N-Body Simulations

- Why is high performance computing needed?
- The case of long range forces.
    - Gravitational N-Body simulations.
- Cosmological N-Body simulations.

Why HPC?

- Real systems of interest often evolve at widely time scales, thus a large number of time steps are required in the simulation.

- It takes a very large number of particles to capture essential features of the system to be simulated.

- Systems of equations to be solved are complex and often do not have any analytical solutions in the regime of interest.

CPU Time Requirements

• Consider a system of N_p particles. If we have to simulate its evolution through N_t time steps and τ_s is the time taken per particle per step then:

$$\tau_{CPU} = N_p N_t \tau_s = 30\, hours \left(\frac{N_p}{10^6}\right)\left(\frac{N_t}{10^3}\right)\left(\frac{\tau_s}{10^{-4}s}\right). \tag{1}$$

○ Thus a large system cannot be simulated unless we can reduce τ_s.

○ A system that has to be evolved through larger number of time steps must be small, or we must be patient.

Memory Requirements

• If we have N_{var} variables per particle, and double precision variables then the memory requirements are:

$$M = 8 N_p N_{var} = 80\, MB \left(\frac{N_p}{10^6}\right)\left(\frac{N_{var}}{10}\right). \tag{2}$$

○ The minimum number of variables per particle is about 10. These are mass, position, velocity and acceleration.

○ If the time requirements are not a constraint, then often the upper limit on the size of the simulation comes from total memory available.

Object	Mass ($M_\odot$)	Size (pc)	V (km/s)	$\lvert E_{gr}\rvert$ (ergs)	τ_{cross} (years)
Binary Star	2	10^{-7}	30	10^{48}	1
Open Cluster	10^3	$1 - 10$	10	10^{48}	10^5
Globular Cluster	10^5	10	20	10^{50}	10^6
Galaxies	$10^8 - 10^{11}$	$10^3 - 10^4$	10^2	$10^{56} - 10^{61}$	$10^6 - 10^8$
Galaxy clusters	10^{15}	10^6	10^3	10^{65}	10^8

Short Range Forces

- If the interaction is very short range, force due to the nearest neighbours is needed and distant particles do not contribute significantly to the force.

- Such simulations are challenging only if the time scale of local motions is very small compared to the relaxation time of the system as a whole.

- Such simulations are easy to parallelise on distributed memory computers.

Long Range Forces

• Forces due to all the particles in the system are important, thus force calculation is an $\mathcal{O}(N_p^2)$ process. Also, $\tau \propto N_p$.

• The main aim in developing algorithms is to approximate the force calculation so that it requires a smaller number of compute operations.

• Information on all particles is needed, hence communication overhead can be very demanding on distributed memory computers.

Tree Code

● Given a sufficiently distant and compact group of particles, force due to the group can be approximated by the force of a particle of the same mass located at the center of mass.

○ We need a criterion to quantify whether a group of particles is far enough, and compact enough.

○ We need to arrange particles in groups.

○ We need to estimate the error due to this approximation.

The Tree Method

• Distant groups of particles can be treated as a unit for the purpose of force calculation.

$$\begin{aligned} a &= -\frac{GM}{(r-d/2)^2} - \frac{GM}{(r+d/2)^2} \\ &= -\frac{2GM}{r^2}\left(1 + \frac{3}{4}\frac{d^2}{r^2} + \frac{5}{16}\frac{d^4}{r^4} + \mathcal{O}\left(\frac{d^6}{r^6}\right)\right) \end{aligned} \qquad (3)$$

• Fractional error scales as θ^2 is we retain only the mono-pole term and θ^4 if we include the quadruple term, here $\theta = d/r$.

The Tree Method

• For $\theta = 0.5$, the worst case error is close to 20% if we do not include the quadruple term. With this term the maximum error is 2%.

• For $\theta = 0.25$, the worst case error is close to 5% if we do not include the quadruple term. With this term the maximum error is 0.12%.

∘ In generic cases the error is much smaller.

∘ Error decreases as we increase N_p.

• Larger values of θ should not be used as this can lead to large errors, even though errors for generic distributions of particles are small.

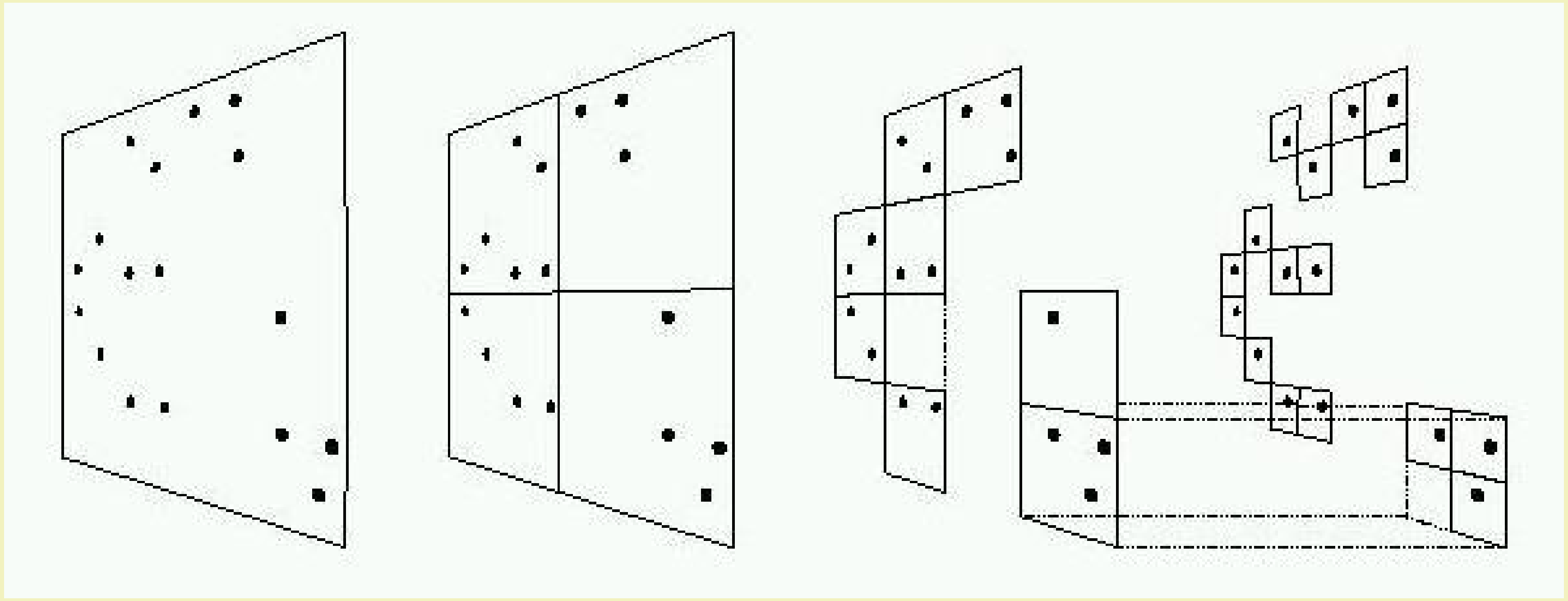
• If $\theta \geq 1/\sqrt{3}$ then it is important to ensure that there is no self force.

Barnes-Hut Tree Code

• The Barnes and Hut method uses geometrical subdivision of the simulation volume. This reduces the problem of force calculation to $\mathcal{O}(N_p \log N_p)$.

• The process of constructing the tree is iterative. The starting point is the simulation volume.

• The simulation box is the "trunk". This is sub-divided into smaller volumes/cells at each level. The cells are "branches". Cells are sub divided till there is at most one particle in the smaller cells. Particles correspond to "leaves".



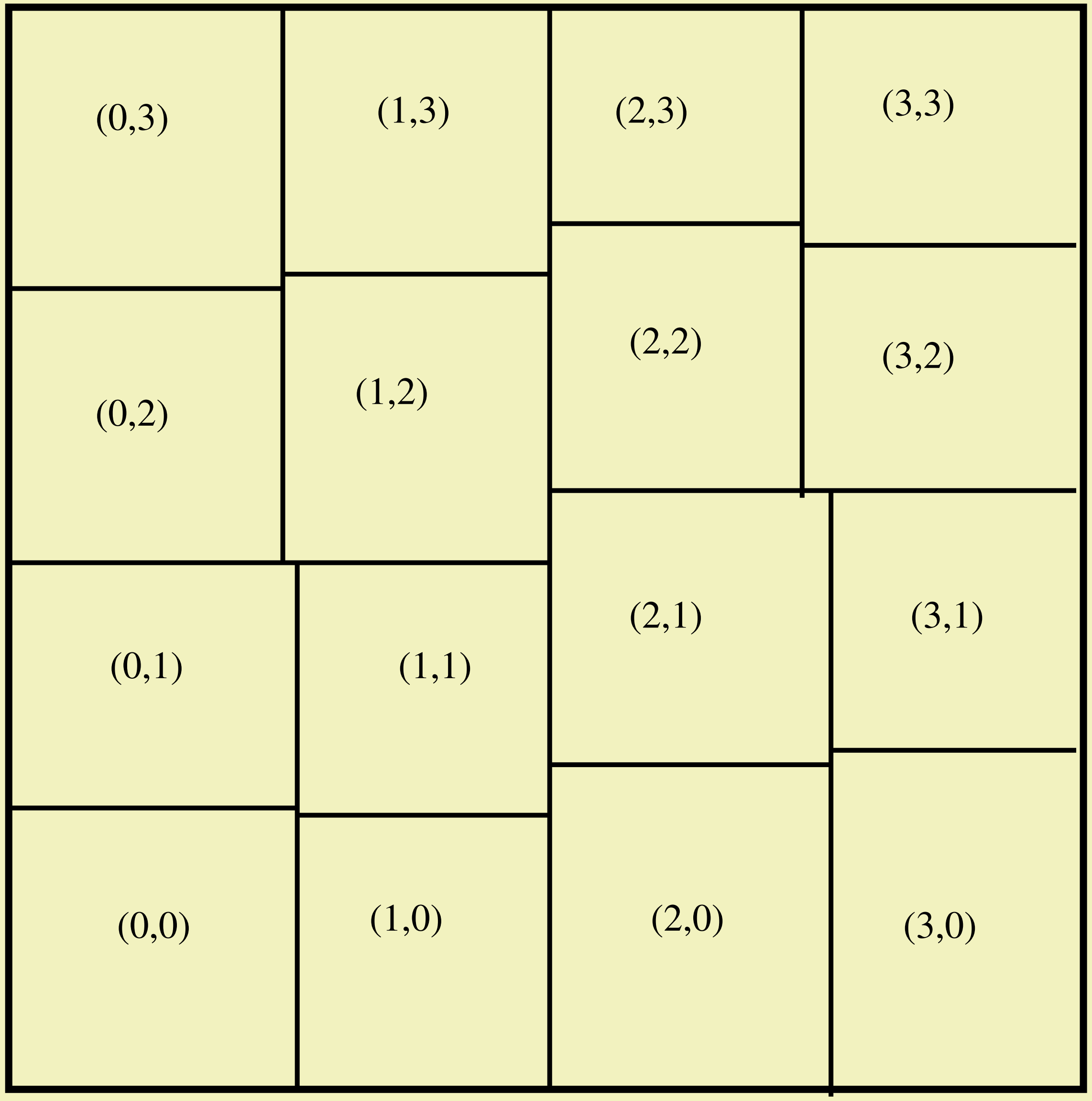
Parallelising Barnes-Hut Tree Code

• Calculation of force for each particle can proceed in parallel.

• Less information is required from distant parts of the particle distribution, so grouping particles in domains will be useful.

• In absence of a mesh, recursive orthogonal bisection of the simulation volume is used to construct domains.

• Other domains send the relevant part of the tree for completion of force calculation. Number of communications requires is large and grows as n^2, n being the number of processors.


(0,3)
(1,3)
(2,3)
(3,3)
(2,2)
(3,2)
(0,2)
(1,2)
(2,1)
(3,1)
(0,1)
(1,1)
(0,0)
(1,0)
(2,0)
(3,0)

Cosmological N-Body Simulations

- The universe does not have a boundary, so it is appropriate to use periodic boundary conditions.
- The universe is expanding. This changes the nature of initial conditions and the equations of motion.
- Hard scattering between particles is to be suppressed as each N-Body particle represents a large number of real particles, each particles is assumed to have a finite size.

Particle-Mesh (PM) Method

● Poisson equation is a simple algebraic equation in Fourier space. Fast-Fourier method can be used to compute Fourier Transforms using $\mathcal{O}(N_p \log N_p)$ operations. FFT can also be used to compute the gradient of the potential at grid points.

○ FFT requires a uniformly spaced grid on which the fields are defined, hence the mesh has to be of this type.

● Use particles to describe the density and velocity field.

○ Use a grid to solve Poisson equation.

○ Use interpolating functions to switch between the particles and the grid/mesh.

PM Method

• Require the sum of weights to be unity. Difficult to construct a spherically symmetric interpolation function.

• A convenient approach is to use the product of three one dimensional weight functions, $W = W_x W_y W_z$.

○ This results in an anisotropic interpolation function, thus the effective kernel is anisotropic.

• This is a very fast method for simulations.

○ Periodic boundary conditions come free with FFT.

○ The force is softened below grid scale, hence the evolution is collisionless. However, the resolution is very poor.

PM Method: Parallelisation

• Dividing the simulation volume into slabs works well for PM codes. (See PMFAST)

• Using FFTW facillitates parallelisation.

• Simulations with as many as 10^9 particles have been done with parallel PM codes.

Slab 1	Slab 2

TreePM Method: Force Decomposition

- We start by partitioning the Poisson equation:

$$\begin{aligned}
\nabla^2 \varphi &= 4\pi G \varrho \\
\nabla^2 \varphi_l &= 4\pi G \varrho \exp\left[-r^2/r_s^2\right] \\
\nabla^2 \varphi_s &= 4\pi G \varrho \left(1 - \exp\left[-r^2/r_s^2\right]\right) \qquad (4)
\end{aligned}$$

Here r_s is a scale that we have introduced. This is to be fixed using estimation of errors.

TreePM Method: Force Decomposition

- Long range potential is calculated on the mesh using FFT:

$$\phi_{l\,\mathbf{k}} = -\frac{4\pi G \varrho_{\mathbf{k}}}{\mathbf{k}^2} \exp\left[-k^2 r_s^2\right] \tag{5}$$

- The short range force is computed in real space using the tree method:

$$\mathbf{f}_s = -\frac{G\mathbf{r}}{r^3}\left(\operatorname{erfc}\left(\frac{r}{2r_s}\right) + \frac{r}{r_s\sqrt{\pi}} \exp\left[-\frac{r^2}{4r_s^2}\right]\right) \tag{6}$$

TreePM Method: Errors

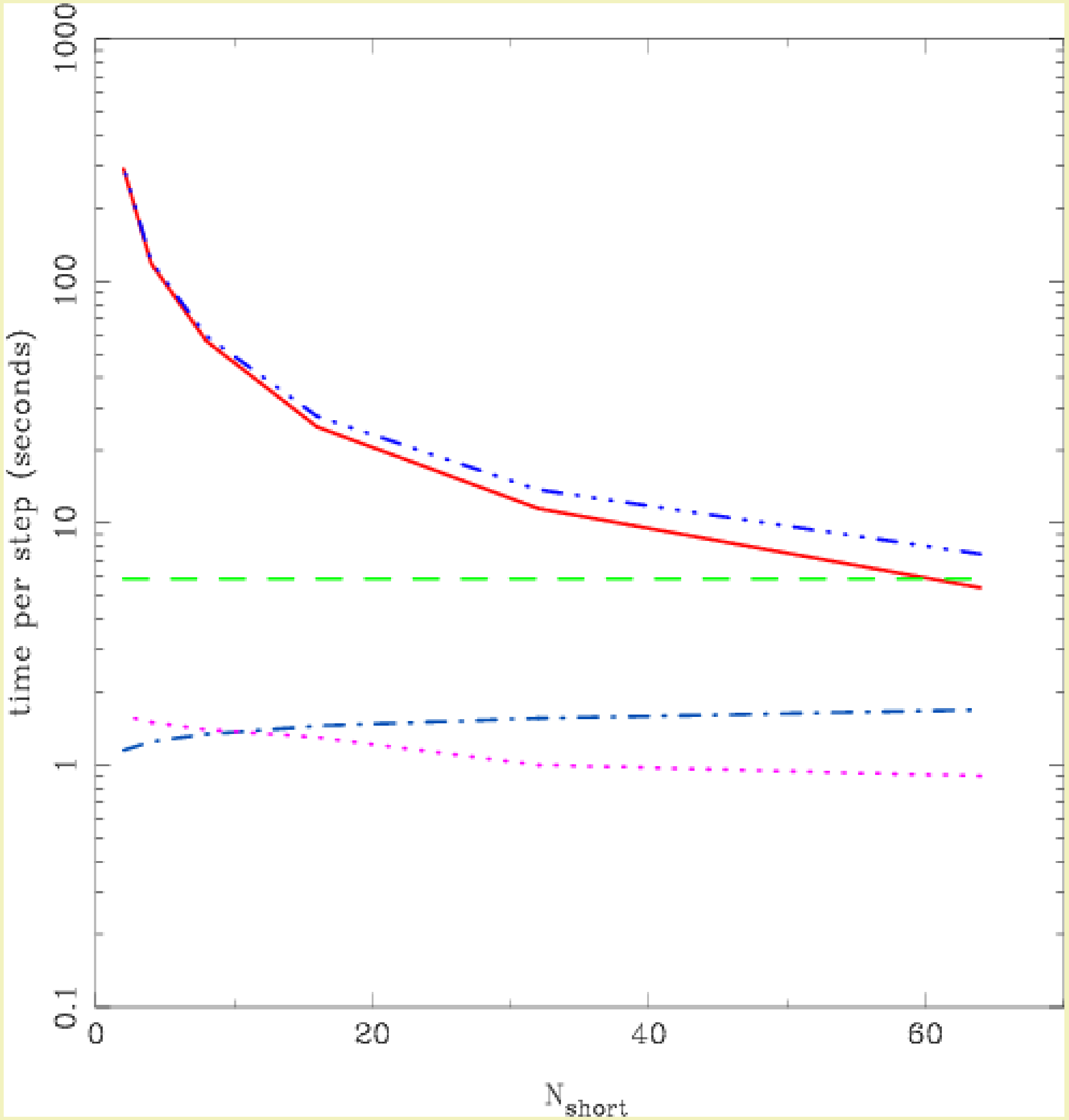
- For $r_s = 1$ and $\theta = 0.5$, 99% particles have less than 0.8% error for a clustered distribution. For an unclustered distribution, 99% particles have less than 2% error.

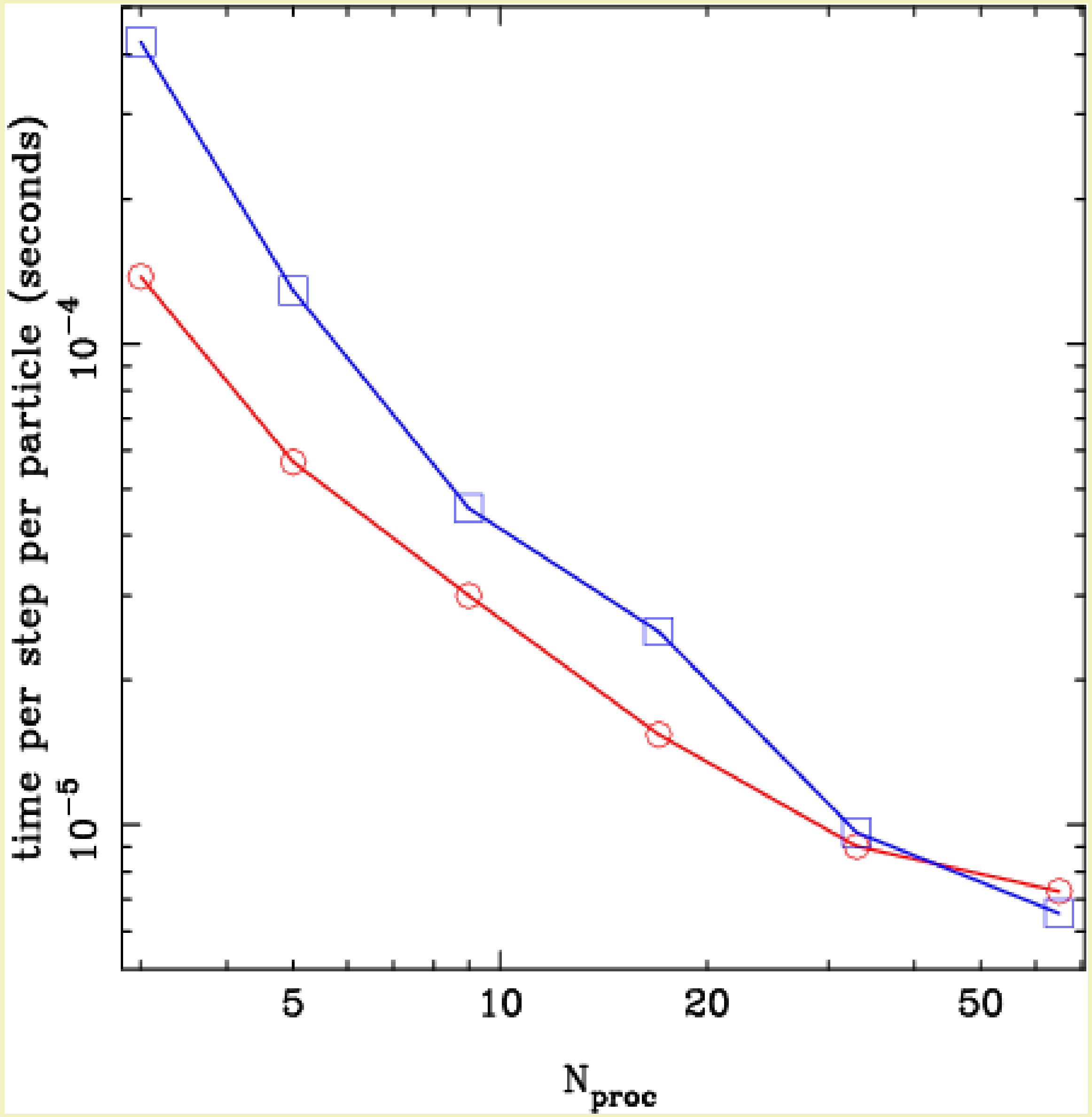
For tree code (Hernquist and Bouchet, 1991), this figure varies between 1% and 6%.

TreePM Method: Parallelisation

- Domain decomposition: Divide the simulation box into domains with equal computational load for short range force calculation.

- Functional decomposition: Divide the computation of the short and the long range force. Typically, only one CPU is sufficient for long range force calculation.


time per step (seconds)
1000
100
10
1
0.1
0
20
40
60
N_{short}



time per step per particle (seconds)
10^{-4}
10^{-5}
5
10
20
50
N_{proc}