimensional array (vector)
- intrusive list for the cell molecules
- list to determine the processing sequence
• offset mask to determine the neighbours
• cache efficiency is influenced by the processing order (temporal locality)
Profiling seq. MD code (LJ 12-6)

force calculation is dominating
Shared Memory Parallelisation

- each process calculates one part \( \frac{N}{p} \) of the molecules (cells)
- availability of all relevant data (position) because of common memory
- “Shared Memory” algorithm: Velocity Störmer Verlet method
  1. parallel explicit Euler method \( r, v \) (half step) for \( \frac{N}{p} \) molecules
  2. parallel force calculations for \( \frac{N}{p} \) molecules or the respective cells
     (force summation critical, respecting Newtons 3rd law: reduction; same with linked-cell algorithm)
  3. parallel implicit Euler method \( v \) (half step) for \( \frac{N}{p} \) molecules
Replicated Data I

- use shared-memory approach also for distributed memory architectures
- every node has to store all position data
- collective communication for the synchronization of redundant data
- result: ”Atom Decomposition” algorithm:
  ➔ Velocity Störmer Verlet method
  1. explicit Euler method $r,v$ (half step) for $\frac{N}{p}$ molecules
  2. distribute (scatter-to-all) the $\frac{N}{p}$ position data for each PE to all other PEs
  3. force calculation for $\frac{N}{p}$ molecules
  4. possible distribution of partial forces to the appropriate PEs
  5. implicit Euler method for $v$ (half step) for $\frac{N}{p}$ molecules
Replicated Data II

- costs
  - calculation: \( \frac{N}{p} \)
  - communication partners per PE: \( p - 1 \)
  - memory requirements: \( N \) positions and \( \frac{N}{p} \) forces
Replicated Data II – Exploit Symmetry

- calculation: \( \frac{N}{2p} \)
- communication partners per PE: \( p - 1 \)
- memory requirements: taking advantage of Newtons 3rd law needs a vector for \( N \) (partial) forces and additional communication
Force Decomposition I

- each process calculates a part of the molecules and the force matrix
- on each node: position data of $2 \frac{N}{\sqrt{p}}$ molecules
- communication: distribution of positions and calculated forces
- ”Force Decomposition” algorithm: Velocity Störmer Verlet method
  1. explicit Euler method for $r, v$ (half step) for $\frac{N}{p}$ molecules
  2. distribution of $\frac{N}{p}$ position data per PE to $2(\sqrt{p} - 1)$ PEs
  3. force calculation of a $\frac{N}{\sqrt{p}} \times \frac{N}{\sqrt{p}}$ sub-matrix
  4. distribution of partial forces to $\sqrt{p} - 1$ PEs
  5. implicit Euler method for $v$ (half step) for $\frac{N}{p}$ molecules
Force Decomposition II – Costs

- calculation: \( \frac{N}{p} \)
- communication partners per PE: \( 2 (\sqrt{p} - 1) \)
Spatial Decomposition

- domain is decomposed into subdomains
- each processor handles one subdomain
- amount of molecules per processor is variable (molecules are moving!)
- overlapping buffer regions (halo, $r_c$) have to be synchronized
Spatial Decomposition

- overlapping buffer regions (halo, $r_c$) have to be synchronized
- point-to-point communication, dependent of
  - decomposition
  - molecule movement (flow velocity)
  - communication method: "x-y-z" vs. "direct"
Domain Decomposition: Cubes ↔ Slices (1)

- assumption:
  - homogeneous molecule distribution
  - subdomains with $\frac{N}{p}$ molecules and volume $\frac{L^d}{p}$: $N \leftrightarrow L^d$
  - communication size proportional to halo volume

- slices:
  - 2 neighbour PEs
  - halo volume: $L^{d-1}2r_c = 2L^d \frac{r_c}{L}$
  - relatively easy to implement
  - bad scaling properties

- communication
  - amount: $2p$
  - ratio molecules to comm.: $\frac{L^d}{p} : 2L^d \frac{r_c}{L} = \frac{L}{2r_cp}$
Domain Decomposition: Cubes ↔ Slices (2)

- same assumptions
- cubes:
  - $3^d - 1$ neighbour PEs
  - side length: $l = \sqrt[3d]{\frac{L}{p}} = \frac{L}{d^{\frac{1}{d}}p}$
  - halo volume:
    $$\begin{align*}
    (l + 2r_c)^d - l^d \\
    = \sum_{i=1}^{d} \binom{d}{i} l^{d-i} (2r_c)^i \\
    \approx \binom{d}{1} l^{d-1} 2r_c \\
    = 2d l^{d} \frac{r_c}{l} = 2d L^d p^{\frac{1}{d} - 1} \frac{r_c}{L}
    \end{align*}$$
- communication
  - amount: $(3^d - 1) p$ (direct) or $2d \cdot p$ (x-y-z)
  - ratio molecules to comm.: $$\frac{L^d}{p} \frac{p^{2L}}{2d L^d p^{\frac{1}{d} - 1} \frac{r_c}{L}} = \frac{L^d}{2d p^{\frac{1}{d} - 1} \frac{r_c}{L}}$$