Development of GENESIS on K for large scale molecular dynamics simulation

Jaewoon Jung
(RIKEN Advanced Institute for Computational Science)

Importance of molecular dynamics

1. MD is a sampling method : We need ensembles because proteins constantly sample configuration space

2. Unlike other sampling method (i.e. Monte Carlo), MD gives dynamics

3. In particular, MD helps us to understand
1) Kinetics – chemical reaction (with QM), conformational changes, and protein folding…
2) Equilibrium Ensemble sampling – thermodynamics (free energy changes, bindings,..)
3) Modeling is possible – structure prediction, solvent effects…)

For theses purpose, we need accurate potential from with efficient computational algorithms

Fast non-bonded interaction
- New lookup table approach

Molecular Mechanics (Force field)

1. The basic functional form of a force field encapsulates both bonded terms relating to atoms that are linked by covalent bonds, and non-bonded (also called "noncovalent") terms describing the long-range electrostatic and van der Waals forces

$$E_{total} = E_{bonded} + E_{nonbonded}$$

$$E_{bonded} = E_{bond} + E_{angle} + E_{dihedral}$$

$$E_{nonbonded} = E_{vanderWalls} + E_{electrostatic}$$

2. A force field defines a set of parameters for each type of atom.

$$E_{total} = \sum_{bonds} \frac{1}{2} k_b (b - b_0)^2 + \sum_{angles} \frac{1}{2} k_a (\theta - \theta_0)^2 + \sum_{torsions} \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)] + \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} \left\{ \varepsilon_{ij} \left[\left(\frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{0ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\varepsilon r_{ij}} \right\}$$

   Parameters k_b, b_0, k_a, θ_0, V_n,γ, ε_{ij}, r_{0ij}, and q_i are defined according to each atom type.

3. The bottleneck of the computation is the non-bond energy calculation due to $O(N^2)$ computation steps for N particle.

How to calculate non-bond energy efficiently?

Cutoff for non-bond energy calculation

1. Non-bond energy calculation is reduced by introducing cutoff

$$\sum_{j=1}^{N-1}\sum_{i=j+1}^{N}\left\{\varepsilon_{ij}\left[\left(\frac{r_{0ij}}{r_{ij}}\right)^{12}-2\left(\frac{r_{0ij}}{r_{ij}}\right)^{6}\right]+\frac{q_iq_j}{4\pi\varepsilon r_{ij}}\right\}$$

$O(N^2)$

$$\sum_{|i-j|<R}^{N}\left\{\varepsilon_{ij}\left[\left(\frac{r_{0ij}}{r_{ij}}\right)^{12}-2\left(\frac{r_{0ij}}{r_{ij}}\right)^{6}\right]\right\}+\mathrm{U}_{elec}$$

$O(N^1)$

2. The electrostatic energy calculation beyond cutoff will be done in the reciprocal space with FFT

$$\mathrm{U}_{elec}=\underbrace{\sum_{|i-j|<R}\frac{q_iq_j}{4\pi\varepsilon_0}\frac{\operatorname{erfc}(\alpha_{ij})}{r_{ij}}}_{\text{Real part}}+\underbrace{\frac{2\pi}{V}\sum_{\mathbf{G}\neq 0}\frac{\exp(-|\mathbf{G}|^2/4\alpha^2)}{|\mathbf{G}|^2}\sum_{ij}\frac{q_iq_j}{4\pi\varepsilon_0}\cos(\mathbf{G}\cdot r_{ij})}_{\text{Reciprocal part}}-\underbrace{\sum_{i}\frac{q_iq_i}{4\pi\varepsilon_0}\frac{\alpha}{\sqrt{\pi}}}_{\text{Self energy}}$$

3. Further, it could be reduced by properly distributing over parallel processors, in particular good domain decomposition scheme.

Why do we need lookup table?

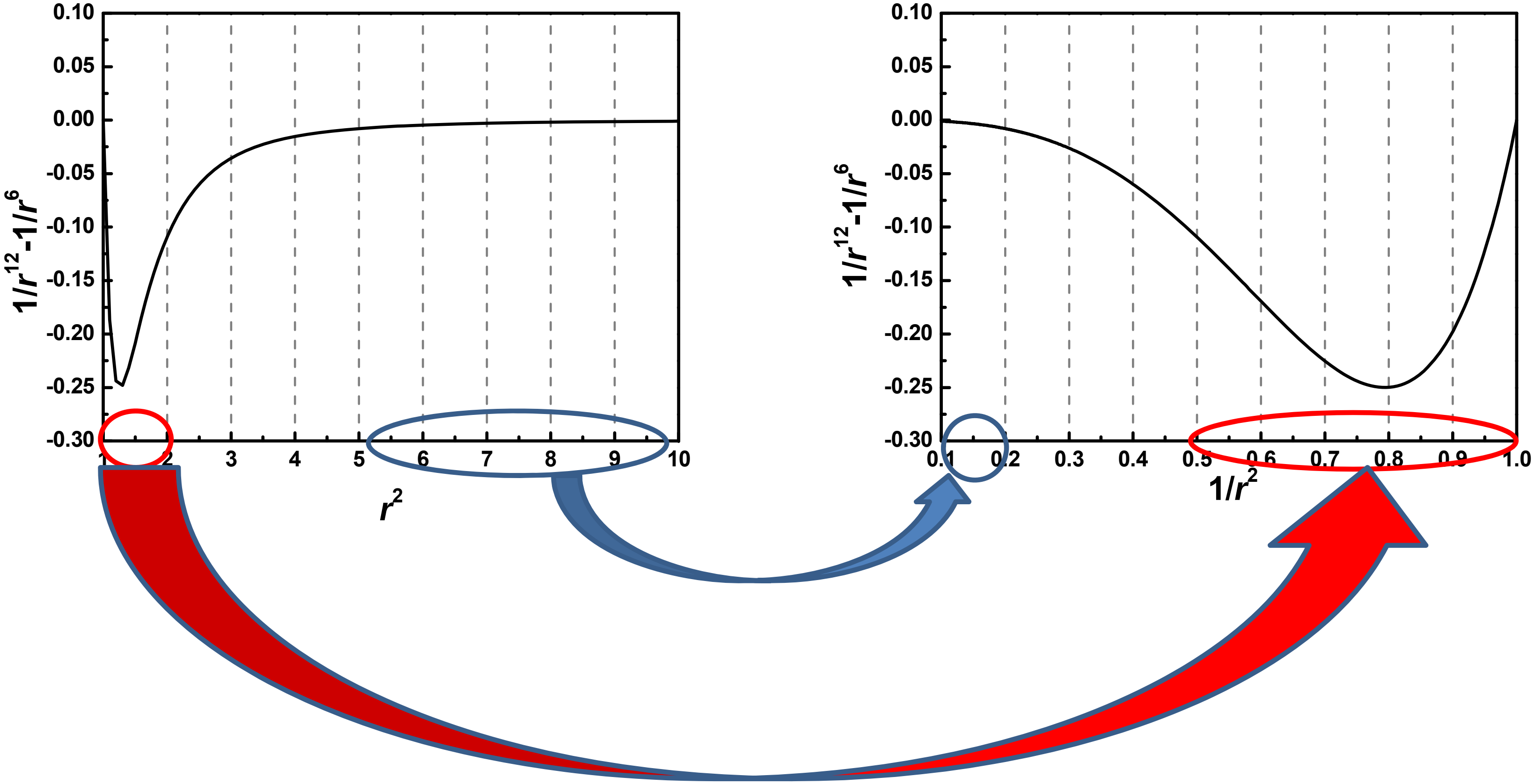
1. The short-range non-bonded interactions are the main computational bottleneck in molecular dynamics simulations

2. When we are using PME for electrostatic interactions, short-range non-bonded interaction includes time-consuming inverse square root and complementary error functions

3. Such time-consuming operations could be avoided by lookup table approach

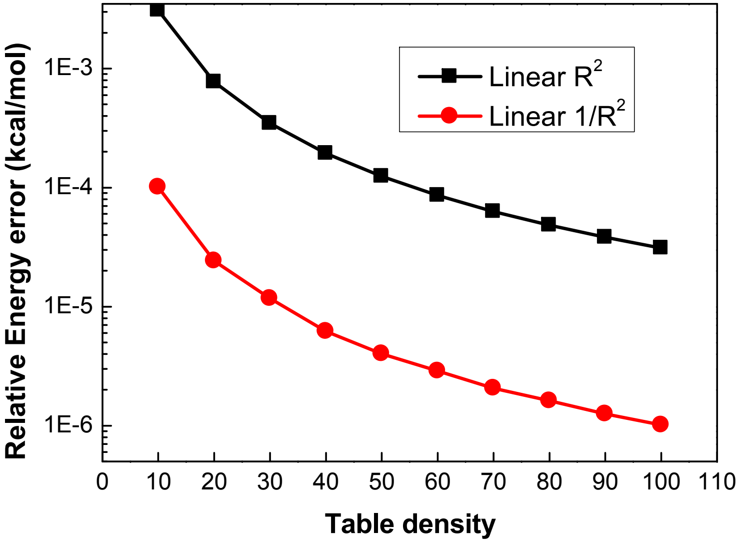
   example : Short-range non-bonded interaction takes 1.23 sec/step without using lookup table, and 0.45 sec/step with lookup table for DHFR

How do we change table points ?
(comparison of linear R^2 table and linear $1/R^2$ table)

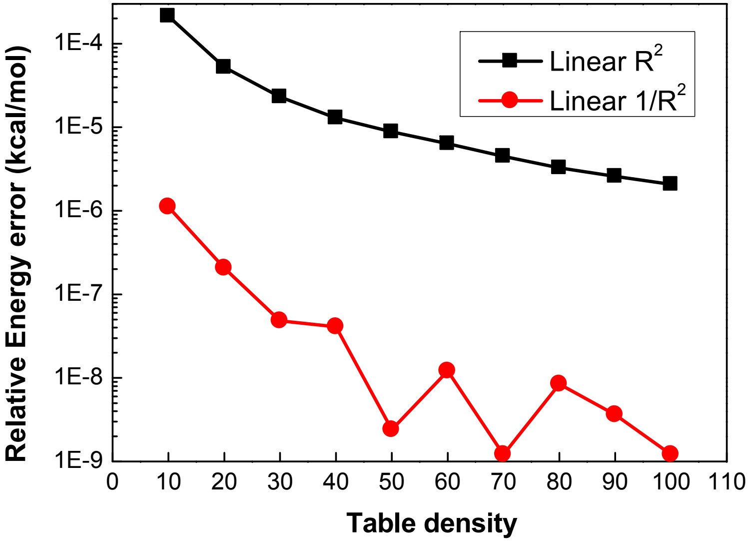


1. Assign more table points at small r^2 where energy and gradient change rapidly
2. For large r^2 where energy and gradient does not change so much, we can reduce the table points for faster evaluation.

Accuracy of single point energy calculation (DHFR)



Van der Waals

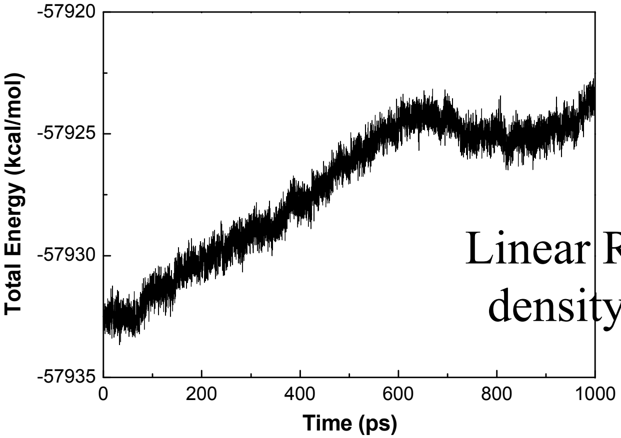


Electrostatic

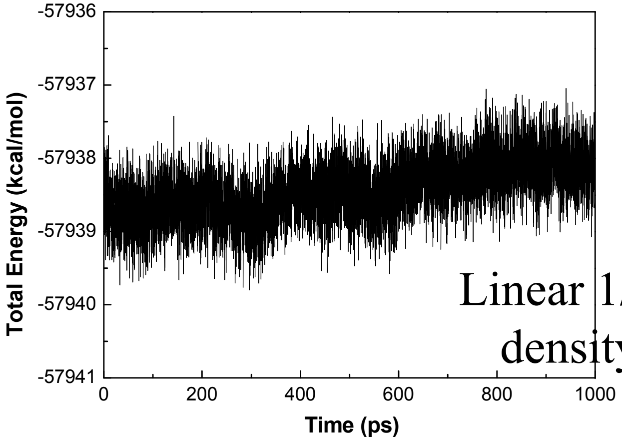
$$\Delta E = |E_{\mathrm{table}} - E_{\mathrm{direct}}| / |E_{\mathrm{direct}}|$$

1. Van der Waals : Linear $1/R^2$ with table density = 20 is more accurate than linear R^2 with table density = 100
2. Electrostatic : Linear $1/R^2$ with table density = 10 is more accurate than linear R^2 with table density = 100

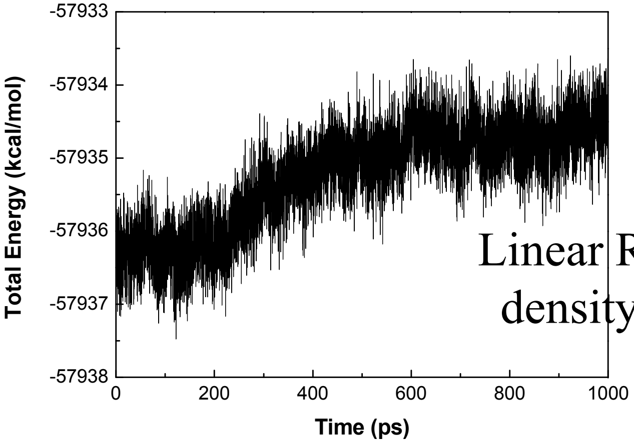
Energy drift (DHFR)



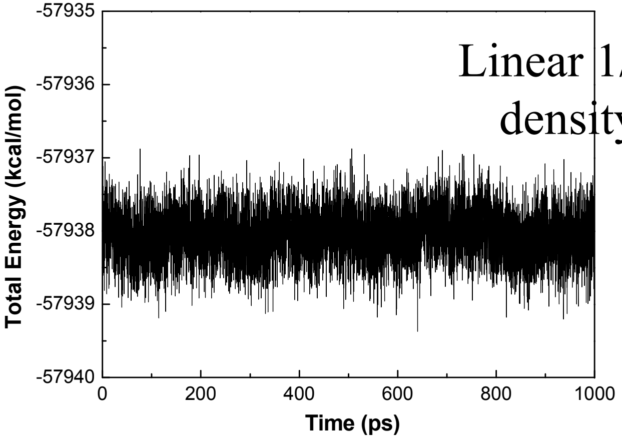
Linear R^2 with density = 10



Linear $1/R^2$ with density = 10



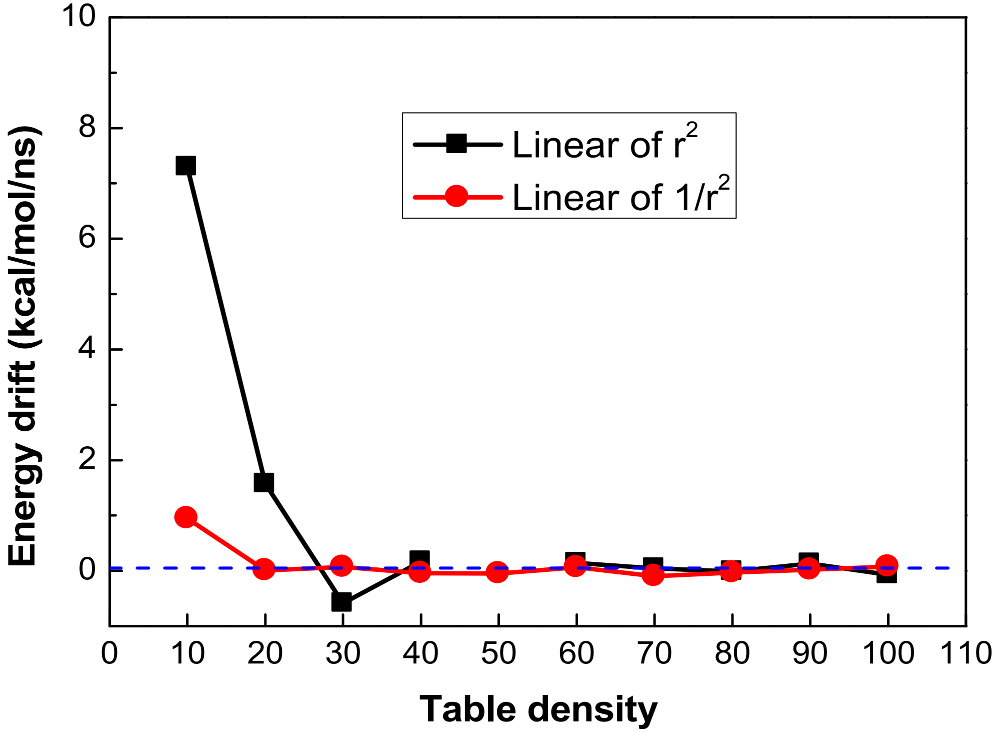
Linear R^2 with density = 20



Linear $1/R^2$ with density = 20

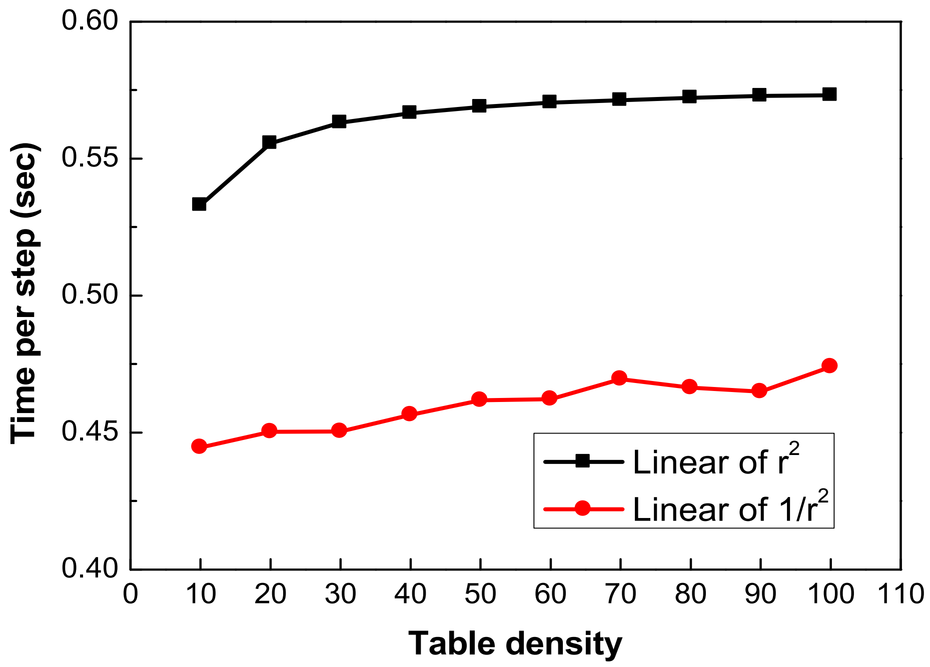
1. Large energy drift for both table density = 10 and 20 using linear R^2 table.
2. Slight amount of energy drift for table density = 10 and no energy drift from table density = 20 using linear $1/R^2$ table.

Energy drift of DHFR according to table density



1. Linear R^2 table : reliable energy drift from table density = 40.
2. Linear $1/R^2$ table : reliable energy drift from table density = 20

Non-bonded interaction time of DHFR (resources : K)



On K, linear 1/ R^2 table shows better performance, due to efficient inverse operation and cache usage.

Why linear 1/ R^2 table is efficient for large table density?

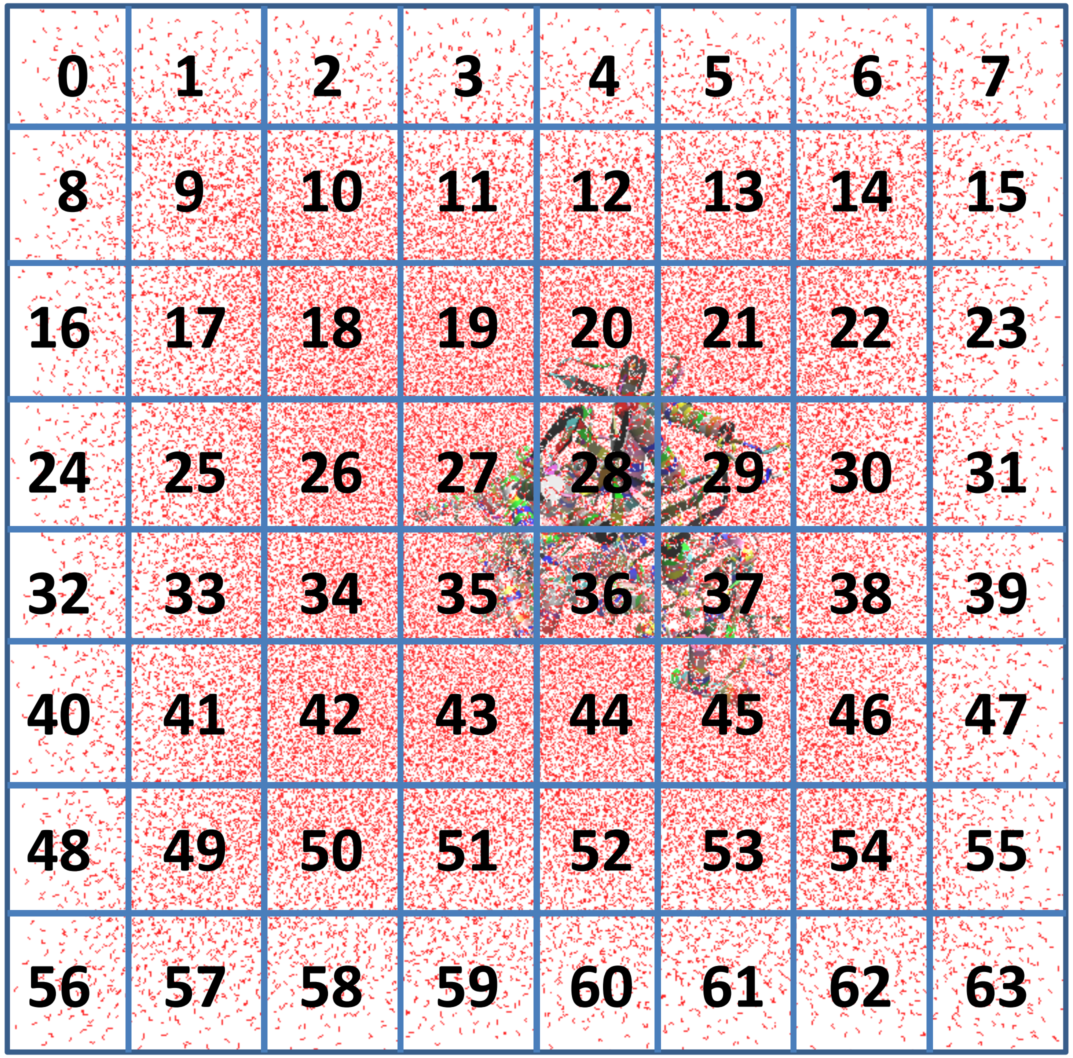
Interval	Number of interactions	Number of sections in linear R^2 table	Number of sections in linear $1/R^2$ table
$1<r^2<16$	269245	15	135
$16<r^2<36$	730108	20	5
$36<r^2<144$	7295834	108	3

Less cache misses

1. Using Linear $1/R^2$ table, 90% interactions are dealt with only three sections and increasing table density does not introduce cache misses for this interval.
2. On the other hand, linear R^2 table uses 108 sections for the same interval and cache misses will be severe by increasing table densities.
3. In K, this phenomena highly affect SIMD and this explains higher speed of linear 1/ R^2 table even at small table density

GENESIS parallelization scheme
- Midpoint cell method

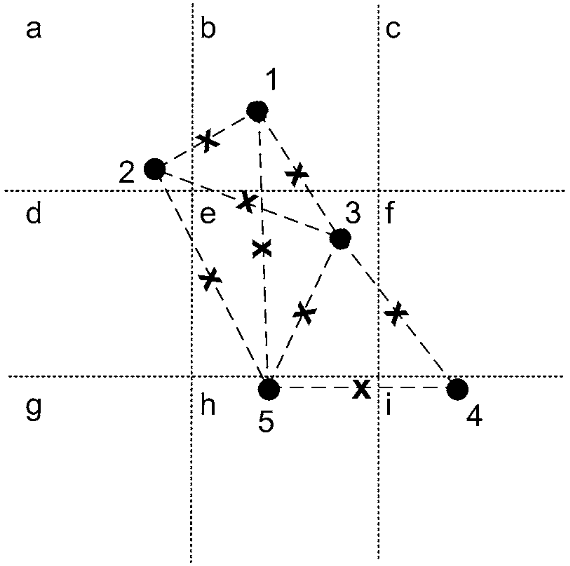
Domain Decomposition



1. Partitioning space into fixed-size boxes, with dimension larger than the cutoff distance
2. We need only information of neighbor space(domain) for computation of energies
3. Communication is reduced by increasing process number
4. Efficient for good parallelization and suitable for large system with large processors

Basic scheme of parallelization - Midpoint method

1. Two particles interact on a particular box if and only if the midpoint of the segment connecting them falls within the region of space associated with that box

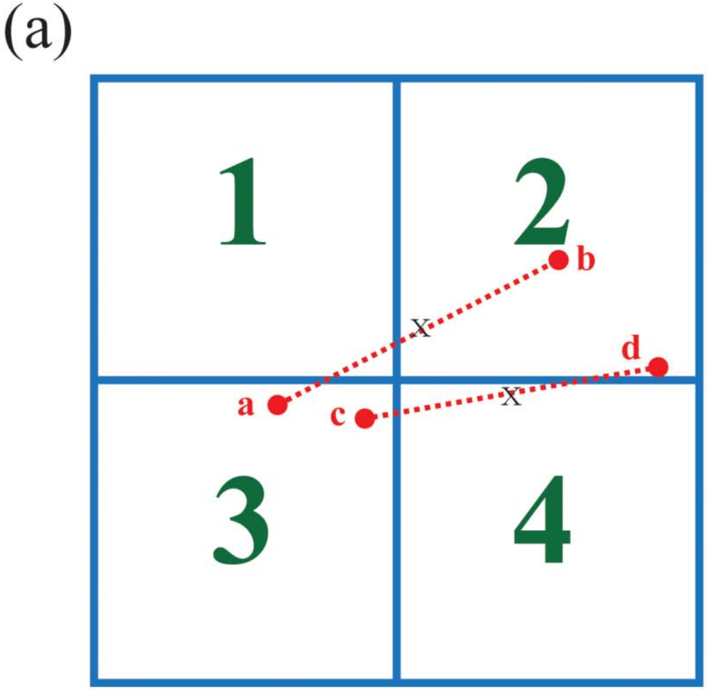


Each pair of particles separated by a distance less than R (cutoff distance) is connected by a dashed line segment, with “x” at its center lying in the box which will compute the interaction of that pair

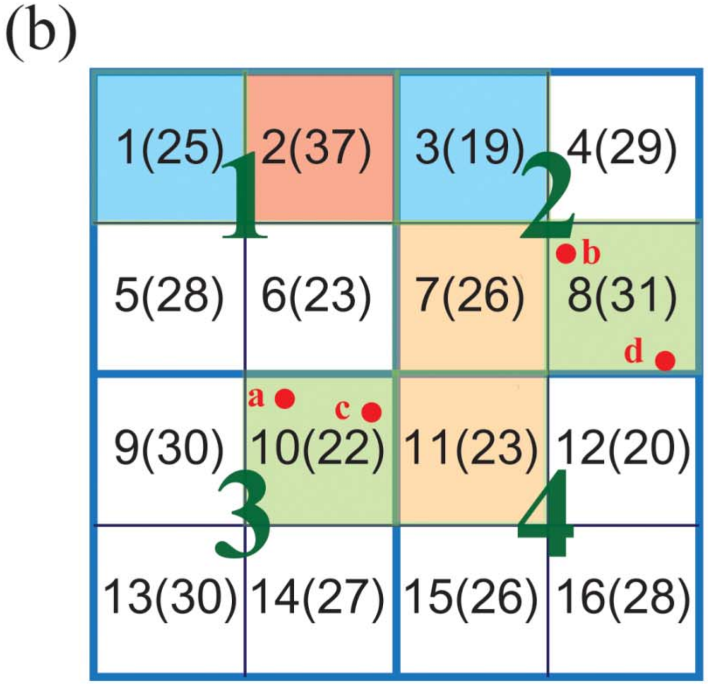
2. This scheme applies not only for non-bonded but also bonded interactions.

Ref : KJ. Bowers et al, JCP 124, 184109 (2006)

Midpoint cell method



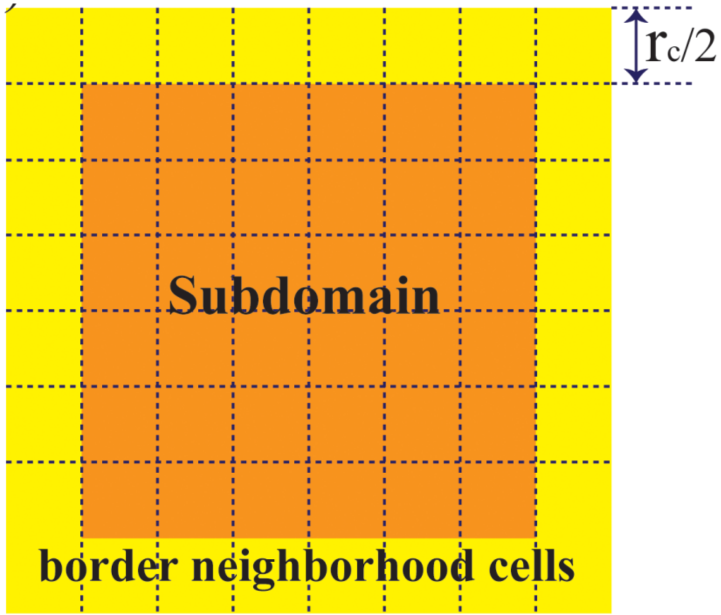
Midpoint method : interaction between two particles are decided from the midpoint position of them.

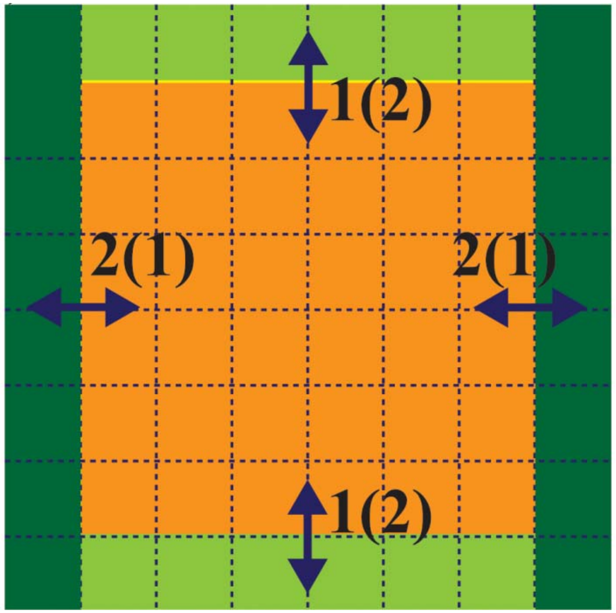


Midpoint cell method : interaction between two particles are decided from the midpoint cells where each particle resides.

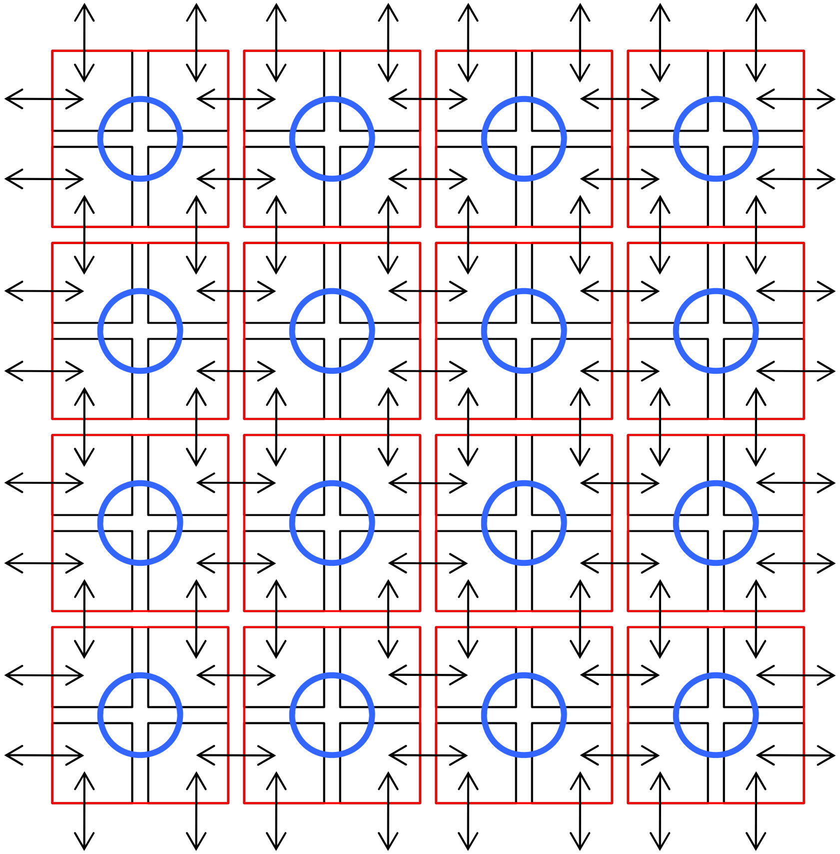
Small communication, efficient energy/force evaluations

Communication space





Hybrid parallelization in GENESIS



1. The basic is the Domain decomposition with the midpoint scheme

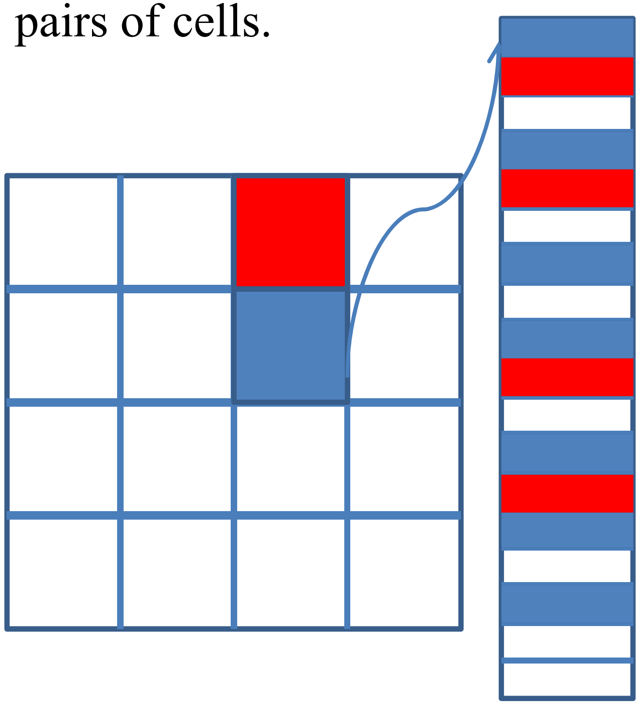
2. Each node consists of at least two cells in each direction.

3. Within nodes, thread calculation (OPEN MP) is used for parallelization and communication is not necessary

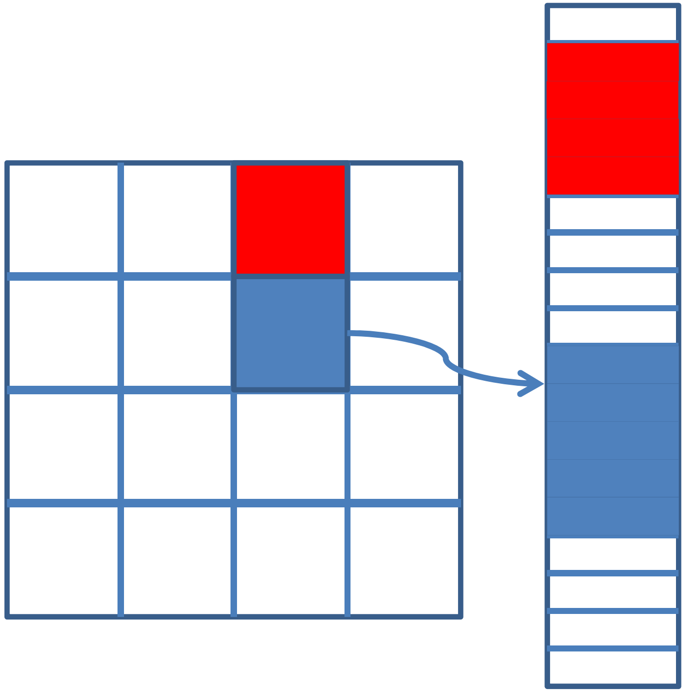
4. Only for different nodes, we allow point to point communication (MPI) for neighboring domains.

Efficient shared memory calculation =>Cell-wise particle data

1. Each cell contains an array with the data of the particles that reside within it
2. This improves the locality of the particle data for operations on individual cells or pairs of cells.



particle data in traditional cell lists



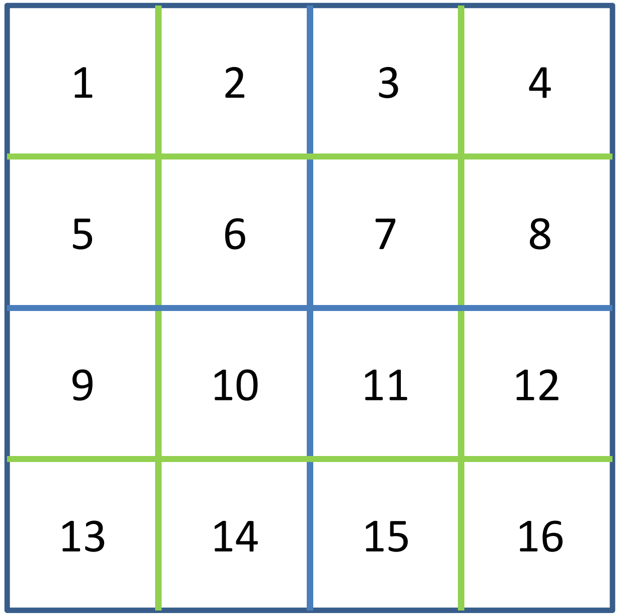
cell-wise arrays of particle data

Ref : P. Gonnet, JCC 33, 76-81 (2012)

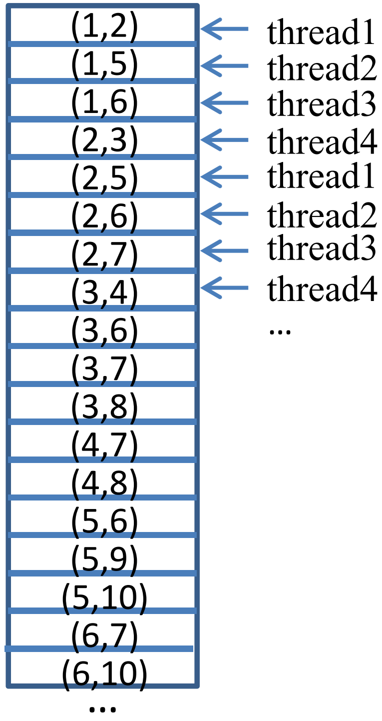
How to parallelize by OPENMP?

1. In the case of integrator, every cell indices are divided according to the thread id.

2. As for the non-bond interaction, cell-pair lists are first identified and cell-pair lists are distributed to each thread



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



(1,2)	thread1
(1,5)	thread2
(1,6)	thread3
(2,3)	thread4
(2,5)	thread1
(2,6)	thread2
(2,7)	thread3
(3,4)	thread4
(3,6)	…
(3,7)	
(3,8)	
(4,7)	
(4,8)	
(5,6)	
(5,9)	
(5,10)	
(6,7)	
(6,10)	
…	

Parallelization result

(K with 8 OPENMP

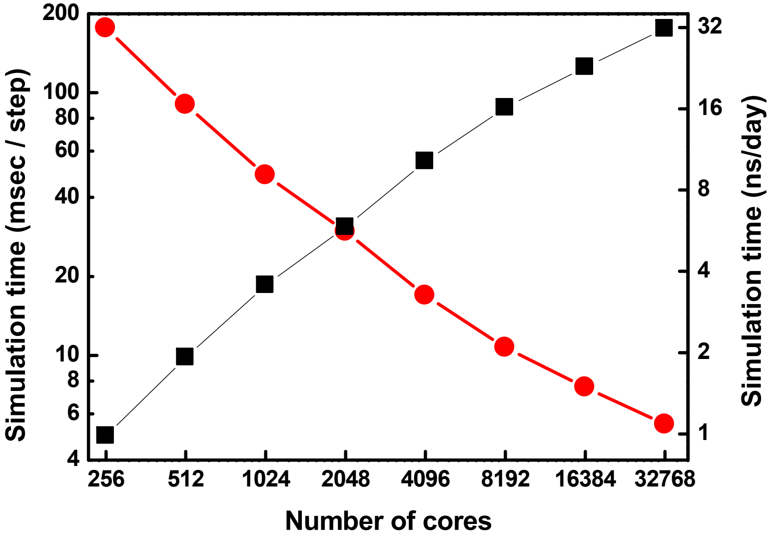
Cutoff = 12.0 Å, Particle Mesh Ewald)

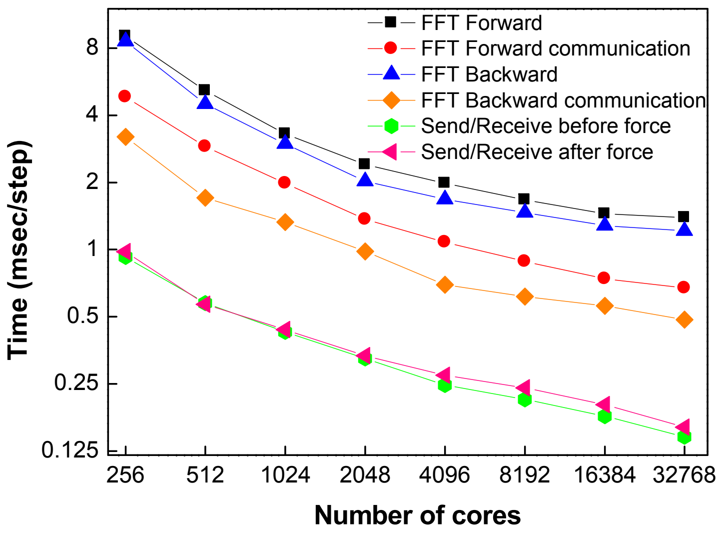
Time (sec) per step of 1M System (STMV)

Core Number	NAMD	GENESIS
256	0.1608	0.1584
512	0.0908	0.0824
1024	0.0580	0.0443
2048	0.0300	0.024
4096	0.0243	0.0163
8192	0.0260	0.0087
16384		0.0058
32768		0.0047

1. Here we used NAMD compiled on K (provided by Dr. Miyashita)
2. Load balance is not considered for NAMD. If load balance option is on, it is faster than w/o load balance up to 2048 cores, but it becomes slower as we increase the process numbers
3. Grid size of namd input : 216x216x216, Grid size of GENESIS : 256x256x256

Overall performance for 1.3 M system

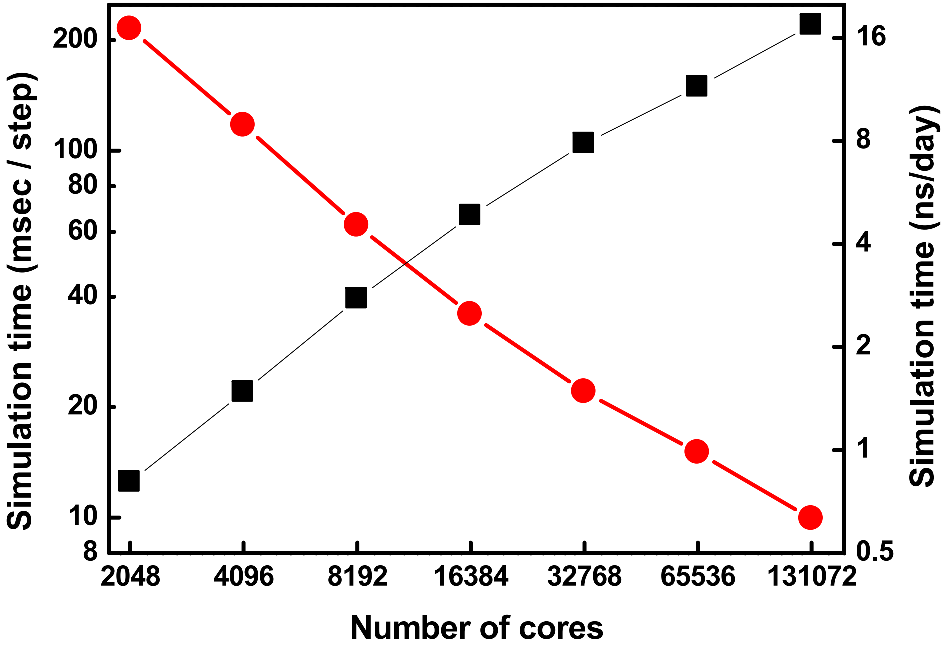


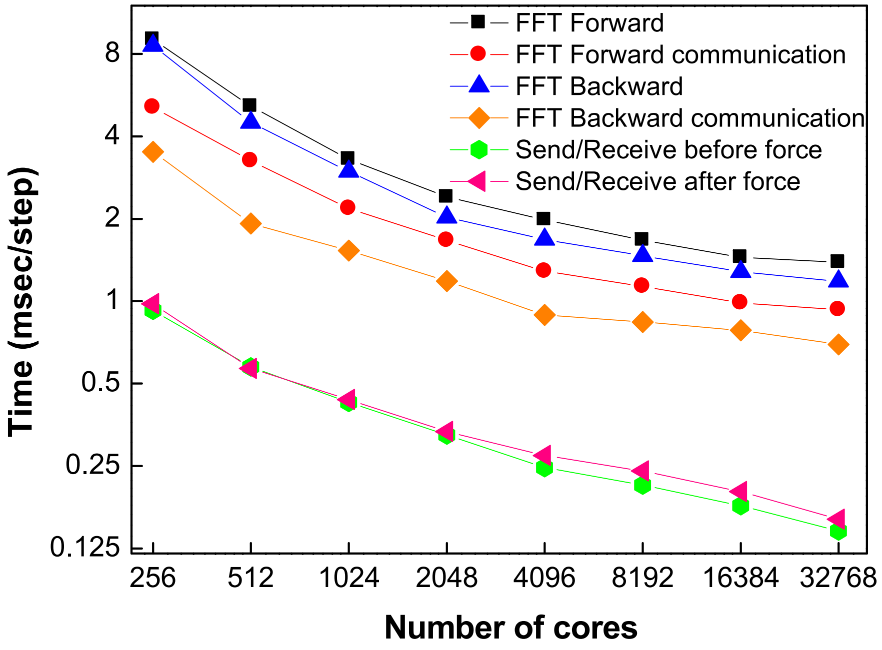


1. It is scalable up to 4,096 nodes (32,768 cores)

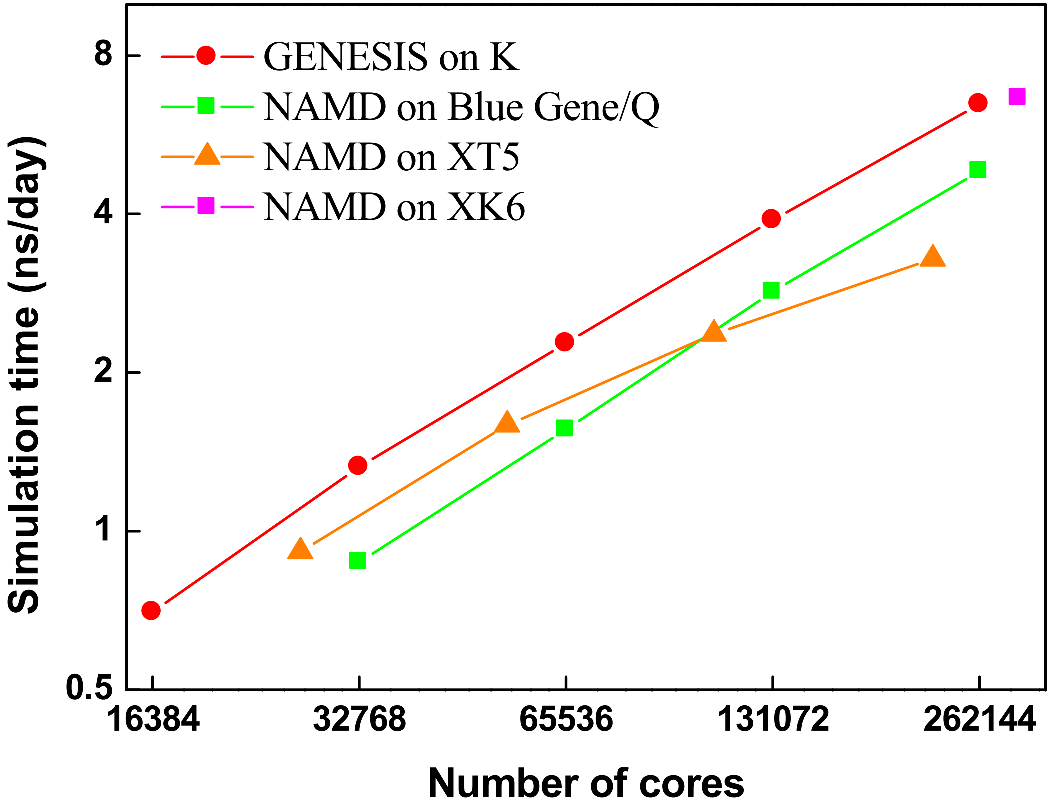
2. Not only energy/force evaluation, but also communications for FFT and send/receive are scalable even though the scaling efficiency is not perfect

System : 11.2 M atoms (time step = 2fs)





System : 100 M atoms



Summary

- GENESIS uses newly developed lookup table for more accurate and fast evaluation of non-bonded interaction
- GENESIS uses the midpoint cell method for our domain decomposition scheme of parallelization.
- Hybrid parallelization (MPI+OPENMP) is considered for efficient parallelization.
- GENESIS is scalable up to the number of cells, in particular, 131,000 cores for 11.7M systems (more than 16ns/day) and 262,cooo cores for 100M systems.