Parallel Implementation of a Monte Carlo Molecular Simulation Program

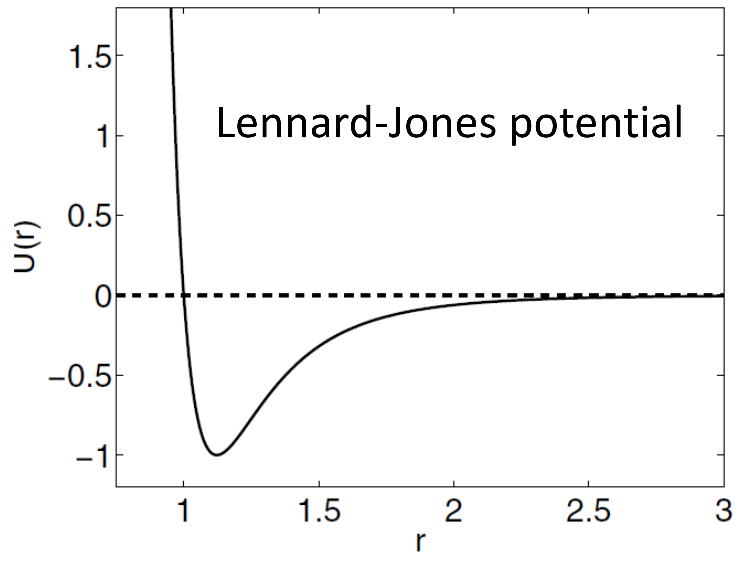
Milan Žeželj and Vladimir Slavnić
Scientific Computing Laboratory
Institute of Physics Belgrade
Serbia
e-mail: milan.zezelj@scl.rs

WWW.SCL.RS

SCIENTIFIC COMPUTING LABORATORY

Monte Carlo molecular method

- Classical approach
- Lennard-Jones potential between two particles
- N particles in the system
- domain of the simulation are box
- periodic boundary conditions
- we want to know energy, pressure, density…



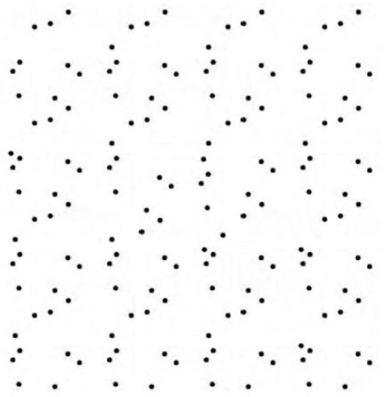
Monte Carlo:
1) Random move the particle
2) If the move is downhill in energy the new state is accepted
3) If the move is uphill in energy the new state is accepted with some probability (Metropolis algorithm)

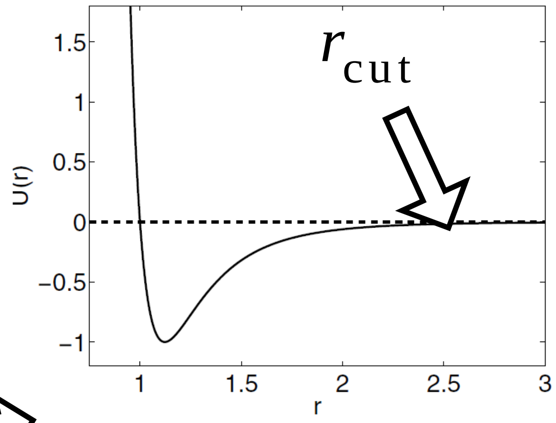
Monte Carlo molecular method is suitable for modeling equilibrium state

Algorithm

$O\left(N^2\right)$ operations to compute energy in this way:

$$V(\mathbf{x}_1,\ldots,\mathbf{x}_N) = \sum_{i=1}^{N}\sum_{j=1,j>i}^{N} U(r_{ij})$$

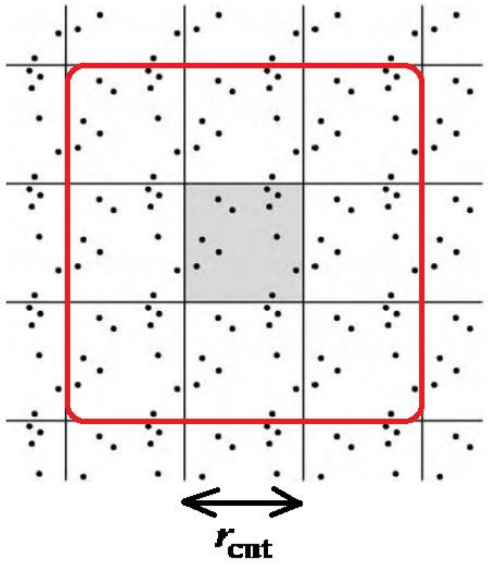




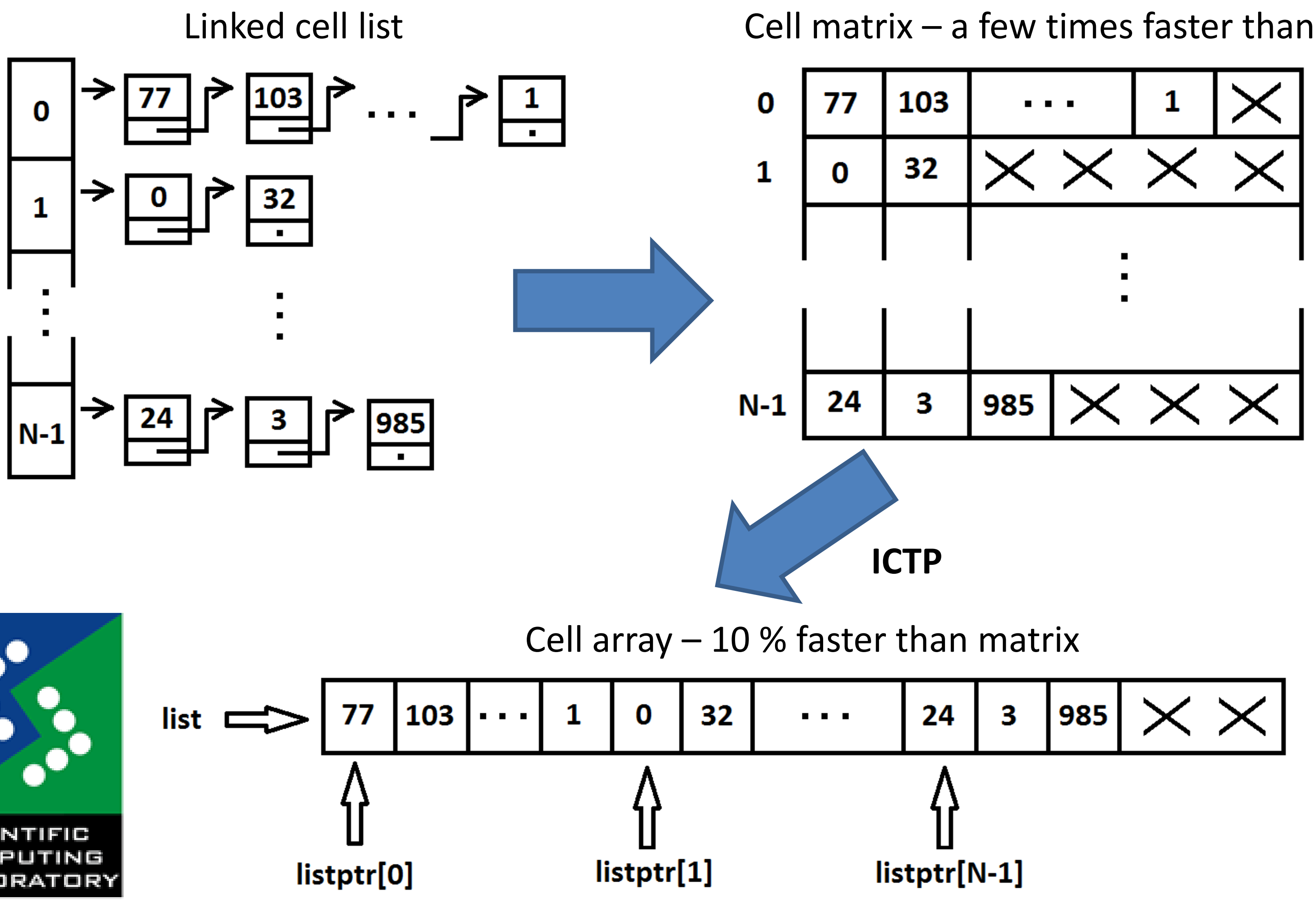
Short-range potential

Simulation box could be divided on M x M x M cells with length larger then r_{cut}

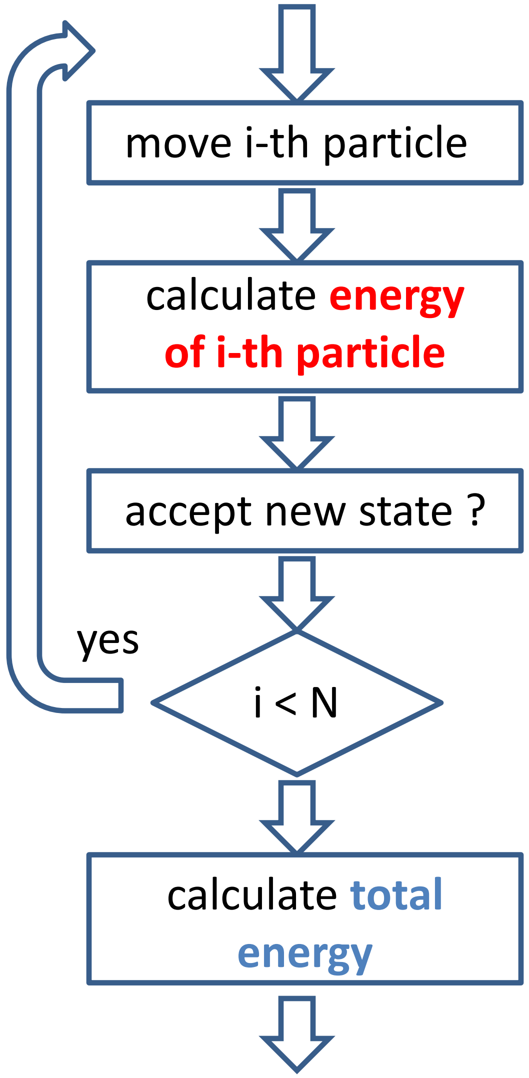
Each particle only interacts with particles in neighbor cells



Improvements of data access patterns



Code profiling

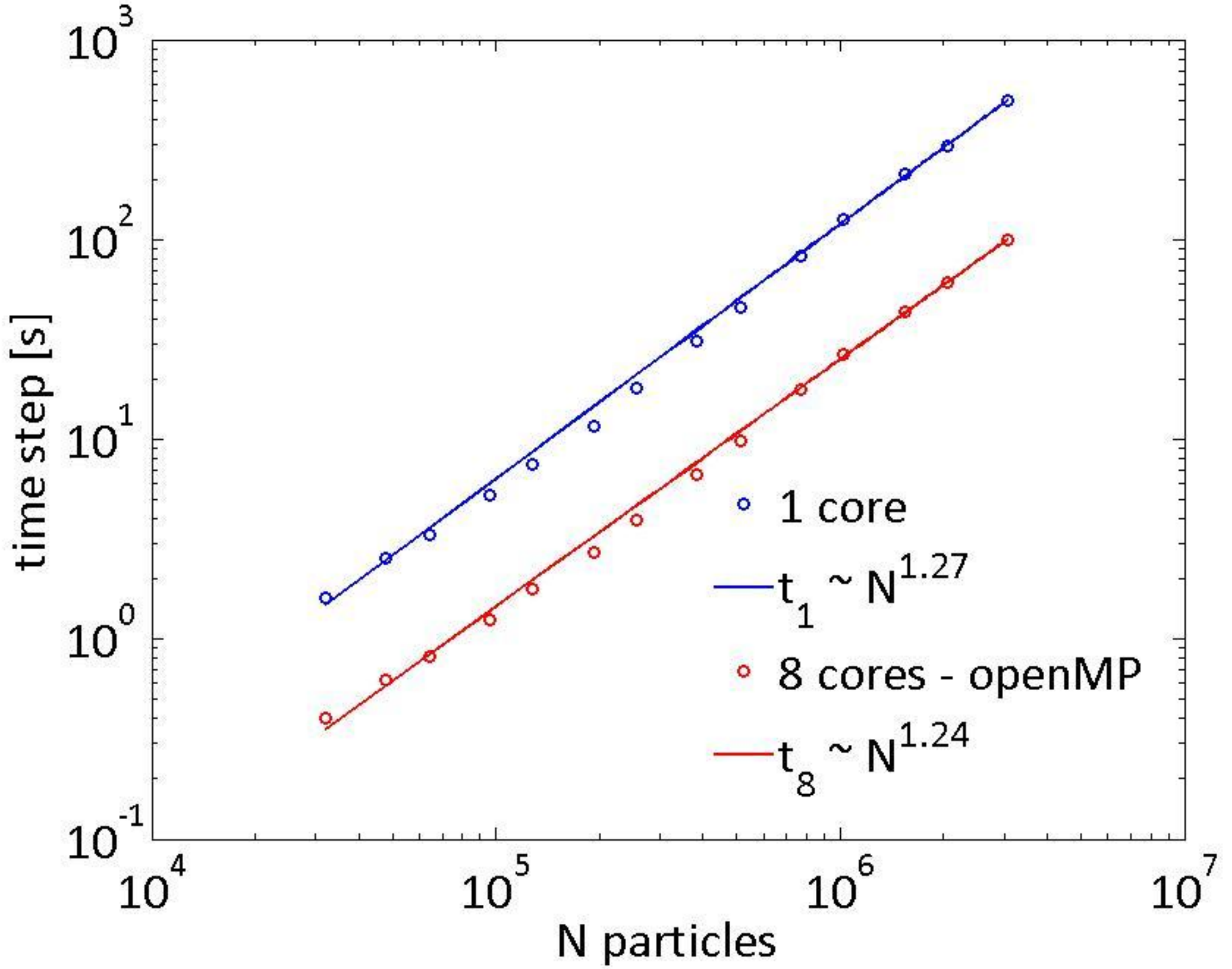


45% of running time used by function **energy of i-th particle**

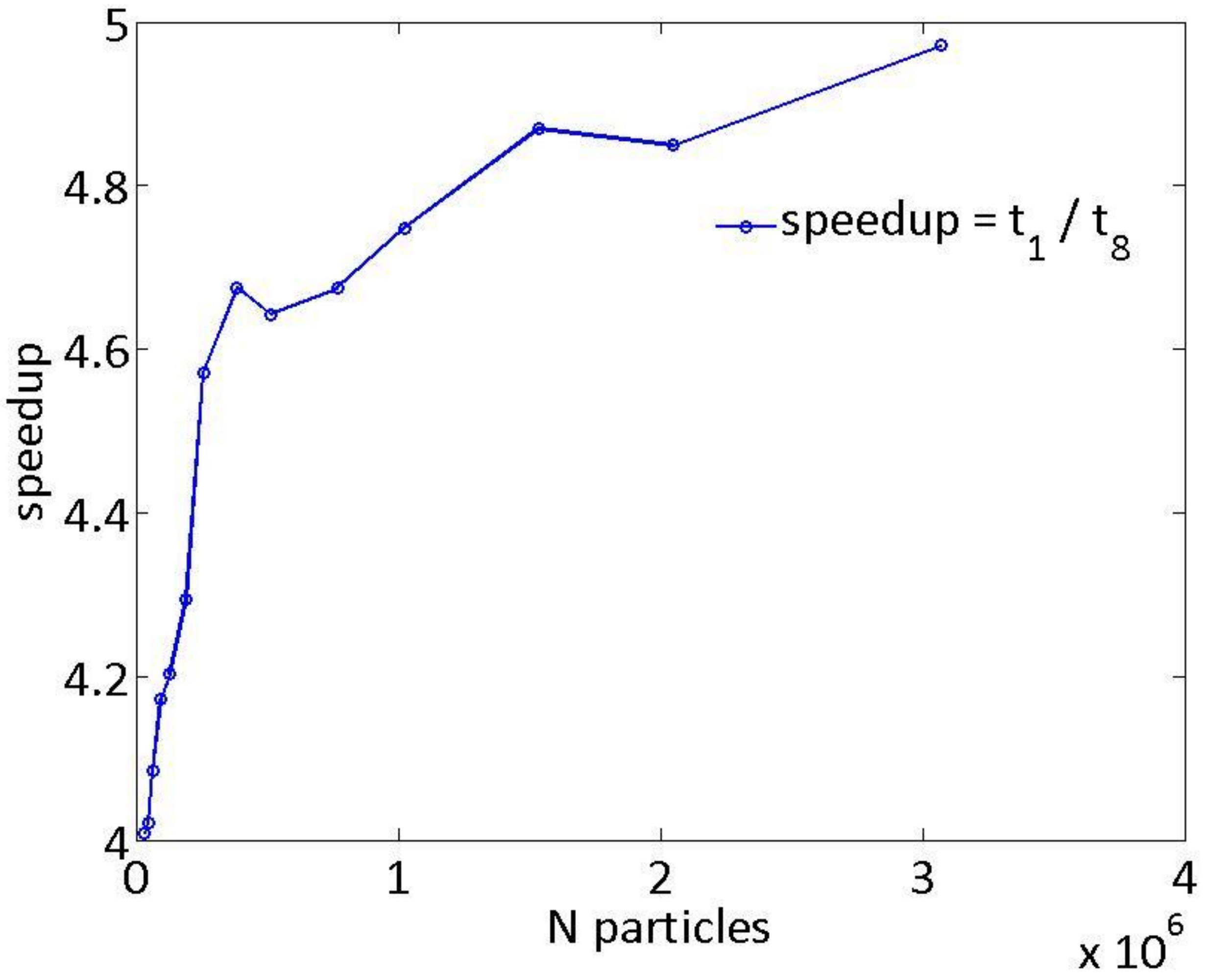
28% of running time used by function **total energy**

Implement **openMP** in these function

Single core vs. eight core openMP



Speedup



Conclusions

- openMP gives speedup up to 5 times using 8 cores

- fast optimization doesn’t failure the results and gives additionally speedup of 10%

Further work:

MPI + openMP

GPU

Thank you!