LAMMPS: Code Transformations in Preparing for Titan - A Hybrid Multi-Core Architecture

Presenter: Arnold Tharrington

Prepared by Arnold Tharrington and William Brown

Scientific Computing Group, OLCF

Outline

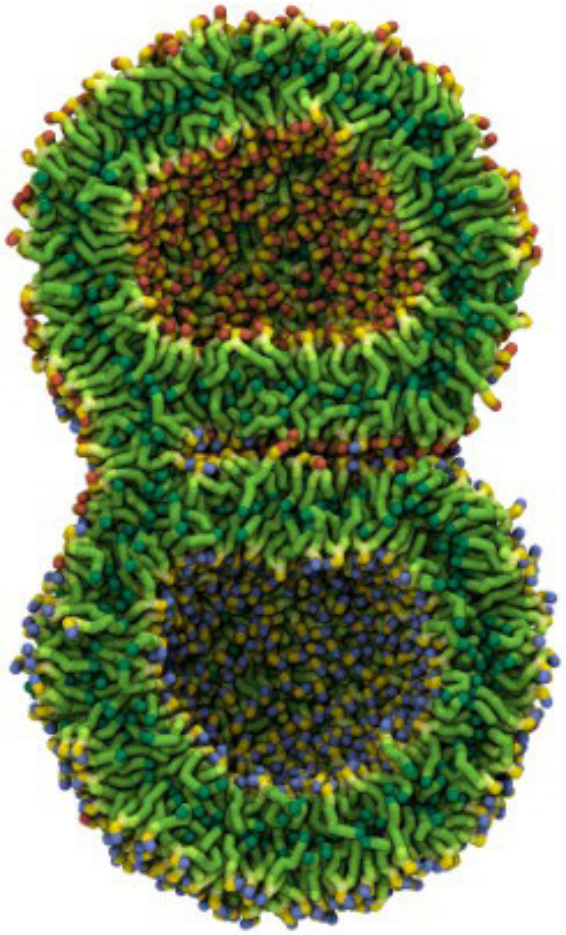
1. Convey ongoing code modifications needed to prepare LAMMPS for Titan
    1. Give an overview of Molecular Dynamics
        - Overview of MD/LAMMPS
        - Runtime profile of LAMMPS
    2. Accelerating LAMMPS
        - Accelerating the nonbonded interactions
        - Accelerating the electrostatic interactions
2. Summary

1. Science problem

Science Problem: Molecular Level Insight into Membrane Fusion from Molecular Dynamics

- Gain molecular level understanding of membrane fusion mechanisms
  - A major means for materials to enter and exit cells
  - Role of lipids and membrane proteins

To date, the majority of simulation work on membranes has focused on model systems containing typically one to two lipid types. However biological membranes are complex mixtures, and to model these systems without finite size effects, simulations of significantly larger systems than what is currently possible needs to be performed.



Computer image of membrane vesicles undergoing fusion.

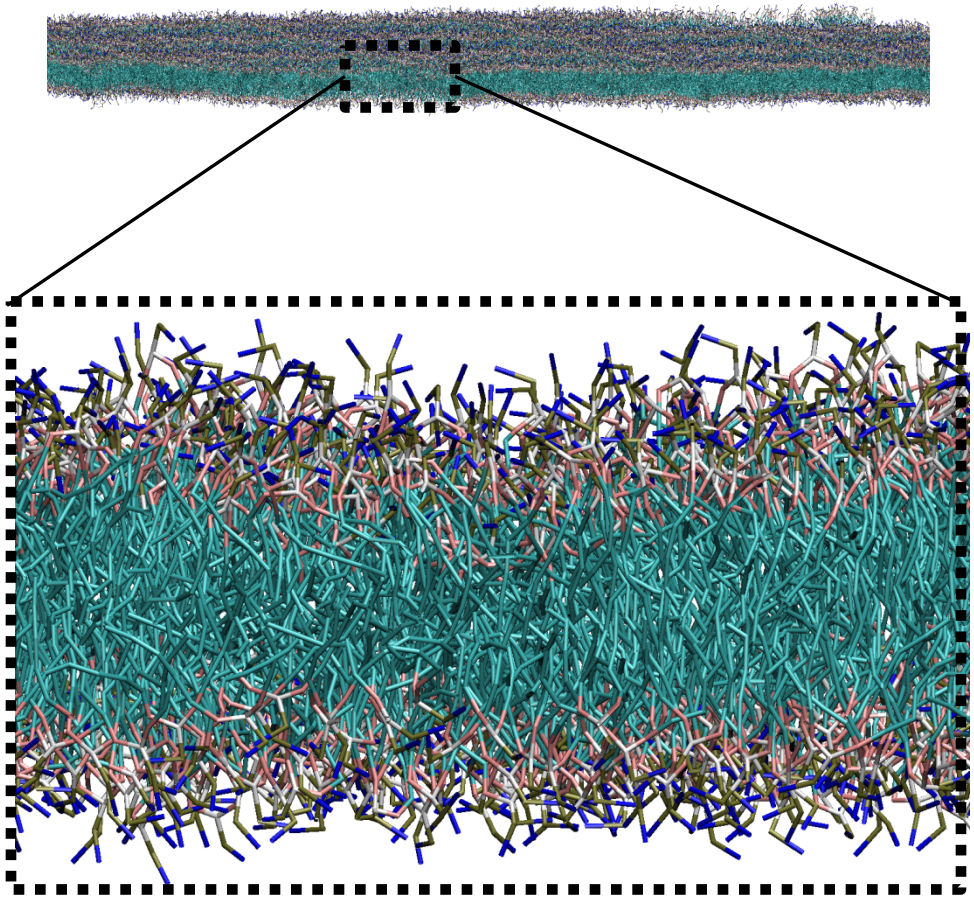
Science Problem: Molecular Level Insight into Membrane Fusion from Molecular Dynamics (cont.)

- Some requirements for realistic simulations are
  - Proper solvation
  - Eliminate modeling artifacts due to interactions between periodic images
  - Capture long wavelength modes
  - Proper treatment of coulombic or electrostatic interactions

- The smallest, experimentally accessible vesicles to date have diameters of approximately 500 Å.

- We need simulations with the minimum size of 750,000,000 coarse grain particles to model experimentally produced vesicles

- Acceleration of the simulation is needed to reduce the time to solution!!



Computer image of the 2 petaflop target science problem. The membrane bilayer represented by coarse grain bead model.

Science Problem: Molecular Level Insight into Membrane Fusion from Molecular Dynamics (cont.)

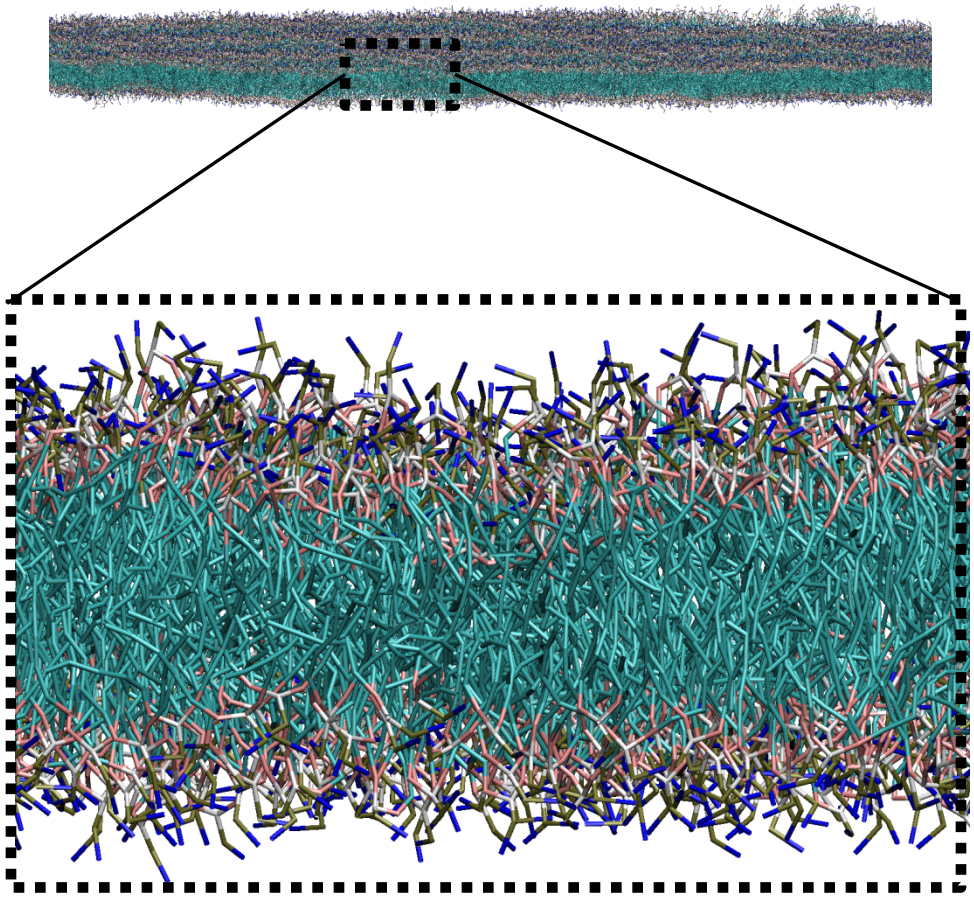
Coarse grain model of Dipalmitoylphosphatidylcholine (DPPC) stacked bilayers

2 PF problem:

- **A charged system of 834,403 coarse grained beads**
- **System dimensions ~1000Å x 1000Å x 33 Å**

20 PF problem:

- **A charge system of ~850,000,000 coarse grained beads**
- **System Dimension ~ 32,000Å x 32,000Å x 33 Å**

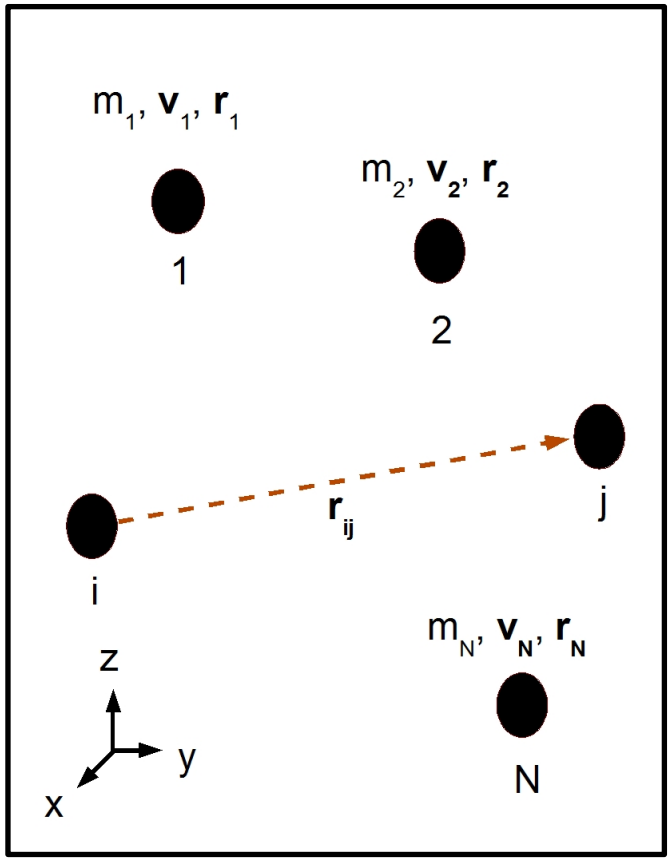


Computer image of the 2PF science problem. The membrane bilayer is represented by coarse grain beads.

2. LAMMPS and MD Overview

MD Simulations and Algorithm, F=ma

Calculate the trajectory of a system of particles interacting with potential U.



$$U = U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N); \quad \vec{F}_i = -\frac{\partial U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)}{\partial \vec{r}_i}$$

$$m_1 \frac{d^2\vec{r}_1}{dt^2} = \vec{F}_1 = -\nabla_1 U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$$

$$m_2 \frac{d^2\vec{r}_2}{dt^2} = \vec{F}_2 = -\nabla_1 U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$$

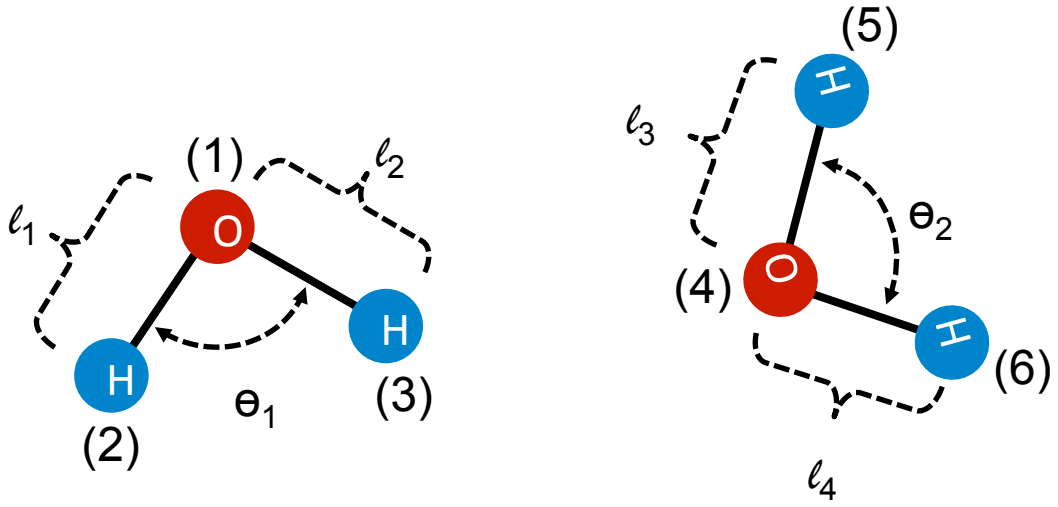
$$\vdots$$

$$m_N \frac{d^2\vec{r}_N}{dt^2} = \vec{F}_N = -\nabla_N U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$$

m_i is the mass of particle i
r_i is the position of particle i
v_i is the velocity of particle i

Solve by Velocity Verlet algorithm

Force fields details



$$U_{system} = U_{bond} + U_{angle} + U_{electrostatic} + U_{nonbonded}$$

$$U_{bond} = \frac{1}{2}k^{b}_{OH}\left(l_{OH} - l_1\right)^2 + \frac{1}{2}k^{b}_{OH}\left(l_{OH} - l_2\right)^2 + \frac{1}{2}k^{b}_{OH}\left(l_{OH} - l_3\right)^2 + \frac{1}{2}k^{b}_{OH}\left(l_{OH} - l_4\right)^2$$

$$U_{angle} = \frac{1}{2}k^{\Theta}_{OH}\left(\Theta_{HOH} - \Theta_1\right)^2 + \frac{1}{2}k^{\Theta}_{OH}\left(\Theta_{HOH} - \Theta_2\right)^2$$

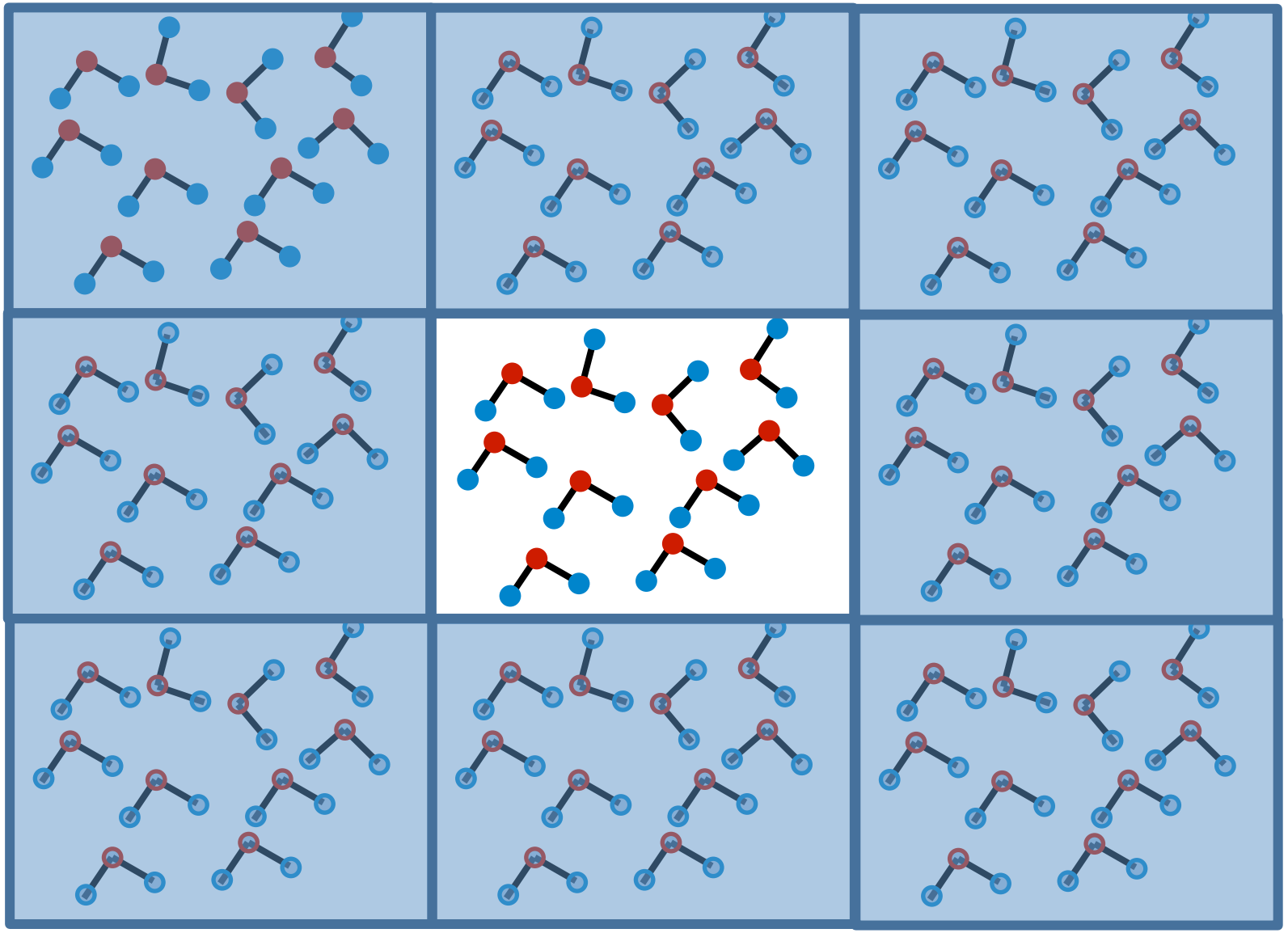
$$U_{nonbonded} = \sum_{i=1}^{N} \sum_{\substack{j<i \\ r_{i,j}<r_{cutoff}}}^{N} \left(\frac{A_{i,j}}{r_{i,j}^{12}} - \frac{B_{i,j}}{r_{i,j}^{6}} \right)$$

$$U_{electrostatic} = \frac{1}{4\pi\varepsilon_o} \sum_{i=1}^{N} \sum_{j<i}^{N} \frac{q_i q_j}{r_{i,j}}$$

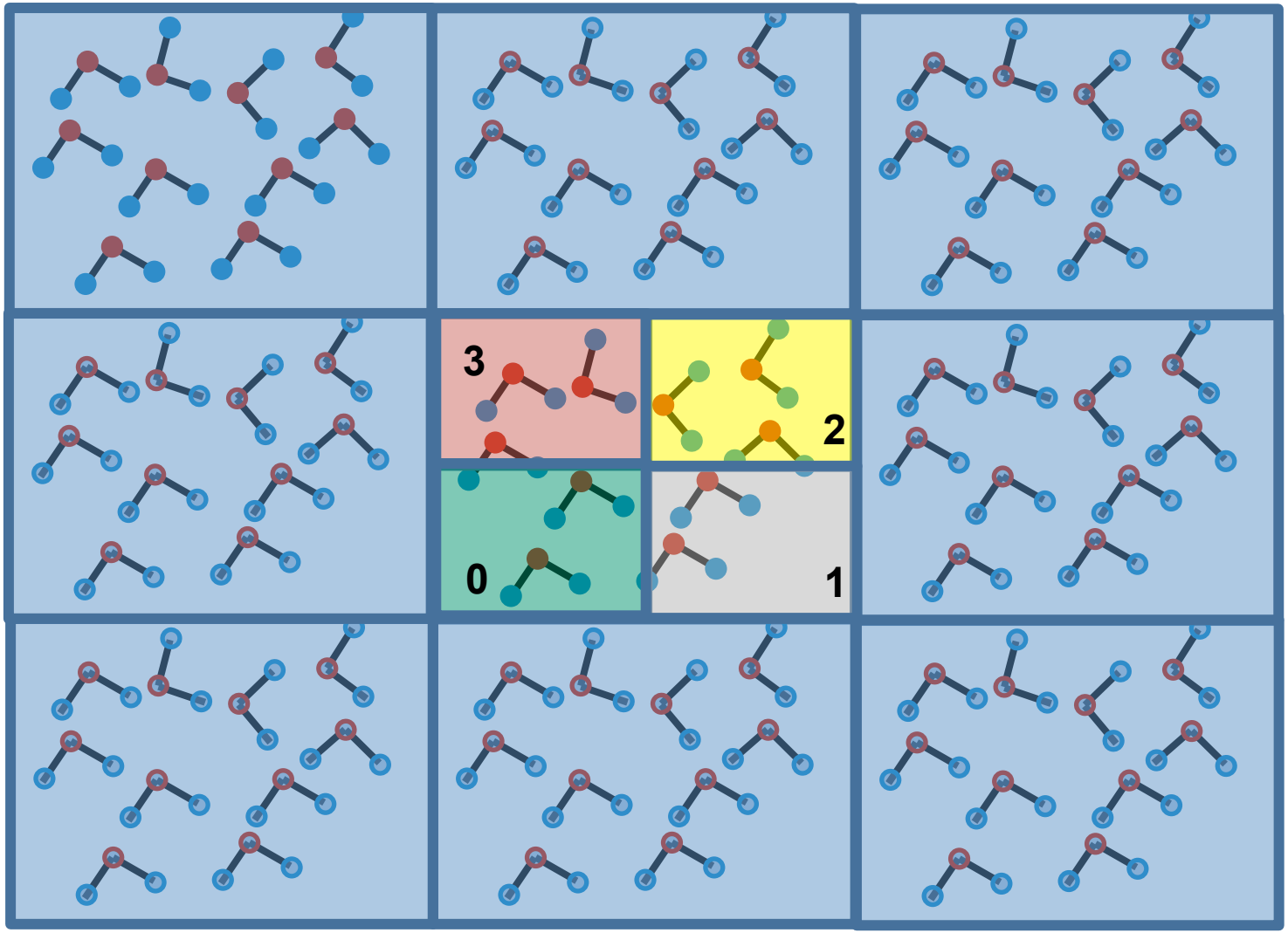
O = -q

H = +q/2

Periodic Boundary Conditions

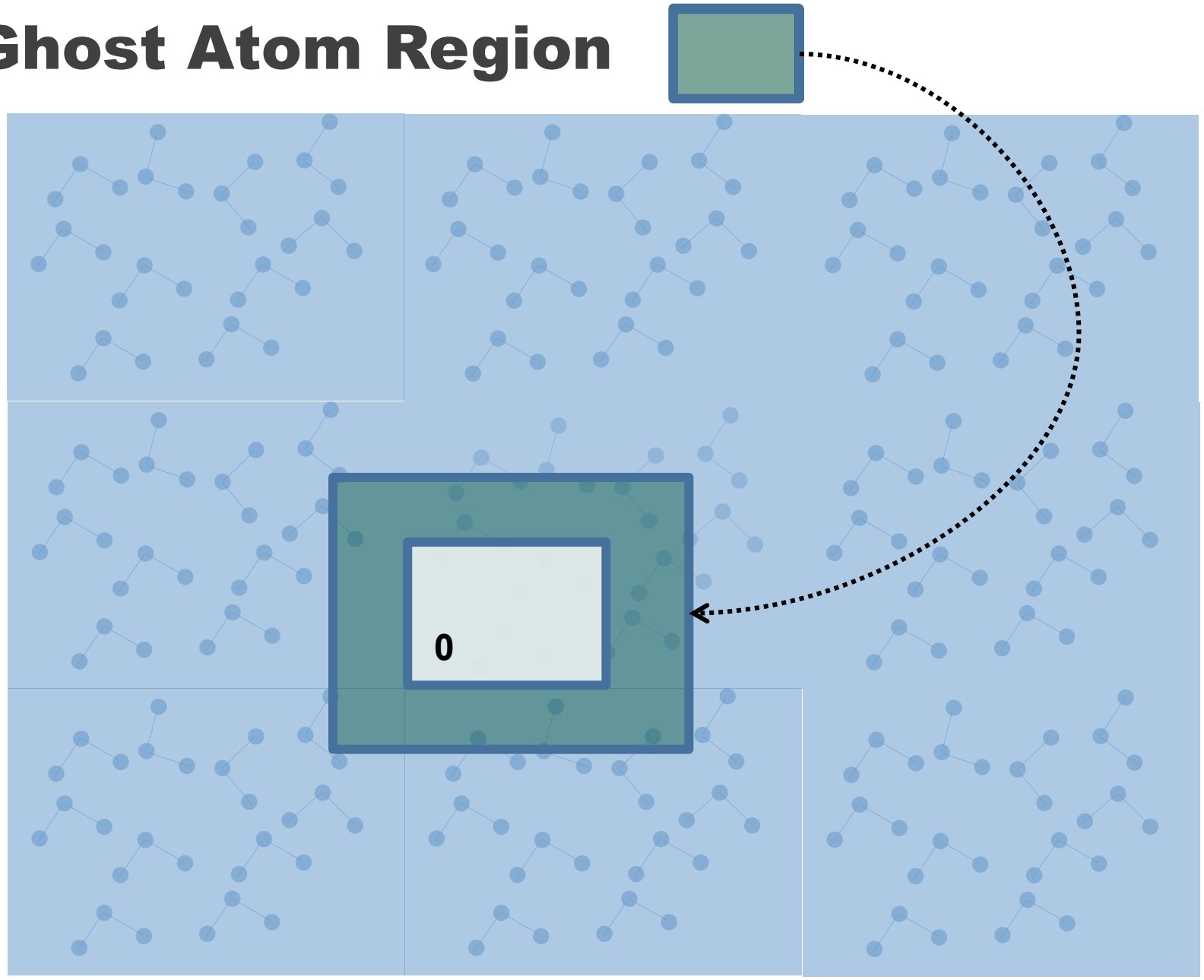


Spatial Decomposition

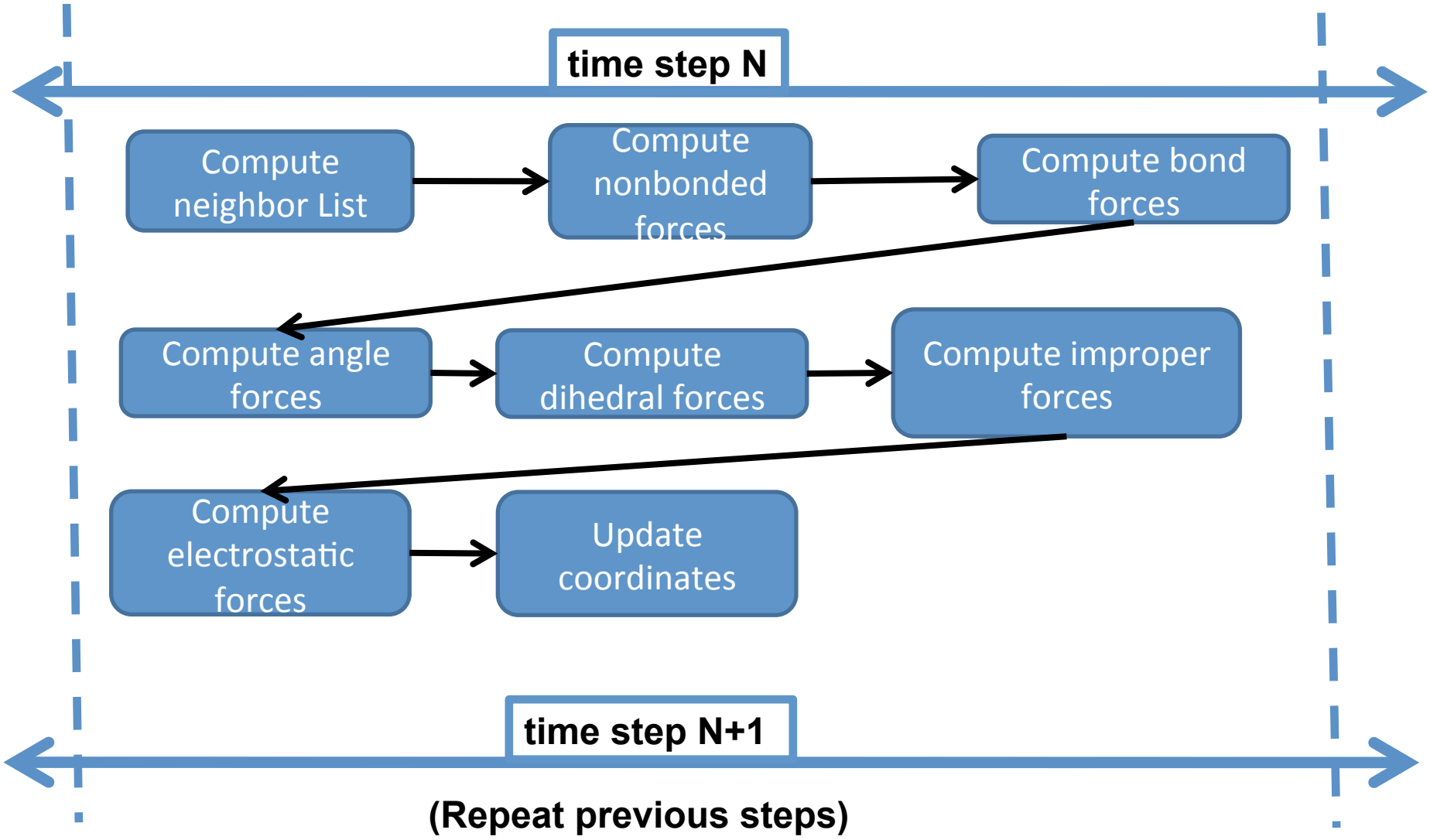


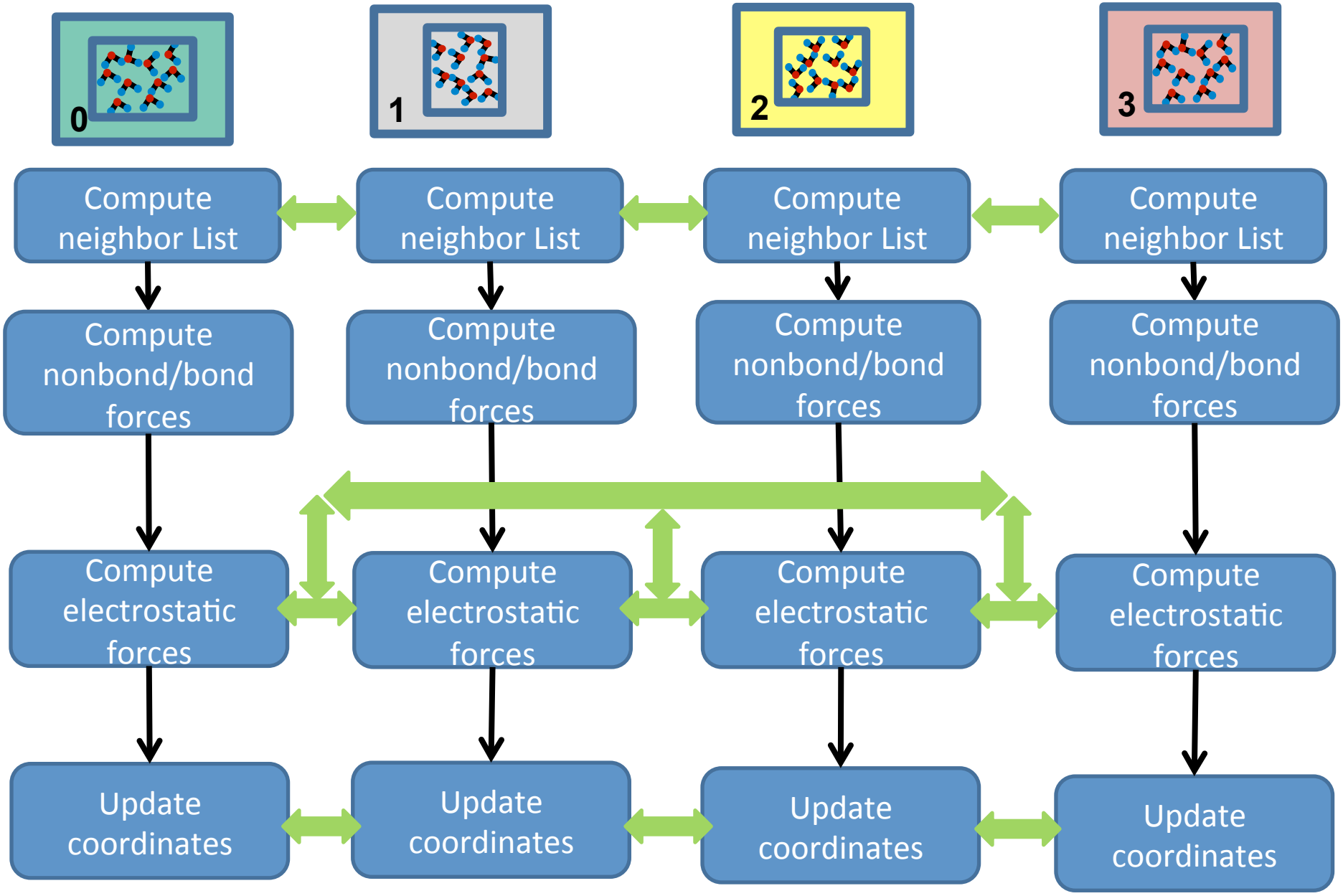
$$U_{nonbonded} = \sum_{i=1}^{N} \sum_{\substack{j<i \\ r_{i,j}<r_{cutoff}}}^{N} \left(\frac{A_{i,j}}{r_{i,j}^{12}} - \frac{B_{i,j}}{r_{i,j}^{6}} \right)$$

Ghost Atom Region



Force Calculation Implementation Within LAMMPS





The Force Terms $\vec{F}_i$

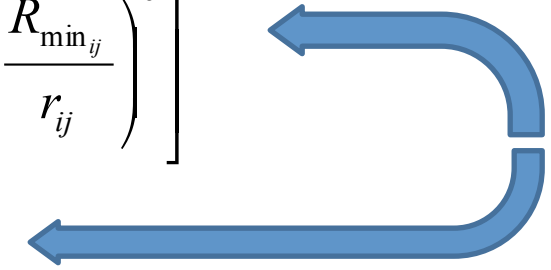
- **Approximately 80% of the total computational work is in computing the forces**

$$\vec{F}_i = m_i \vec{a}_i = -\frac{\partial U_{system}}{\partial \vec{r}_i}$$

$$U_{system} = \sum_{bonds} k_b (b - b_o)^2 + \sum_{angles} k_\Theta (\Theta - \Theta_o)^2 + \sum_{dihedrals} k_\phi (n\phi - \phi_o)^2 + \sum_{impropers} k_\omega (\omega - \omega_o)^2$$

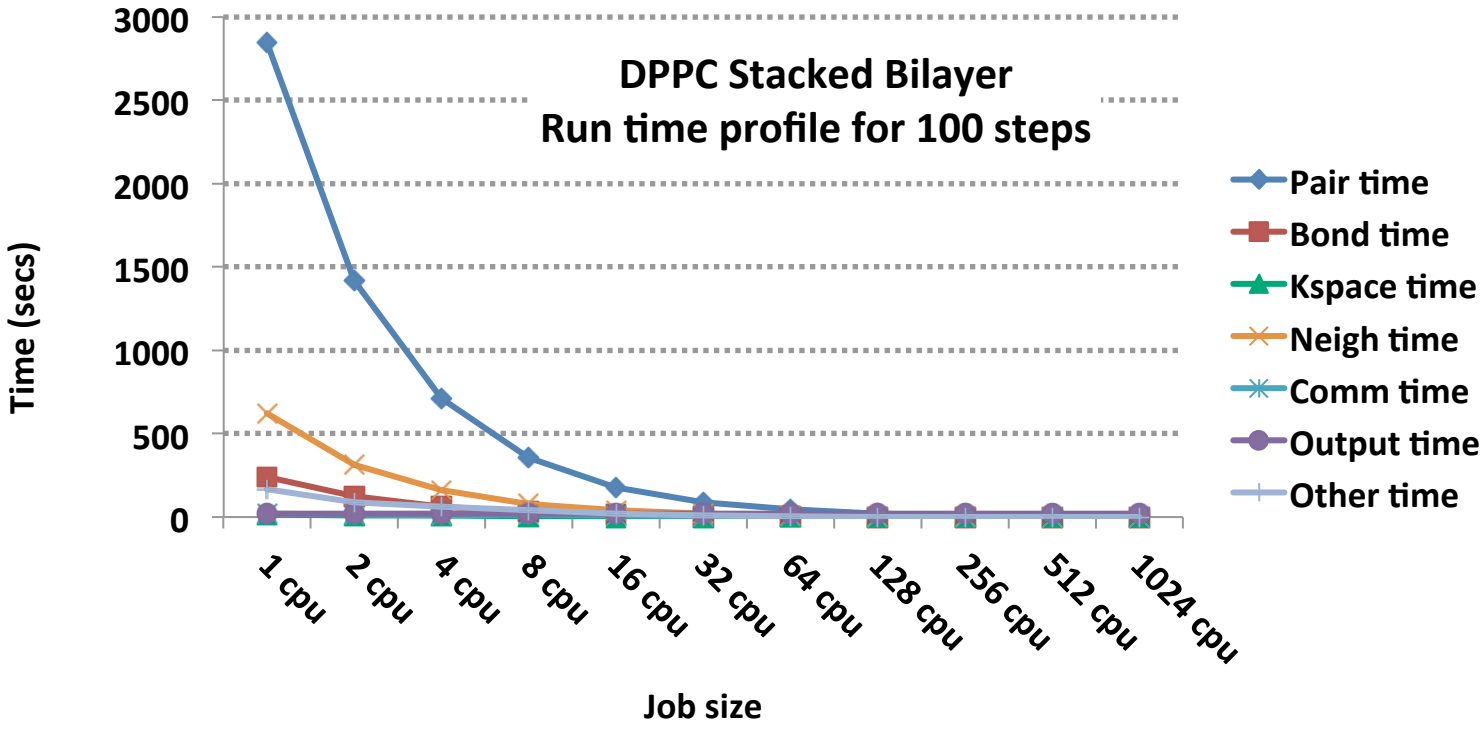
$$+ \sum_{nonbonded} \varepsilon \left[\left(\frac{R_{\min_{ij}}}{r_{ij}} \right)^{12} - \left(\frac{R_{\min_{ij}}}{r_{ij}} \right)^{6} \right]$$

$$+ \sum_i \sum_{i<j} \frac{1}{4\pi\varepsilon_o} \frac{q_i q_j}{r_{i,j}}$$



These two terms, nonbonded and electrostatic, respectively dominate the computational work

DPPC Stacked Bilayer Profile: 2 PF Problem

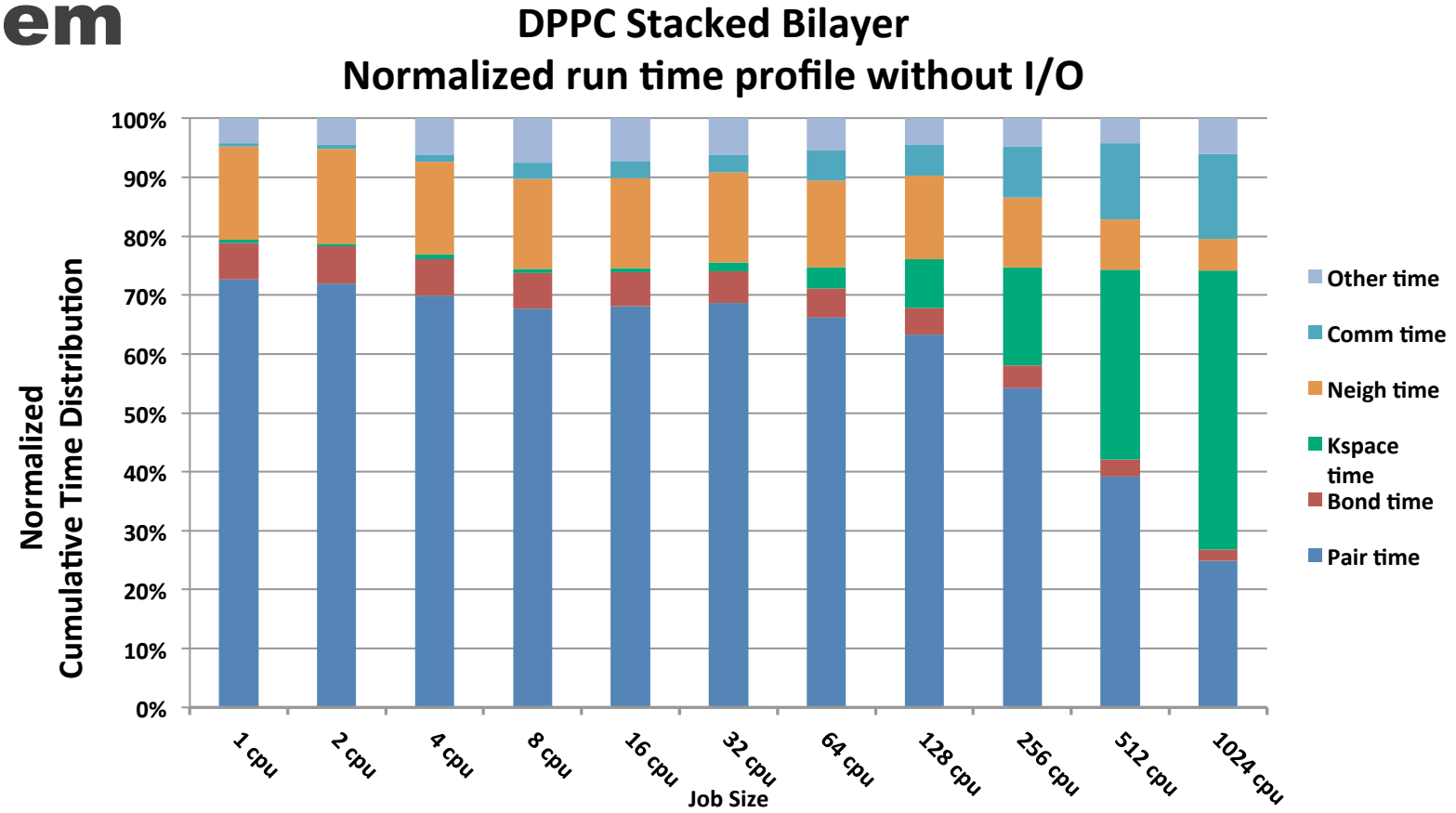


The pair component is part of the $U_{nonbonded} = \sum_{i} \sum_{i \neq j} \varepsilon \left[\left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{12} - \left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{6} \right]$ interactions

The Kspace component is part of the $U_{electrostatic} = \sum_{i} \sum_{i \neq j} \frac{1}{4\pi\varepsilon_o} \frac{q_i q_j}{r_{i,j}}$ interactions

These two terms, nonbonded and electrostatic, respectively dominate the computational work

DPPC Stacked Bilayer Profile: 2 PF Problem

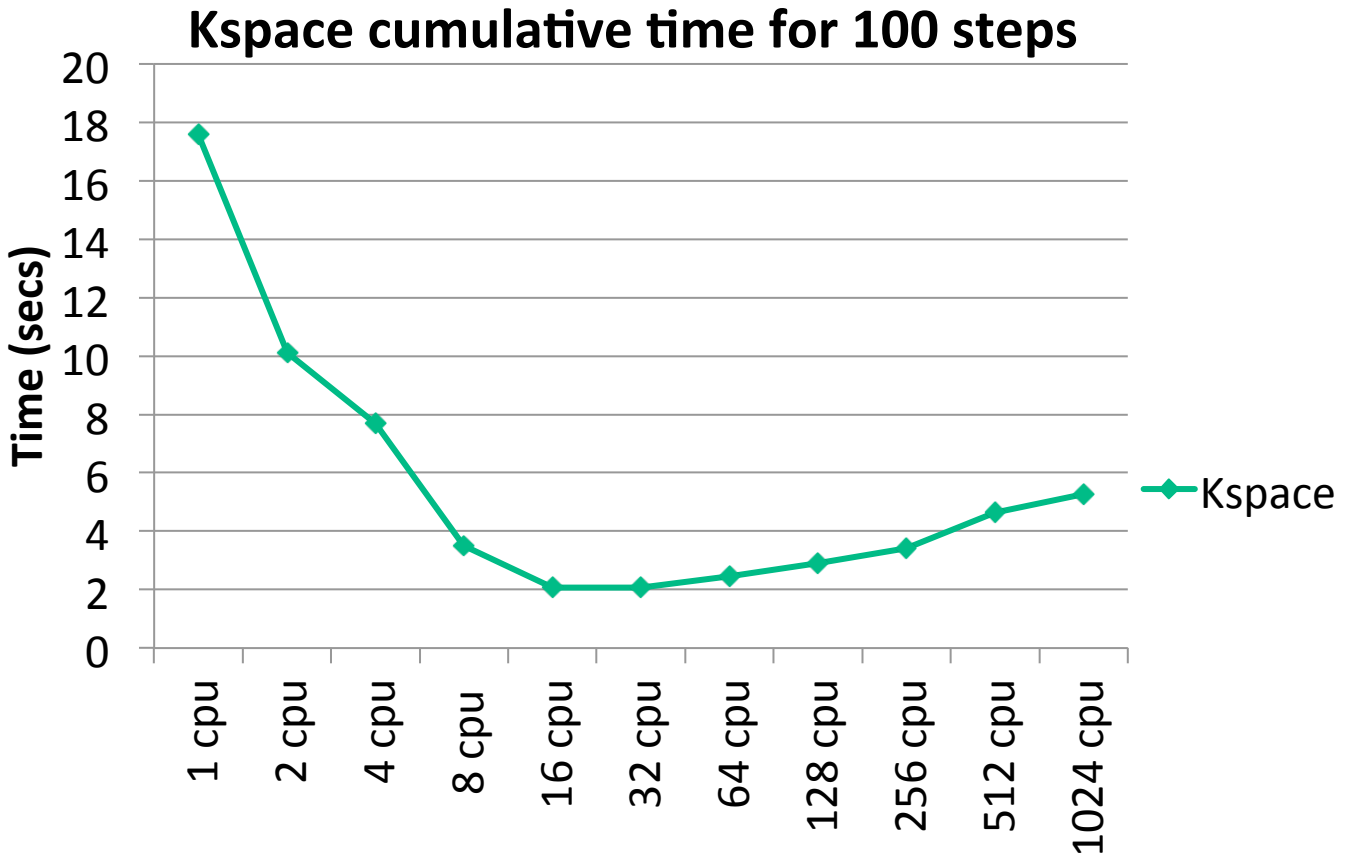


The pair component is part of the $U_{nonbonded} = \sum_{i} \sum_{i \neq j} \varepsilon \left[\left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{12} - \left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{6} \right]$ interactions

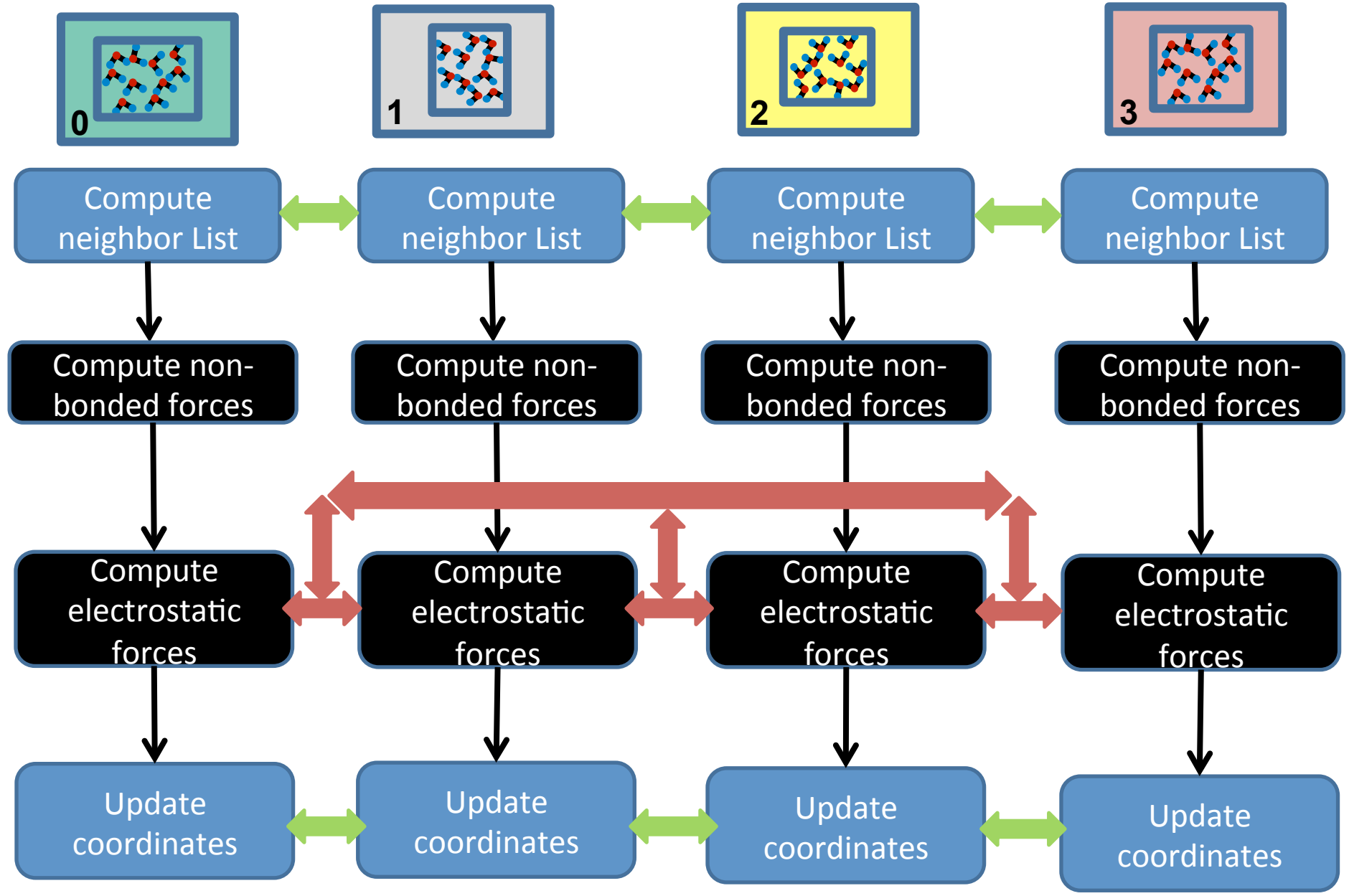
The Kspace component is part of the $U_{electrostatic} = \sum_{i} \sum_{i \neq j} \frac{1}{4\pi\varepsilon_o} \frac{q_i q_j}{r_{i,j}}$ interactions

These two terms, nonbonded and electrostatic, respectively dominate the computational work

DPPC Stacked Bilayer Profile: 2 PF Problem



The Kspace component is part of the $U_{electrostatic} = \sum_{i} \sum_{i \neq j} \frac{1}{4\pi\varepsilon_o} \frac{q_i q_j}{r_{i,j}}$ interactions



3. Accelerating LAMMPS

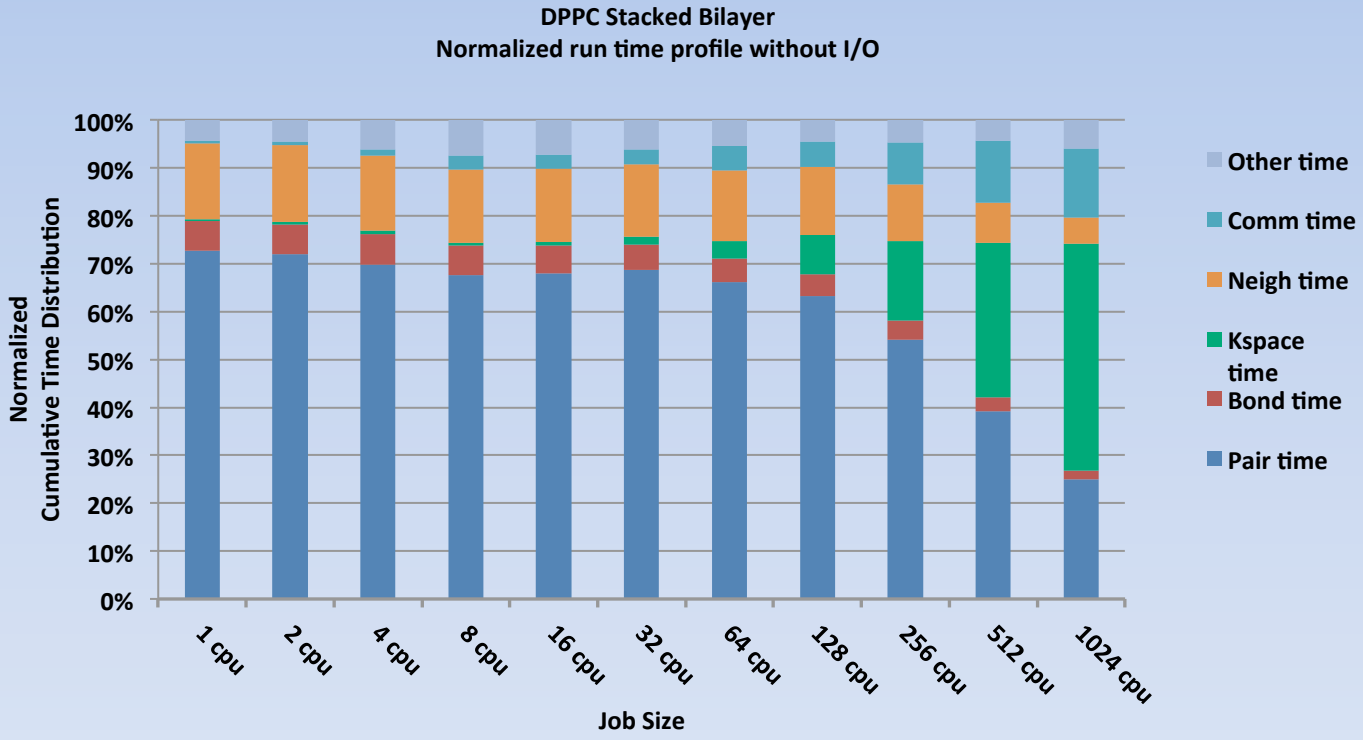
Need to accelerate the nonbonded forces

The pair component is part of the $U_{nonbonded} = \sum_{i} \sum_{i \neq j} \varepsilon \left[\left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{12} - \left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{6} \right]$ interactions

and the Kspace component of the electrostatic forces

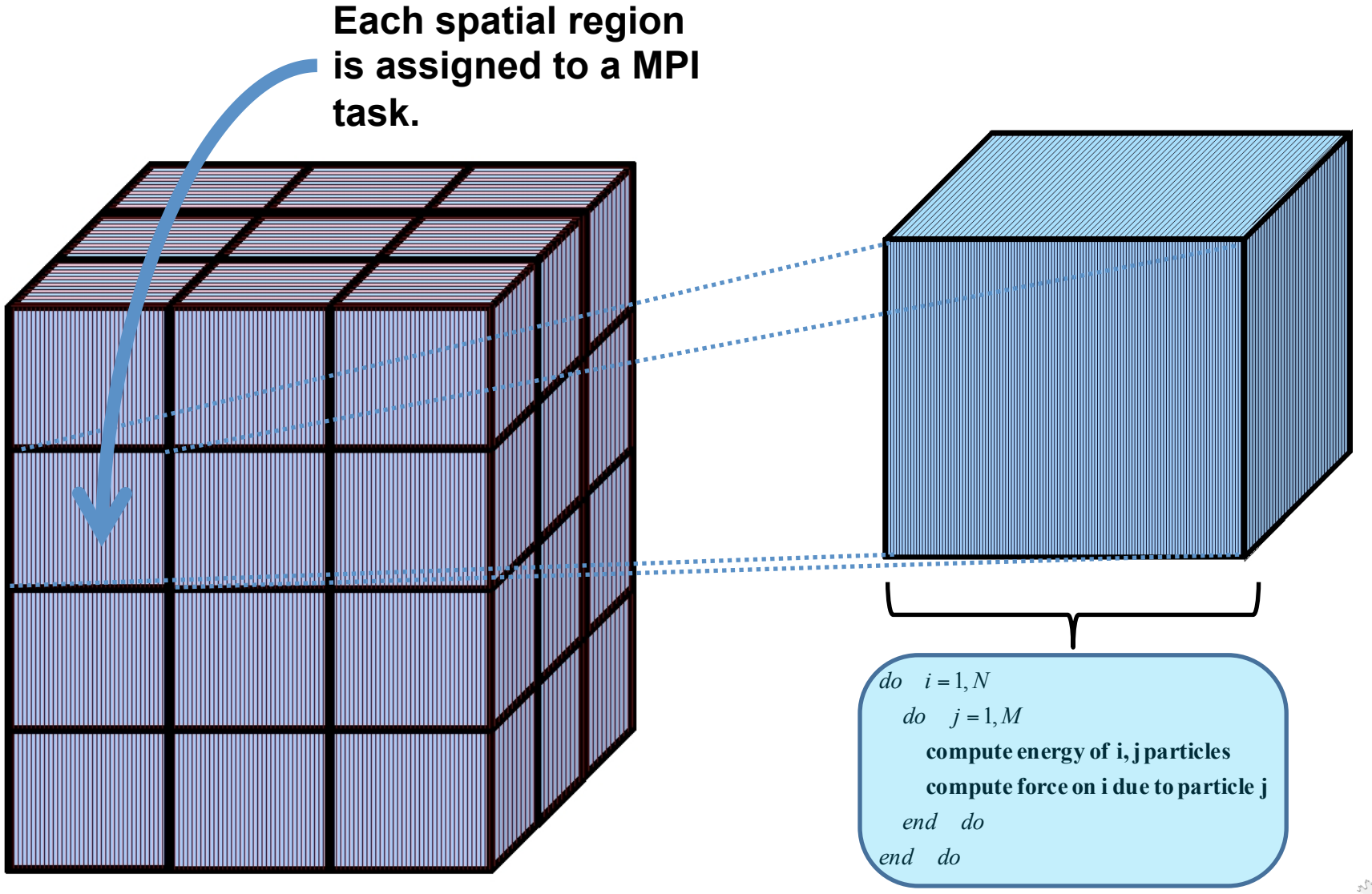
The Kspace component is part of the $U_{electrostatic} = \sum_{i} \sum_{i \neq j} \frac{1}{4\pi\varepsilon_o} \frac{q_i q_j}{r_{i,j}}$ interactions

Accelerating the Nonbonded Interactions within LAMMPS



The pair component is part of the $U_{nonbonded} = \sum_{i} \sum_{i \neq j} \varepsilon \left[\left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{12} - \left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{6} \right]$ interactions

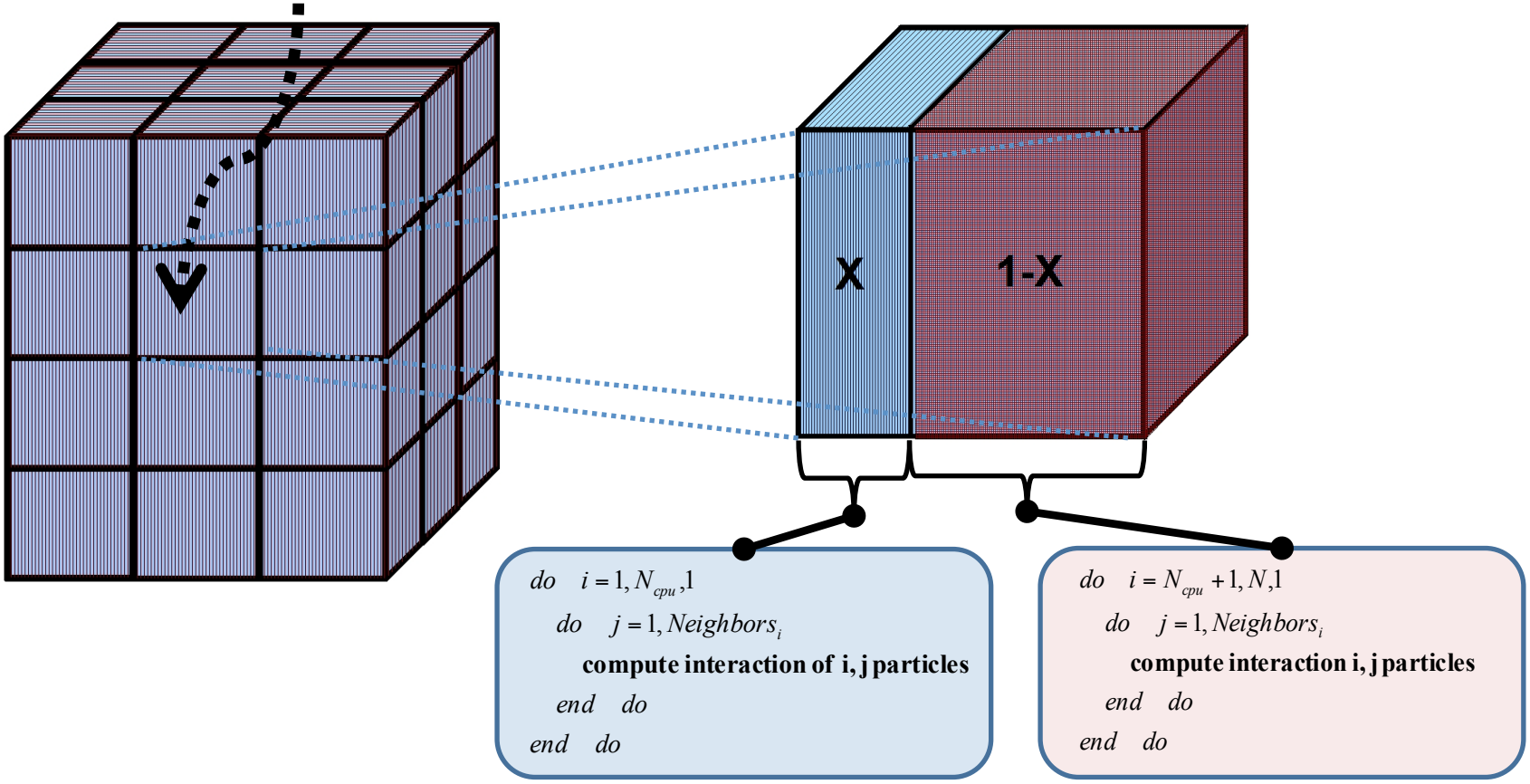
Calculation of Nonbonded Interactions (Current CPU Only Implementation)



Calculation of Non-bonded Interactions Accelerated Implementation (Planned Work)

- Spatial region maps to a node.
- Spatial region hosts a MPI task
- Particles in the region will be evolved by CPUs and GPUs attached to this node

- A *numerical* fraction X of the N particles will be assigned the CPUs
- The remaining *numerical* fraction (1-X) will be assign to the accelerator

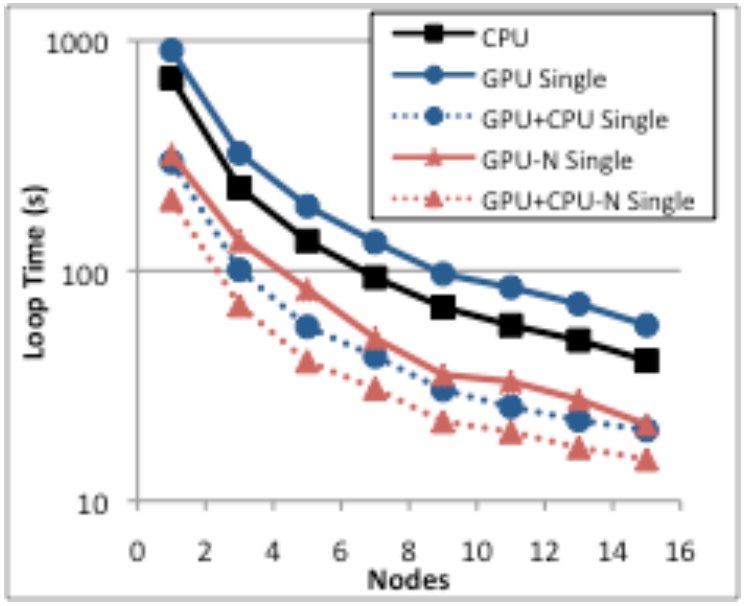


Done on host CPUs

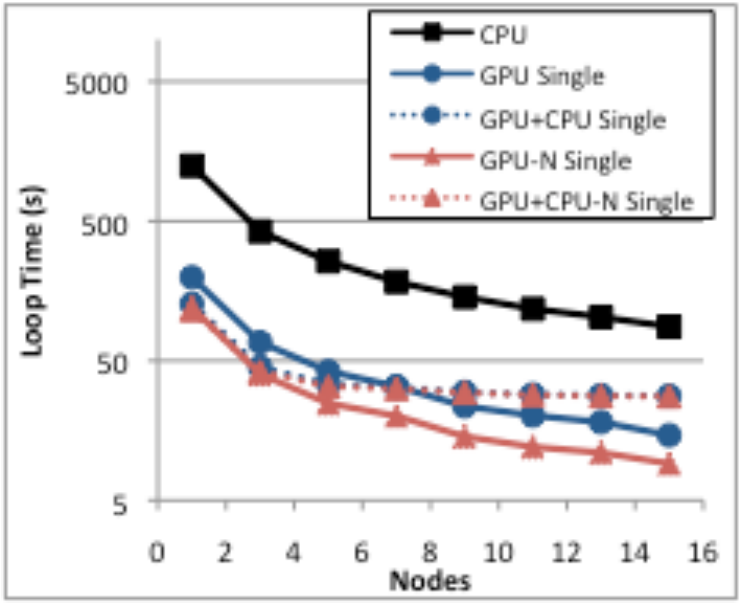
Asynchronously done on GPUs

Load Balancing of Nonbonded Interactions

- Multiple cores can be assigned to each GPU
- Dynamic load balancing assigns particles to CPU and GPU for asynchronous force calculation
  - Not the only benefit, splitting per GPU data into smaller subdomains improves memory latencies and splits "other" time (integration, etc.) across more cores



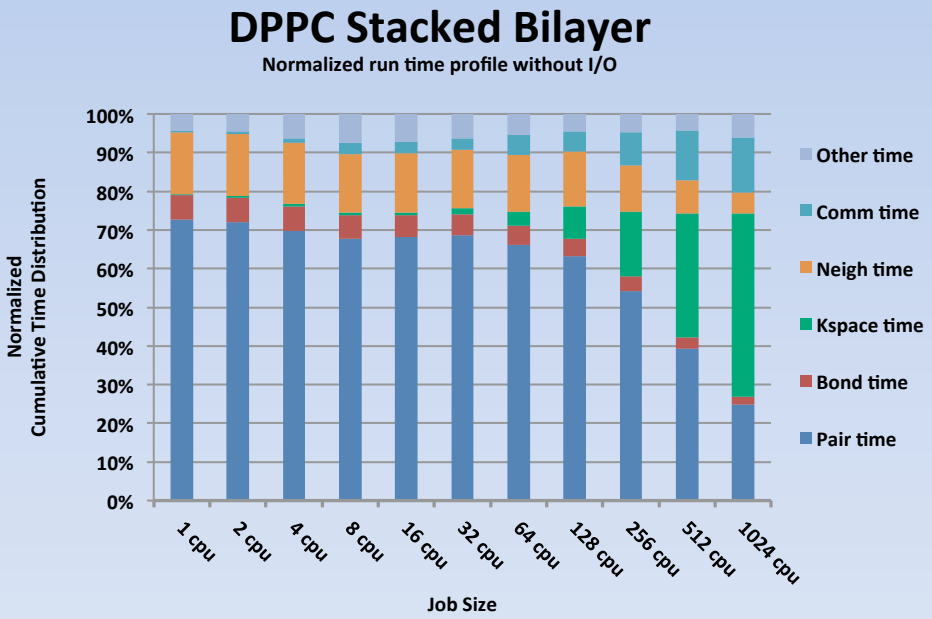
Lennard-Jones kernel (*864k Particles*)
cutoff=2.5 x radius (~78 neighbors)
Low Arithmetic Intensity

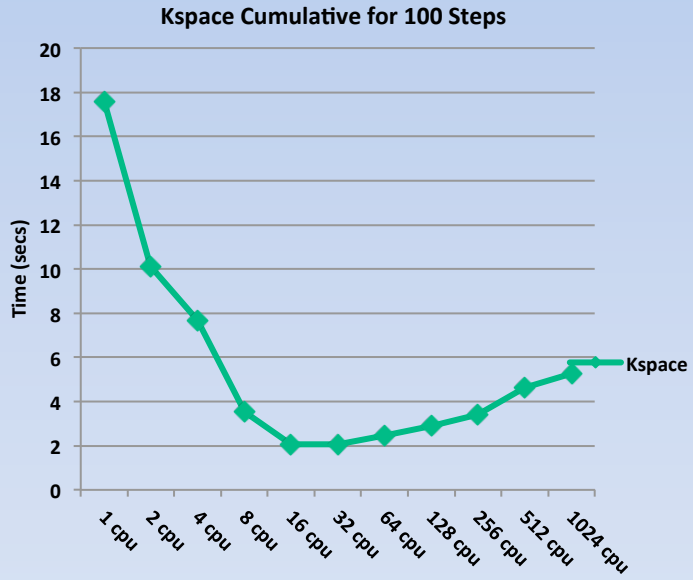


Gay-Berne kernel *(125k Particles)*
cutoff=7 x short radius
High Arithmetic Intensity

Dual Hex-Core AMD Istanbul 2.6 GHz, Dual Fermi C2050, Infiniband

Acceleration of the Electrostatic Interactions by Means of Multilevel Summation Method (MSM)





The Kspace component is part of the $U_{electrostatic} = \sum_{i} \sum_{i \neq j} \frac{1}{4\pi\varepsilon_o} \frac{q_i q_j}{r_{i,j}}$ interactions

Multilevel Summation Method (MSM) vs. Particle Mesh Ewald (PME)

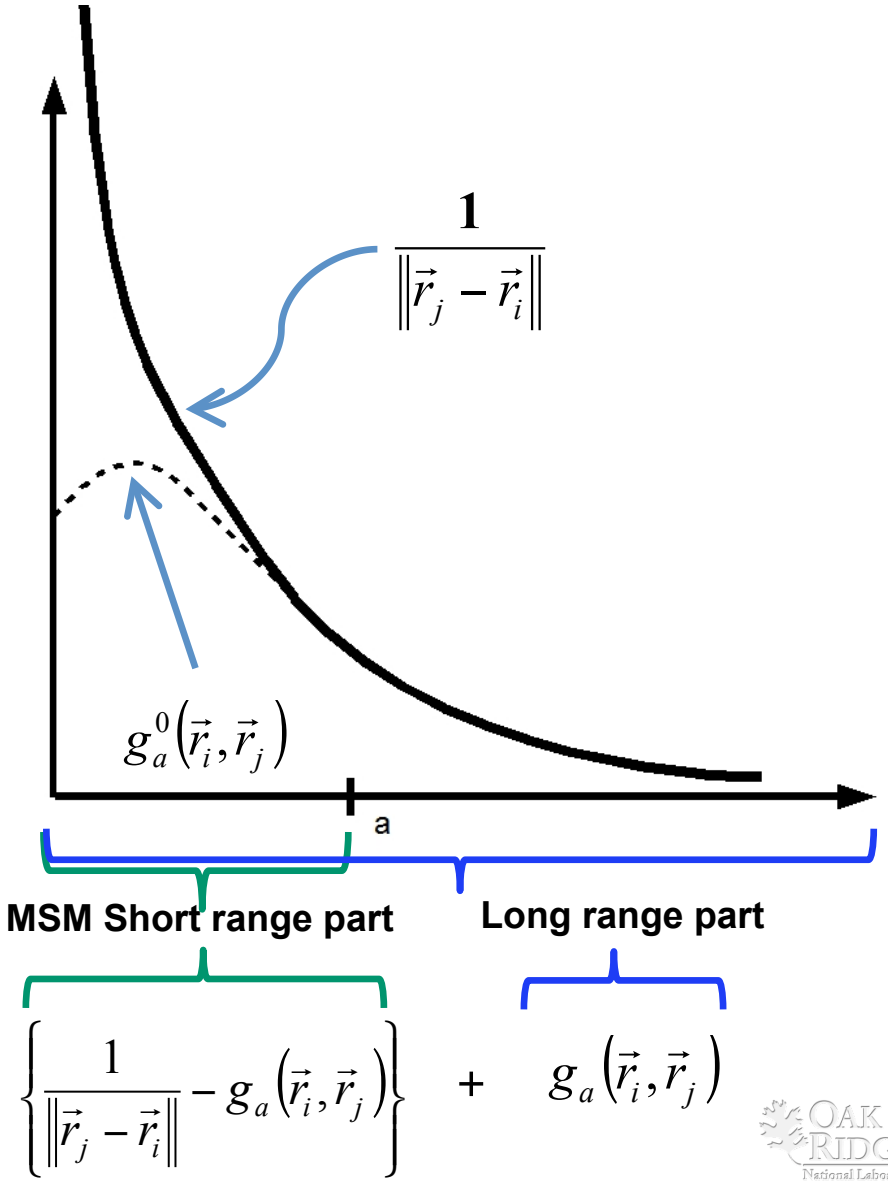
- LAMMPS currently uses the PME algorithm solve the electrostatic interactions

- MSM is more scalable than PME
  - MSM is grid based method whose communication pattern is nearest neighbor while PME involves 2 ALL-TO-ALLs over reciprocal lattice points to compute 2 3d FFTs (Hardy, 2009)
  - MSM scale like O(N) whereas PME scales as O(NlnN)
  - For typical systems, the crossover between MSM and PME is approximately 100,000 atoms

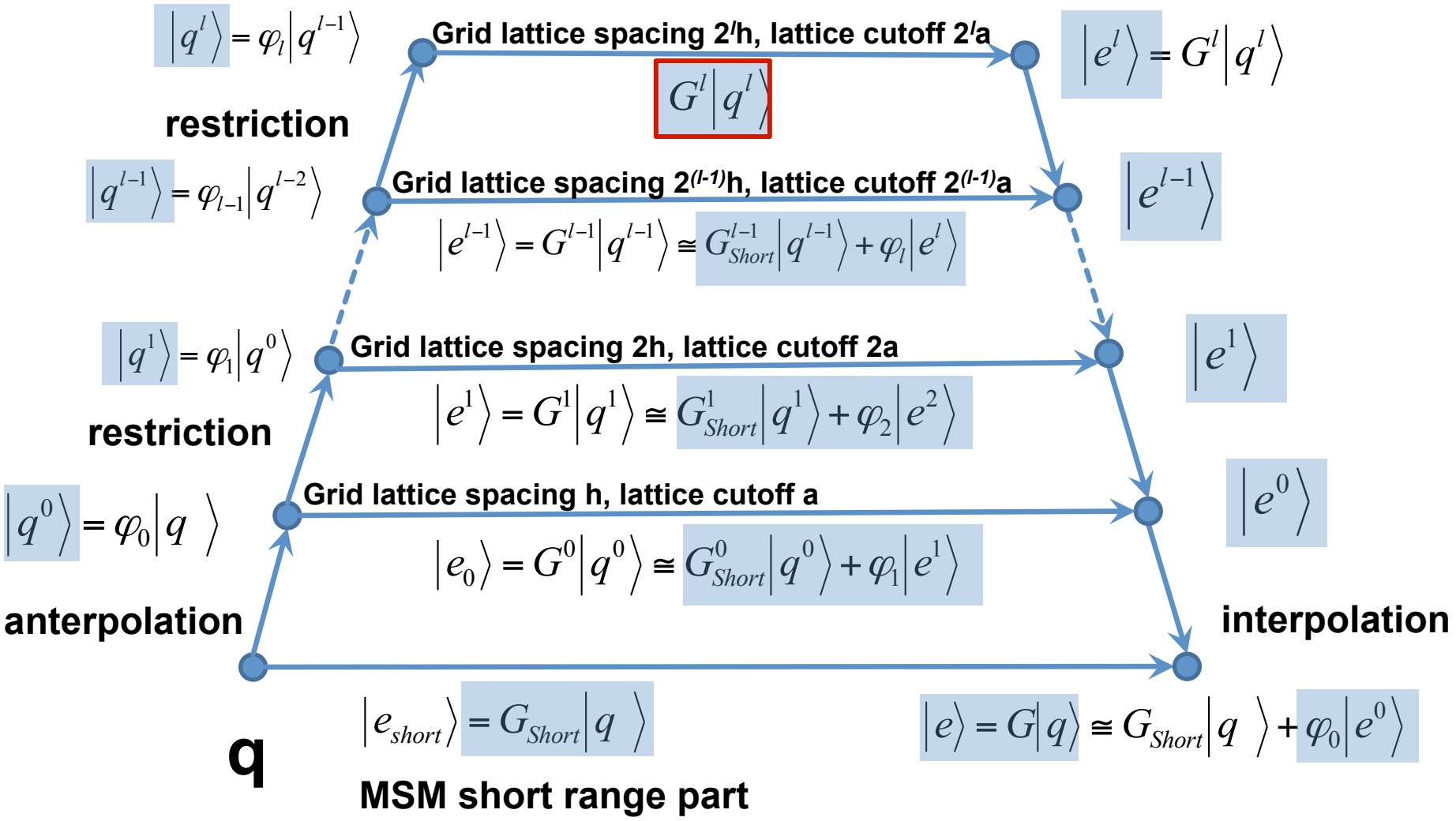
- MSM is essential for the scalability of large problems like the ones proposed for OLCF3

Multilevel Summation Method Overview

1. Separation of the electrostatic interaction potential into a MSM short range part plus a MSM long range, slowly varying part
2. Approximate the slowly varying MSM long range part to a grid.
3. Recursive application of the first two ideas on a hierarchy of coarser grids (Hardy, 2006, p. 12).



MSM Algorithmic Steps



MSM Direct

$$\textbf{for } (i_c, j_c, k_c) \in \{\Omega^n indices\}$$

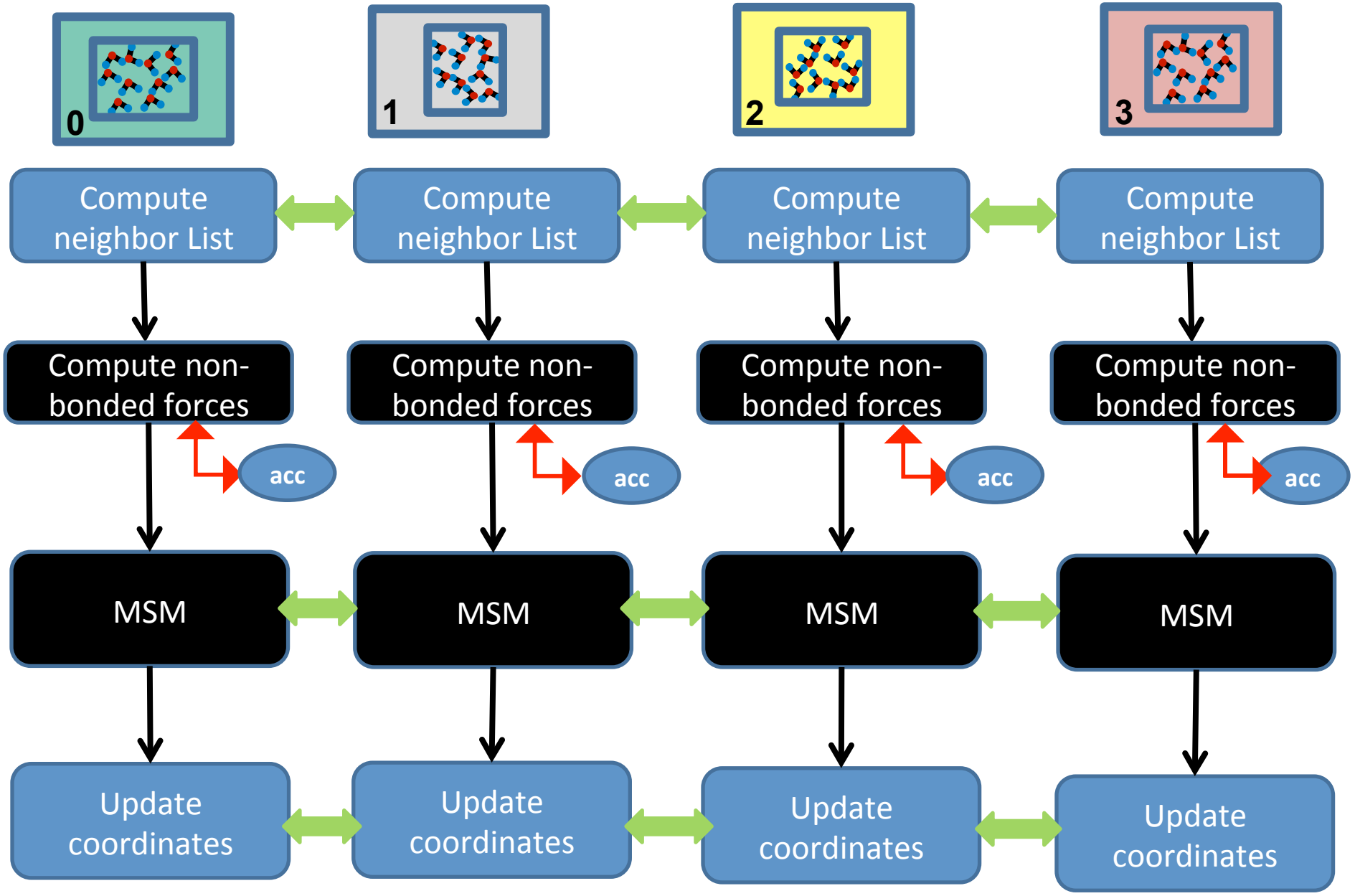
$$e^{n,short}[i_c, j_c, k_c] = 0$$

$$\textbf{for } k = \left\lfloor -\frac{2a}{h_z} \right\rfloor, \ldots, \left\lfloor \frac{2a}{h_z} \right\rfloor$$

$$\textbf{for } j = \left\lfloor -\frac{2a}{h_y} \right\rfloor, \ldots, \left\lfloor \frac{2a}{h_y} \right\rfloor$$

$$\textbf{for } i = \left\lfloor -\frac{2a}{h_x} \right\rfloor, \ldots, \left\lfloor \frac{2a}{h_x} \right\rfloor$$

$$e^{n,short}[i_c, j_c, k_c] += \left(g^n\left(\mathbf{0}, 2^n h(i,j,k)\right) - g^{n+1}\left(\mathbf{0}, 2^n h(i,j,k)\right)\right) \times q^n[i_c + i, j_c + j, k_c + k]$$



- Need non-trivial knowledge of what the code is doing
  - Algorithmic details
    - Use a debugger to step through the code
  - Problem/data decomposition
  - Interproccessor data communication patterns
- Profile the code to discover bottlenecks
  - VampirTrace, CrayPat, homemade timers. etc.
- "There is no free lunch for porting to GPUs
  - (Except for those who registered for this workshop)
  - There is a ***significant communication cost*** with the accelerator; some algorithms may not be computational intense enough to be worth this cost.

Summary

- We have identified the major challenges and have a technologically reasonable plan for accelerating LAMMPS
  - Nonbonded interactions
    - Capturing various levels of parallelism on CPU and GPU
    - Load balancing between CPU and GPU
  - Electrostatic interactions
    - Developing the MSM algorithm
      - Capturing various levels of parallelism on CPU and GPU
      - Load balancing between CPU and GPU
  - Regardless of the accelerator, we are forced to restructure some parts LAMMPS to effectively capture the inter and intra node parallelism

Acknowledgements

- Bobby Phillips and Richard Mills – Multi-grid discussions
- Jim Schwarzmeier and Sarah Anderson – Algorithmic discussions, benchmarking, ….
- Scott Hampton and Pratul Agarwal – CUDA and algorithmic discussions

References

- **Hardy, David.** "Multilevel Summation For the Fast Evaluation of Forces for the Simulation of Biomolecules" Ph.D. diss., University of Illinois at Urbana-Champaign, 2006.
- **Hardy, David., Stone John., and Schulten K.** 2009. Multilevel Summation of Electrostatic Potentials Using Graphics Processing Units. Parallel Computing vol 35: pgs 164-177.

- Team Members and Expertise
  - Oak Ridge National Laboratories
    - Arnold Tharrington - Computational biophysicist, primary programmer for code modifications
    - Michael Brown - Computational biophysicist, primary programmer for code modifications
    - Bobby Philips - Mathematician and multigrid expert
  - Sandia National Laboratories
    - Steve Plimpton - Computational scientist, originator of LAMMPS
    - Paul Crozier - Computational scientist, LAMMPS developer
  - NVIDIA
    - Peng Wang - Computational scientist and CUDA expert
  - Cray Inc.
    - James Schwarzmeier - Computational scientist, benchmarking, code profile and optimization expert, threading, etc.
    - Sarah Anderson - Computational Scientist, benchmarking, code profile and optimization expert, threading, etc.
  - Institute for Computational Molecular Science at Temple University (Director: Michael Klein)
    - Axel Kohlmeyer - Computational Biophysicist, science lead for 20 PF target problem

Supplementary Slides

Energy and Forces

$$U(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N) \cong \frac{1}{2}\langle q|G_{Short}|q\rangle + \frac{1}{2}\langle q|[\![\varphi_0]\!]^T|e^0\rangle$$

$$F_{l\alpha} \cong \frac{1}{2}\left\langle q\left|\frac{\partial}{\partial r_{l\alpha}}G_{Short}\right|q\right\rangle - q_l\left\langle e^0\left|\frac{\partial}{\partial r_{l\alpha}}\varphi(r_l)\right|\right\rangle$$

$$\textbf{where}\ |e^0\rangle = G^0|q\rangle \cong G^0_{Short}|q^0\rangle + \varphi_1|e^1\rangle$$

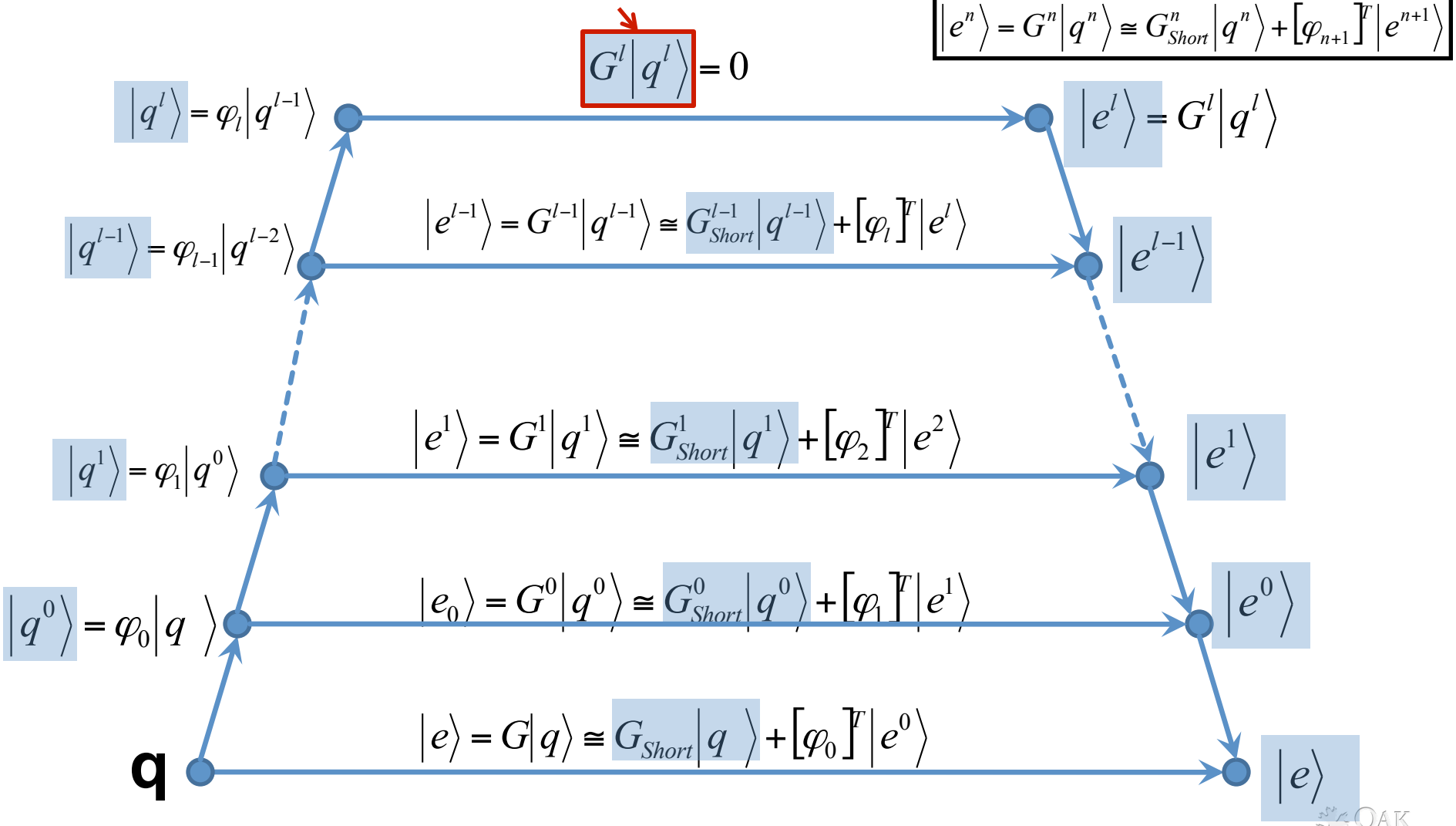
Matrix Vector Decomposition

This product is zero for neutral charged systems

$$G^l \left| q^l \right\rangle = 0$$

$$\left| q^{n+1} \right\rangle = \varphi_{n+1} \left| q^n \right\rangle$$

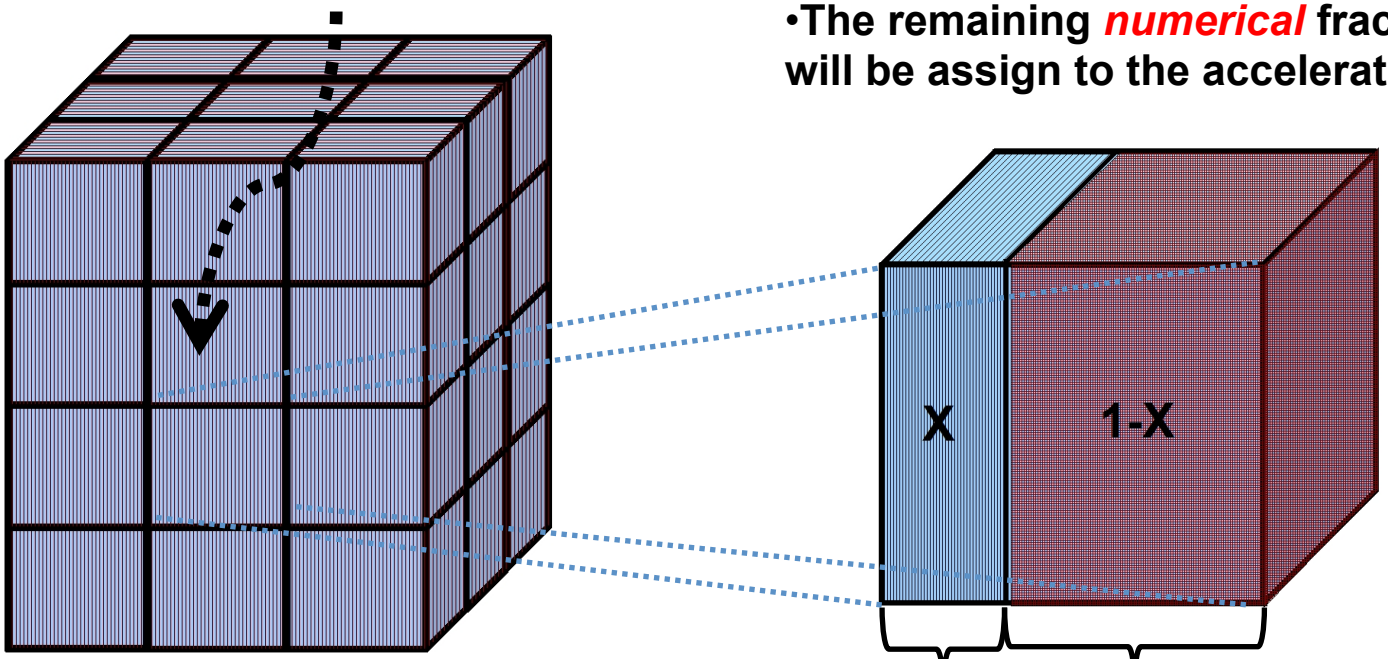
$$\left| e^n \right\rangle = G^n \left| q^n \right\rangle \cong G^n_{Short} \left| q^n \right\rangle + \left[\varphi_{n+1} \right]^T \left| e^{n+1} \right\rangle$$



Calculation of MSM Short Range Part

- Spatial region maps to a node.
- Spatial region hosts a MPI task
- Particles in the region will be evolved by CPUs and GPUs attached to this node

- A *numerical* fraction X of the N particles will be assigned the CPUs
- The remaining *numerical* fraction (1-X) will be assign to the accelerator




```
do   i = 1, N_cpu, 1
  do   j = 1, Neighbors_i
    compute interaction of i, j particles
  end   do
end   do
```

Done on host CPUs

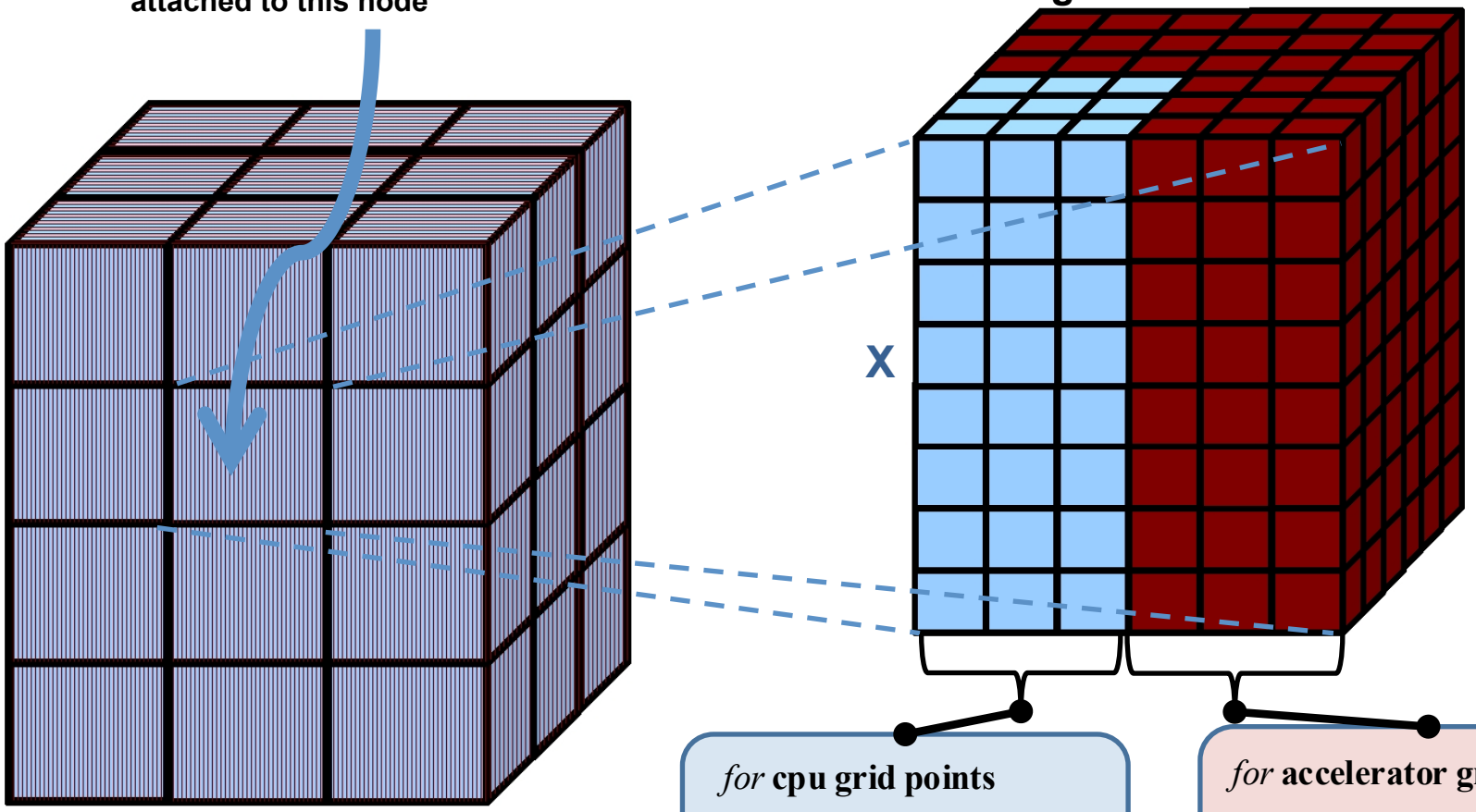
```
do   i = N_cpu + 1, N, 1
  do   j = 1, Neighbors_i
    compute interaction i, j particles
  end   do
end   do
```

Asynchronously done on GPUs.

MSM Grid Computations

•Spatial region maps to a node.
•Spatial region hosts a MPI task
•Particles in the region will be evolved by CPUs and GPUs attached to this node

•A *numerical* fraction X of the grid points will be assigned the CPUs
•The remaining *numerical* fraction (1-X) will be assign to the accelerator



K^n total grid points for level n

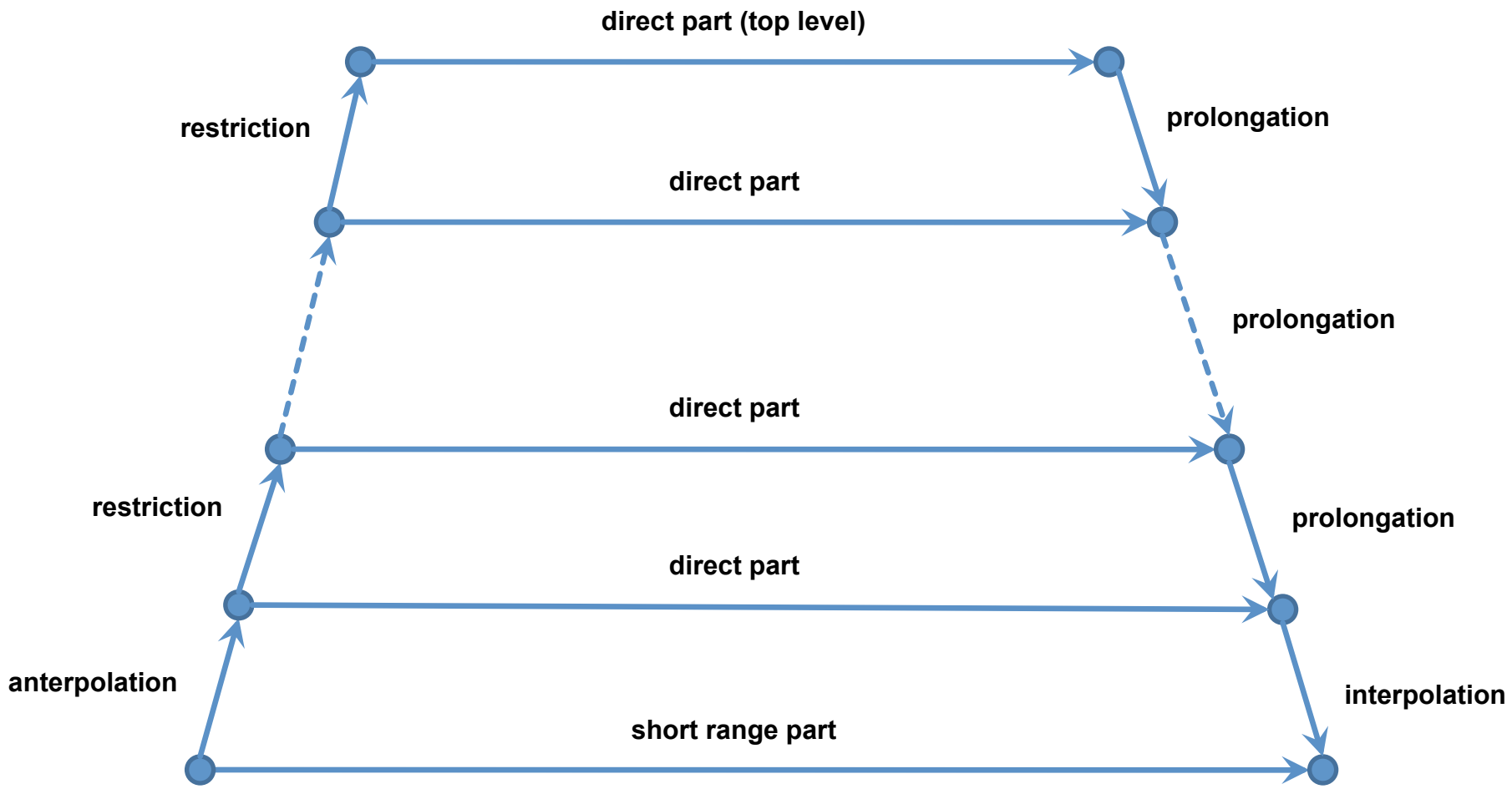
```
for cpu grid points
    compute on grid point i
end do
```

Done on host CPUs

```
for accelerator grid points
    compute on grid point i
end do
```

Done on GPUs

Algorithmic Steps



Particle Level

Operational Counts

- Short range part $\sim (a/h)^3 N$
- Anterpolation $\sim (p^3 + 7p^2 + 9p + 6)N$
- Direct part $\sim (a/h)^3 K_n$
- Interpolation $\sim (5p^3 + 24p^2 + 27p + 11)N + K_0$
- Restriction and Prolongation $\sim (2p+2)(1-1/8^{\ell})K_0$

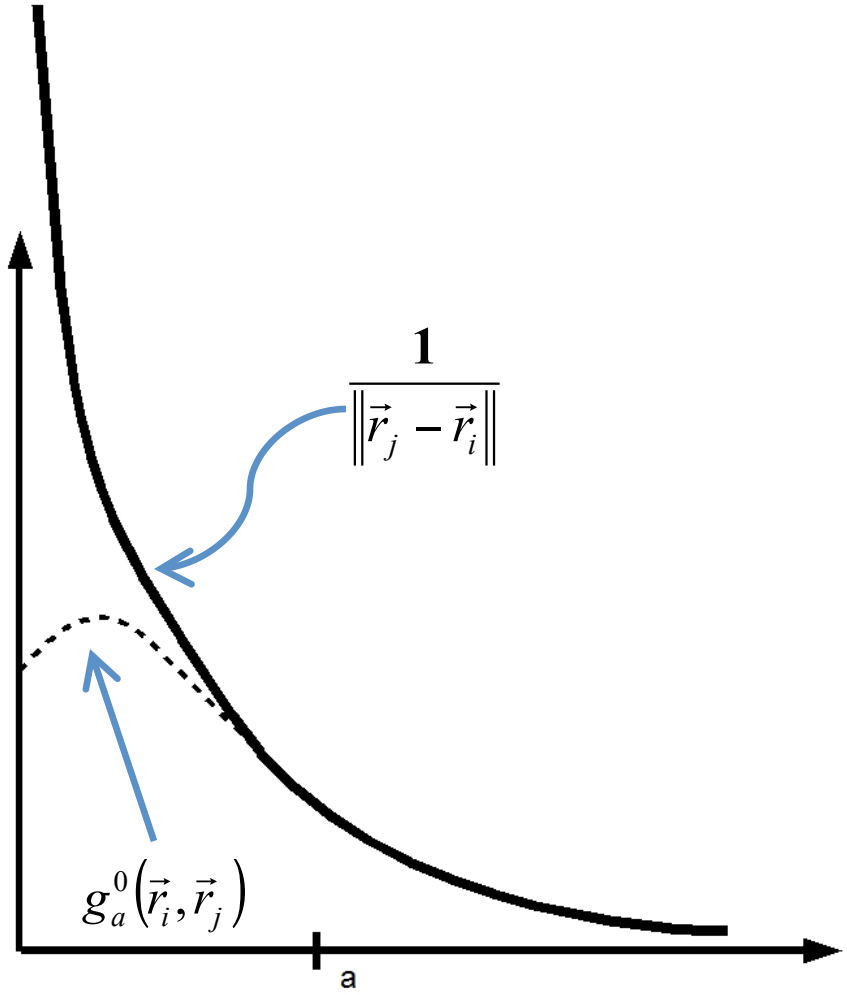
Hardy, J. 2006 Multilevel Summation for the Fast Evaluation of Forces For The Simulation of Biomolecules. PhD Thesis, University of Illinois at Urbana-Champaign.

Splitting Operation

$$\left\{\frac{1}{\left\|\vec{r}_j-\vec{r}_i\right\|}-g_a\left(\vec{r}_i,\vec{r}_j\right)\right\}+g_a\left(\vec{r}_i,\vec{r}_j\right)$$

Short range

Long range



Splitting Operation (cont.)

$$G_{Long} = \begin{cases} g_a\left(\vec{r}_i, \vec{r}_j\right), & i \neq j \\ 0, & i = j \end{cases}$$

$$G_{Short} = \begin{cases} \dfrac{1}{\left\|\vec{r}_j - \vec{r}_i\right\|} - g_a\left(\vec{r}_i, \vec{r}_j\right), & j \notin \chi(i) \text{ and } i \neq j \\ -g_a^0\left(\vec{r}_i, \vec{r}_j\right), & j \in \chi(i) \text{ and } i \neq j \\ 0, & i = j \end{cases}$$

where

$$g_a^n\left(\vec{r}_i, \vec{r}_j\right) \equiv \frac{1}{2^n a}\gamma\left(\frac{\left\|\vec{r}_j - \vec{r}_i\right\|}{2^n a}\right), \text{ and } \gamma(\rho) = \frac{1}{\rho} \text{ for } \rho \geq 1$$

Splitting Operation (cont.)

$$\left\{\frac{1}{\left\|\vec{r}_j-\vec{r}_i\right\|}-g_a\left(\vec{r}_i,\vec{r}_j\right)\right\}+g_a\left(\vec{r}_i,\vec{r}_j\right)$$

convolve charges to grid of length a; split long range part

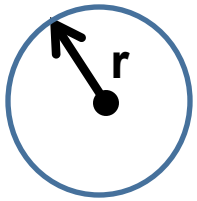
$$\left\{g_a\left(\vec{r}_i,\vec{r}_j\right)-g_{2a}\left(\vec{r}_i,\vec{r}_j\right)\right\}+g_{2a}\left(\vec{r}_i,\vec{r}_j\right)$$

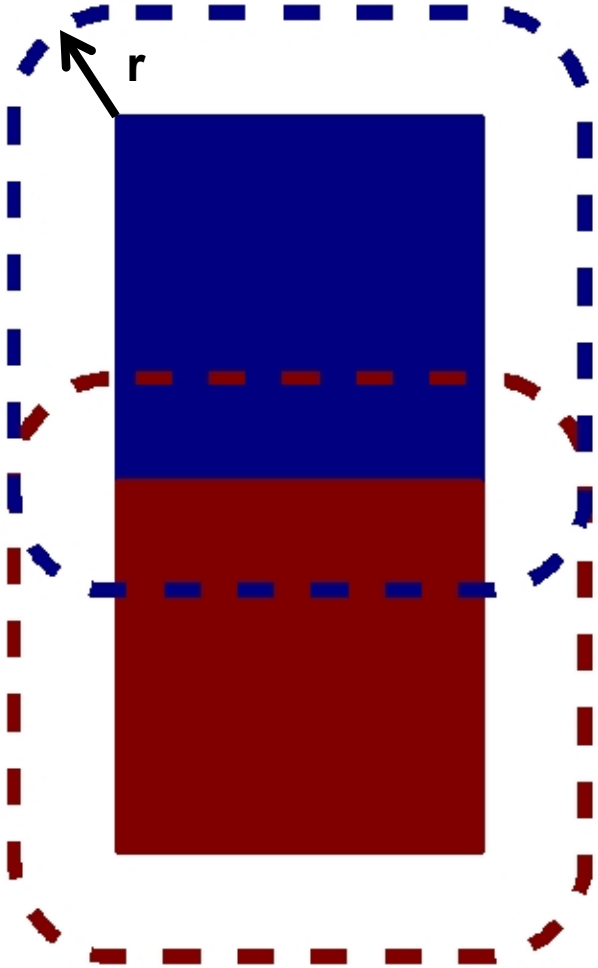
convolve charges to grid of length 2a; split long range part

$$\left\{g_{2a}\left(\vec{r}_i,\vec{r}_j\right)-g_{4a}\left(\vec{r}_i,\vec{r}_j\right)\right\}+g_{4a}\left(\vec{r}_i,\vec{r}_j\right)$$

MPI Internode Communication

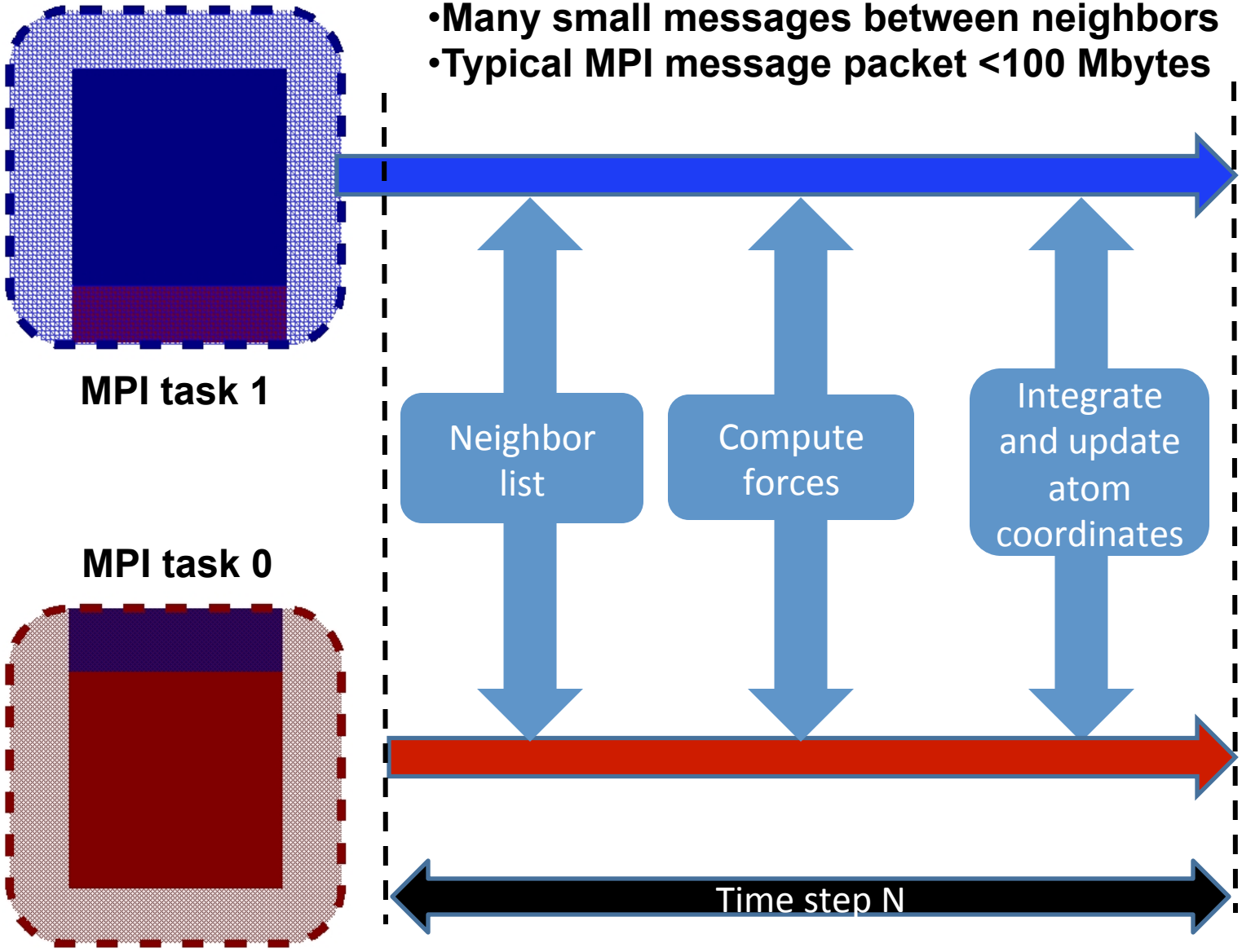
Cutoff distance r for atom



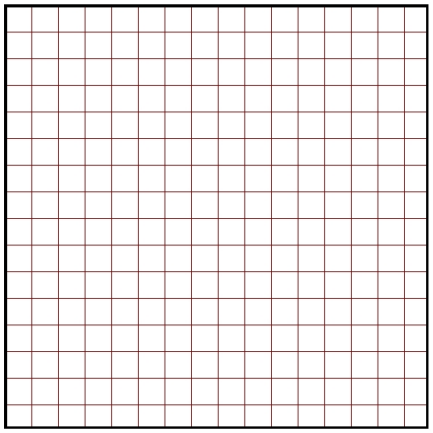


MPI Internode Communication (cont.)

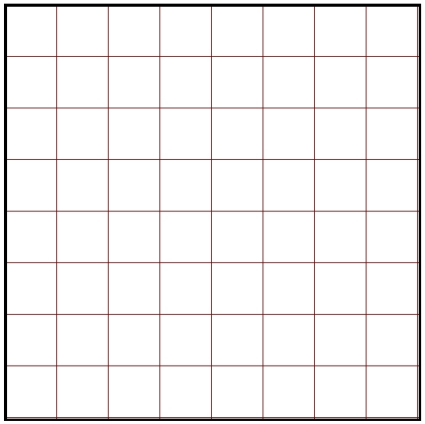
- Many small messages between neighbors
- Typical MPI message packet <100 Mbytes



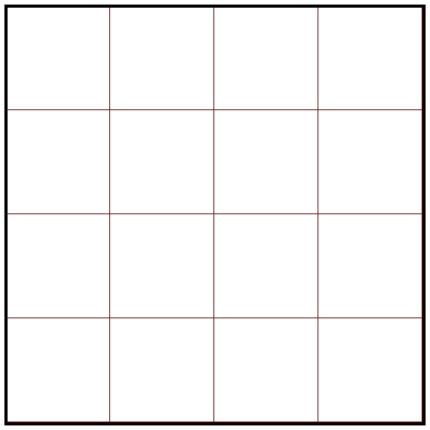
MSM Grid Structure



Level 0
Lattice spacing h
16x16 grid

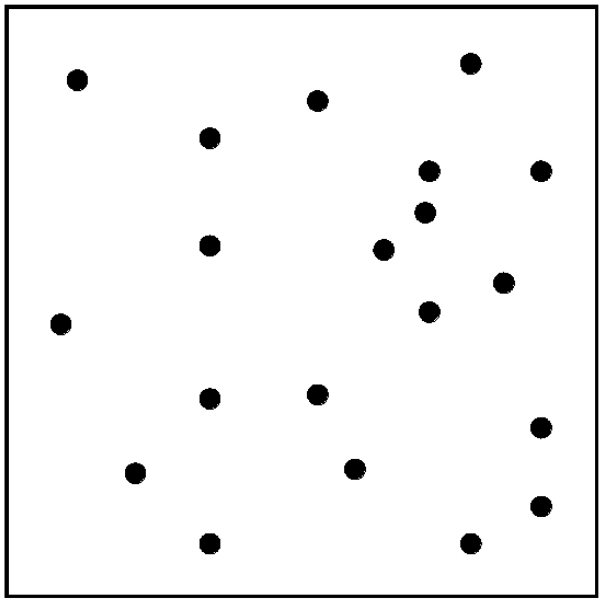


Level 1
Lattice spacing 2h
8x8 grid

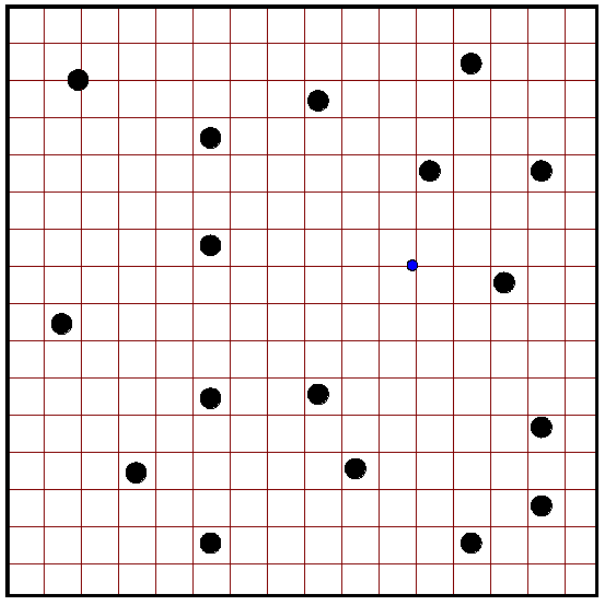


Level 2
Lattice spacing 4h
4x4 grid

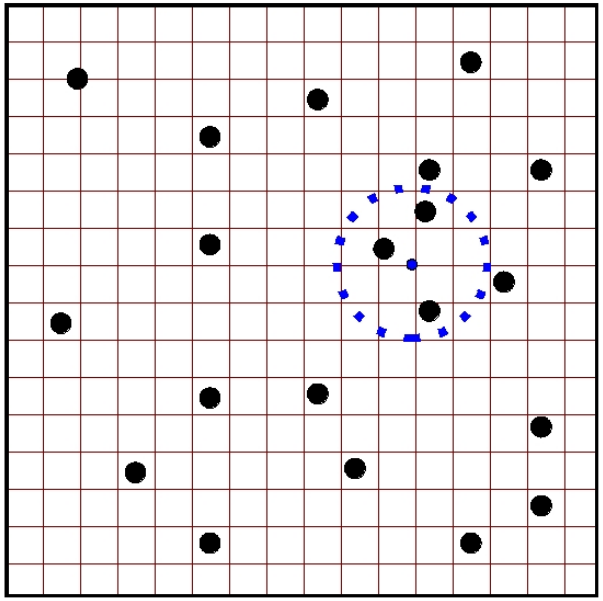
Anterpolation – Convolution of Charges



Particle level

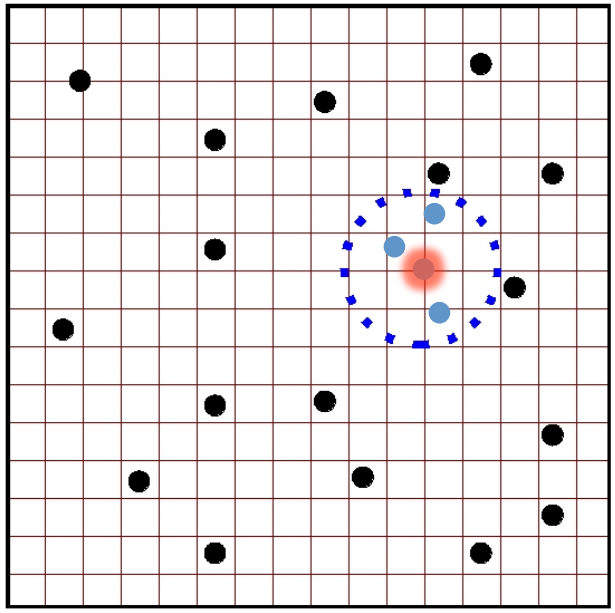


Level 0



Basis functions provide local support about each grid point

Level 0



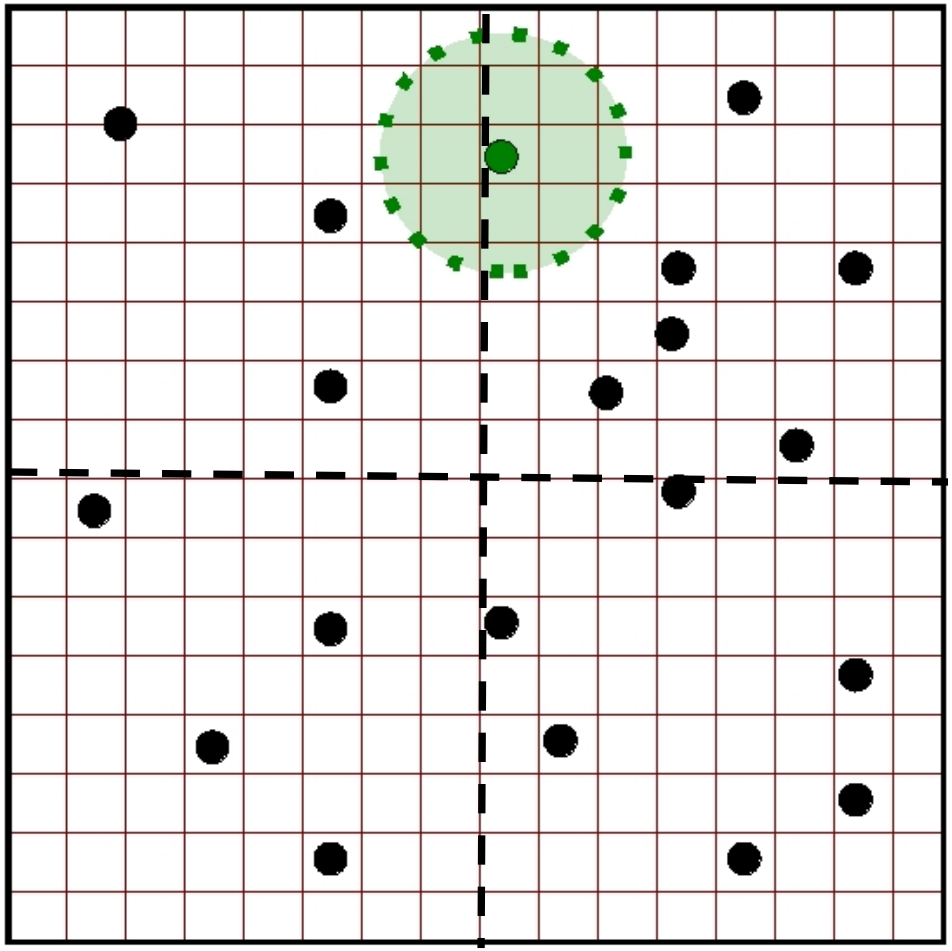
Level 0

Anterpolation (cont.)

Basis functions provide local support about each grid point

MPI task 0

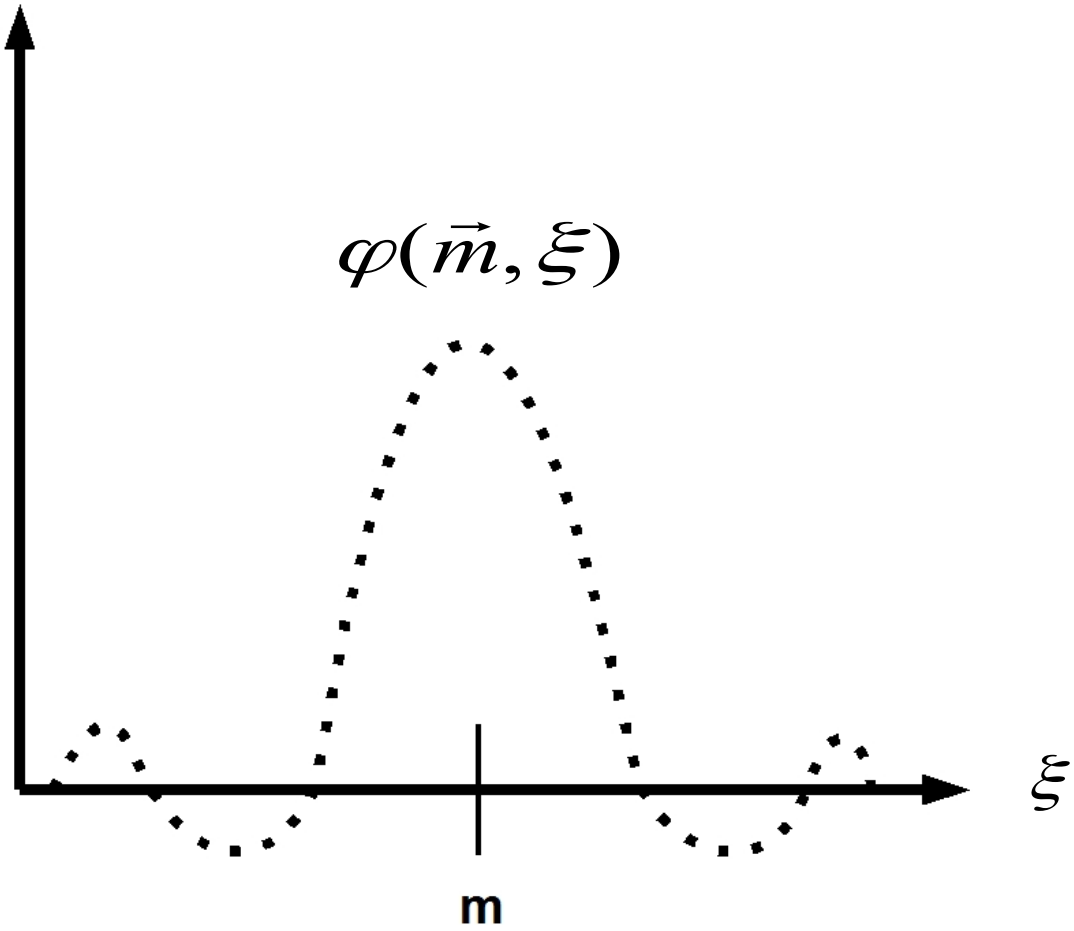
MPI task 1



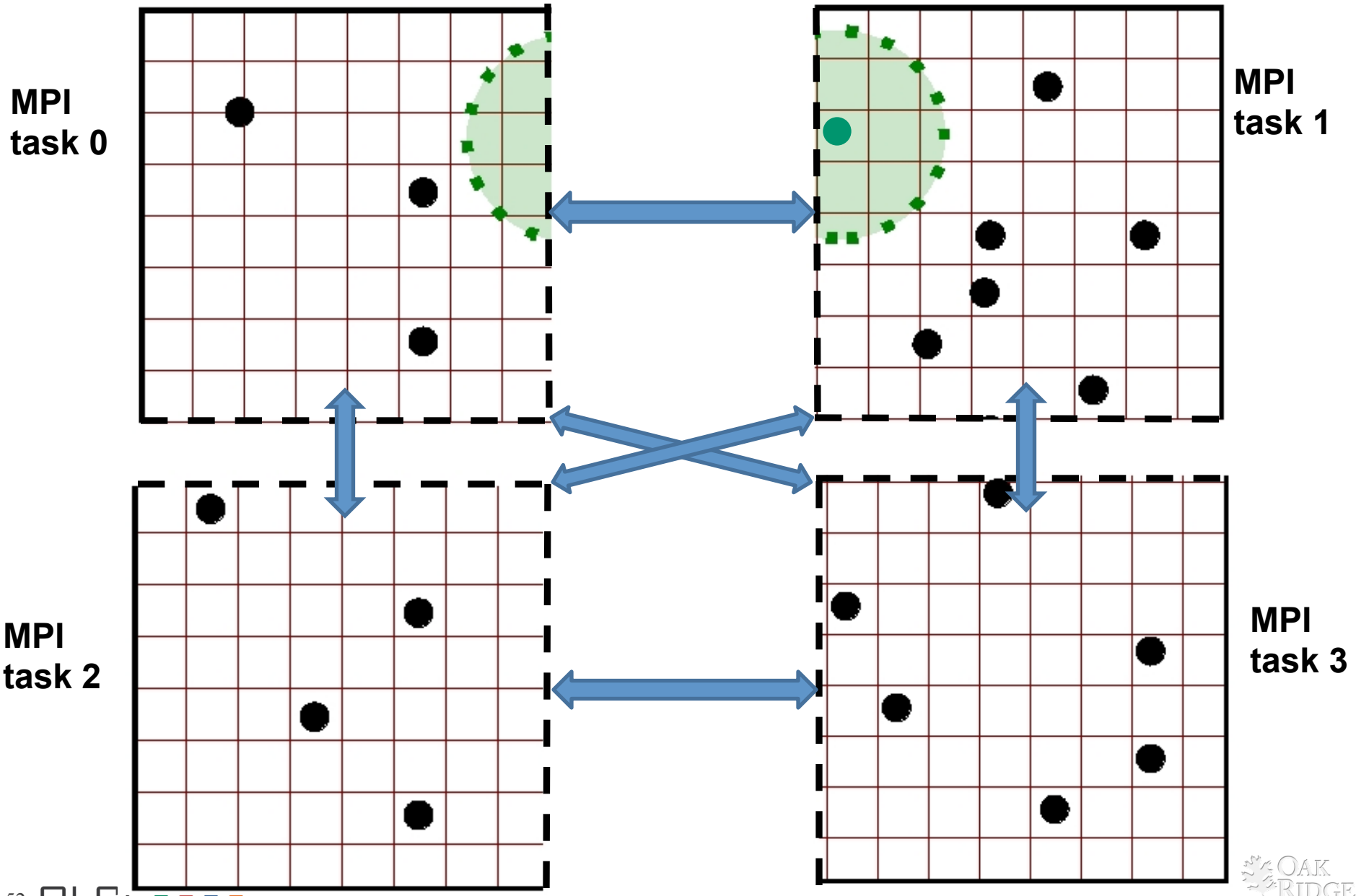
MPI task 2

MPI task 3

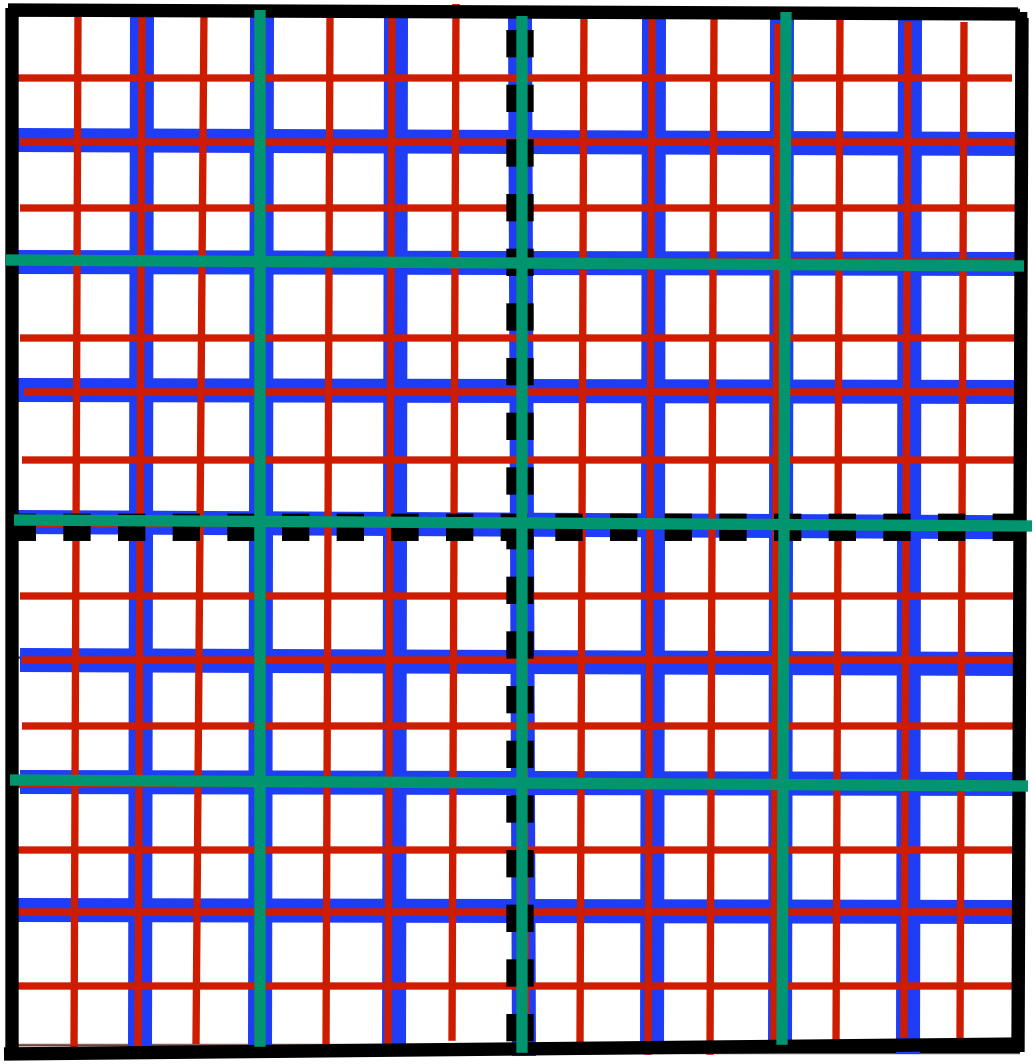
Nodal Basis Functions



Anterpolation Communication-Nearest Neighbor



Restriction of Grid



MPI task 0

MPI task 1

Level 2 grid

Restrict charges from level 1 to level 2

Animation of grid restriction

Level 1 grid

Restrict charges from level 0 to level 1

Level 0 grid

MPI task 2

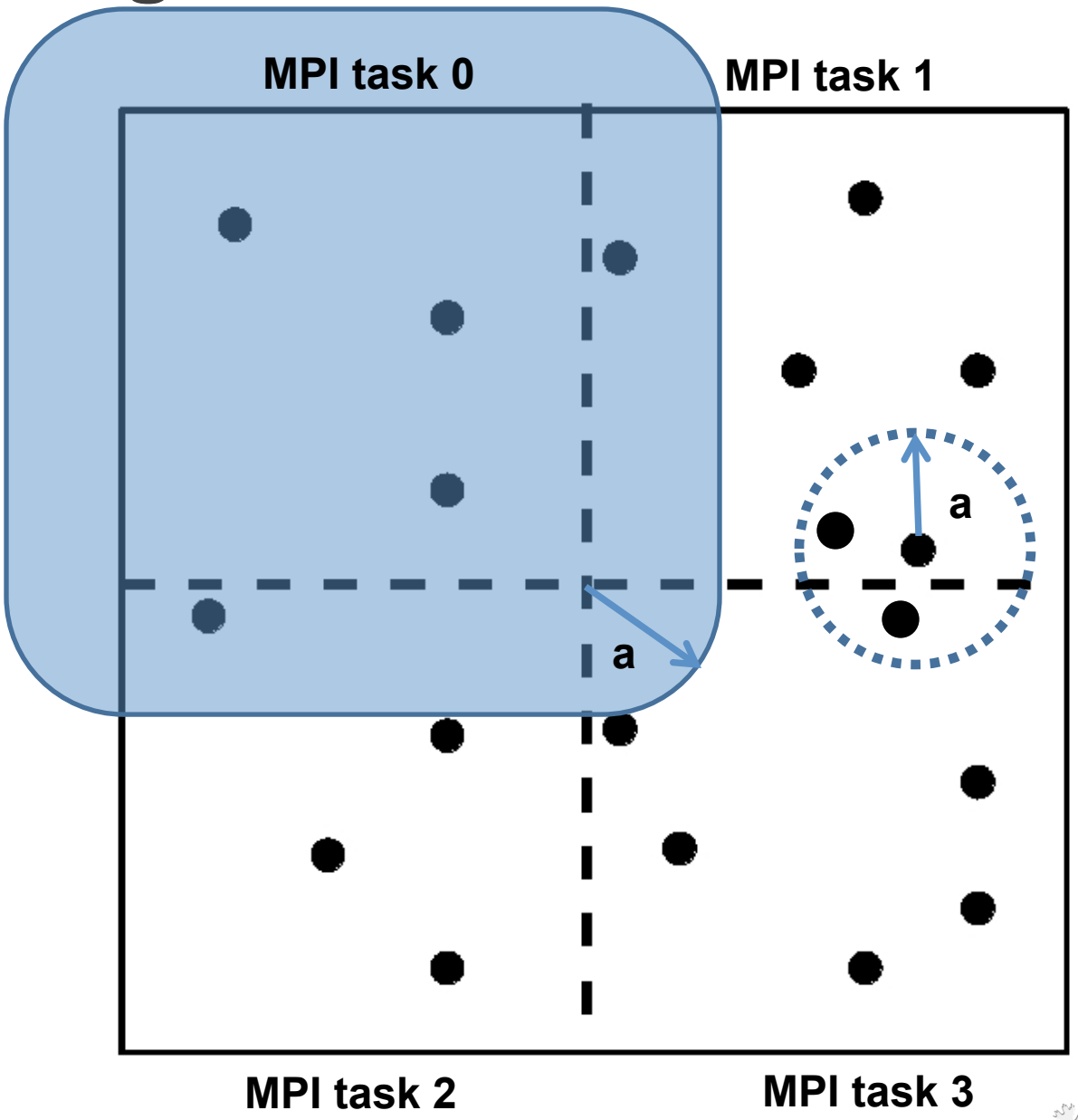
MPI task 3

MSM Short Range

MSM short range cutoff of distance a

$$U^{Short} = q_i q_j \left(\frac{1}{r} - \frac{1}{a} \gamma\left(\frac{r}{a}\right) \right) \frac{\vec{r}_{ij}}{r}$$

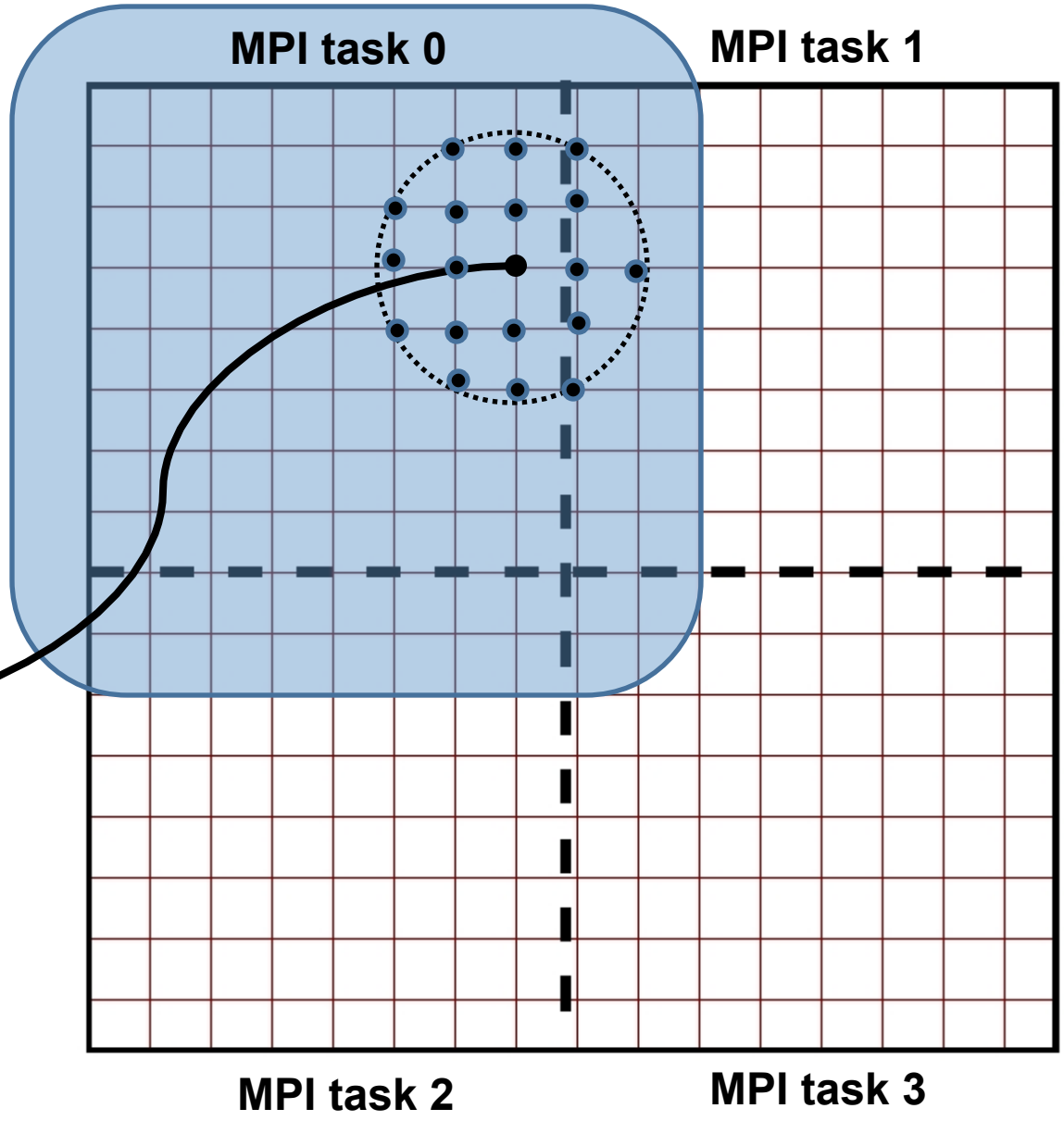
$$f_j^{Short} = q_i q_j \left(\frac{1}{r^2} - \frac{1}{a^2} \gamma'\left(\frac{r}{a}\right) \right) \frac{\vec{r}_{ij}}{r}$$



MSM Direct

Animation of direct calculation for MPI task 0

(i_c, j_c, k_c)

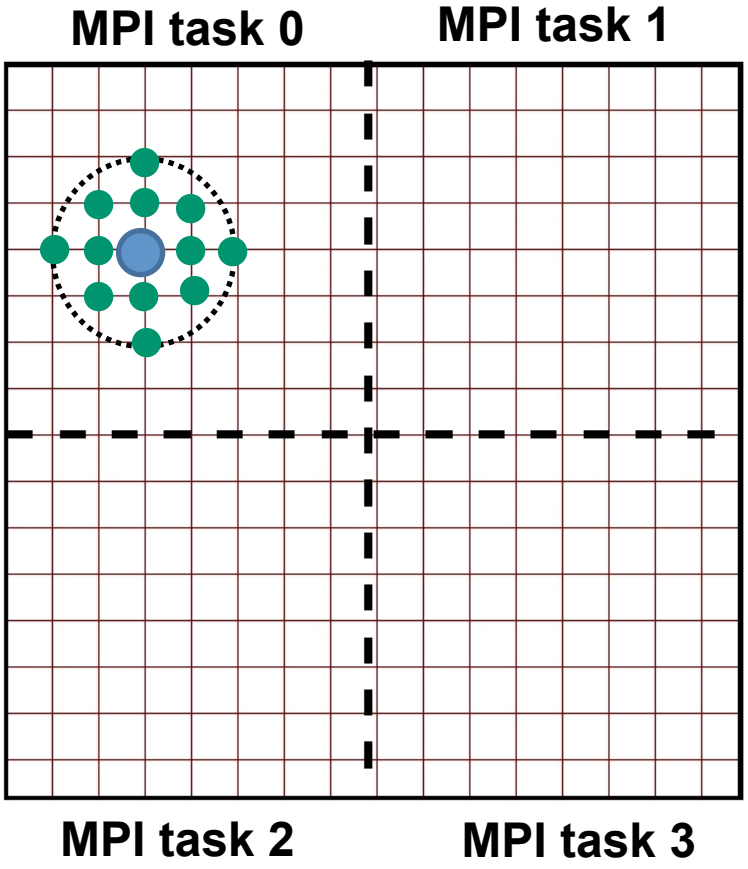


$$e^{n,short}[i_c, j_c, k_c] += \left(g^n(\mathbf{0}, 2^n h(i,j,k)) - g^{n+1}(\mathbf{0}, 2^n h(i,j,k))\right) \times q^n[i_c + i, j_c + j, k_c + k]$$

MSM Prolongation

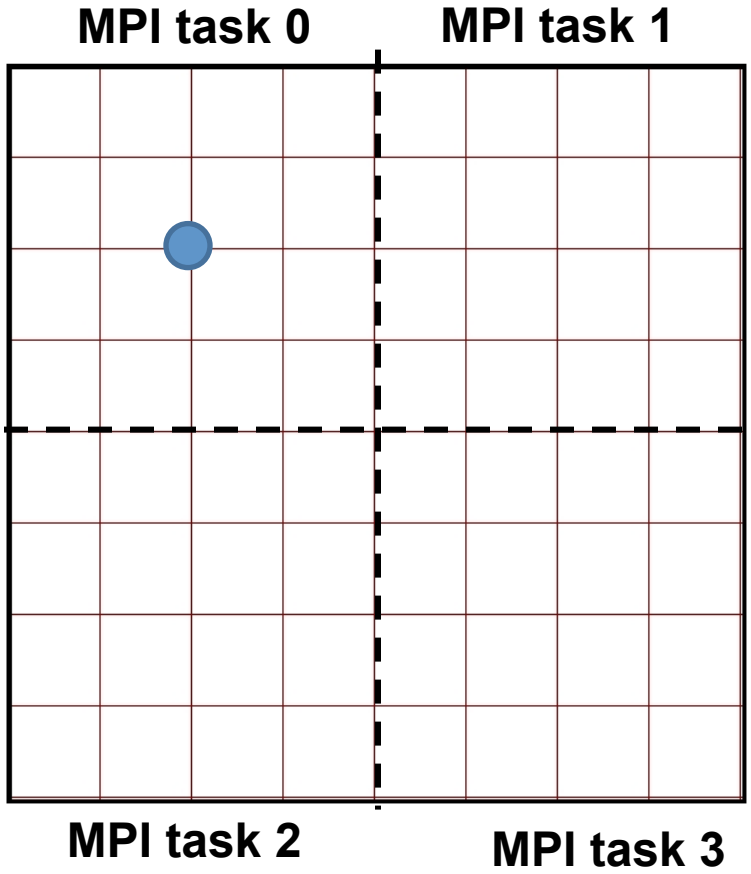
Ω^{n-1}

$\left|e^n\right\rangle = ?$



Ω^n

$\left|e^{n+1}\right\rangle$



$$\left|q^{n+1}\right\rangle = \varphi_{n+1}\left|q^n\right\rangle$$

$$\left|e^n\right\rangle = G^n\left|q^n\right\rangle \cong G^n_{Short}\left|q^n\right\rangle + \left[\varphi_{n+1}\right]^T\left|e^{n+1}\right\rangle$$

Animation of prolongation calculation for MPI task 0

Initial Results of NAMD Lite with MSM

(GPU: NVIDIA GTX-285, using CUDA 3.0;
CPU: 2.4 GHz Intel Core 2 Q6600 quad core)

Box of 21950 flexible waters, 12 A cutoff, 1ps	CPU only	with GPU	Speedup vs. NAMD/CPU
NAMD with PME	1199.8 s	210.5 s	5.7 x
NAMD-Lite with MSM	5183.3 s (4598.6 short, 572.23 long)	176.6 s (93.9 short, 63.1 long)	6.8 x (19% over NAMD/ GPU)

Hardy, D. 2010, "Multiscale Molecular Modeling", Edinburgh, UK, June 30 - July 3, 2010

Data Transfer Between Host and Device

Number of MPI tasks: 18,000
Unit Cell Dimensions: 32,000 Å x 32,000 Å x 33 Å

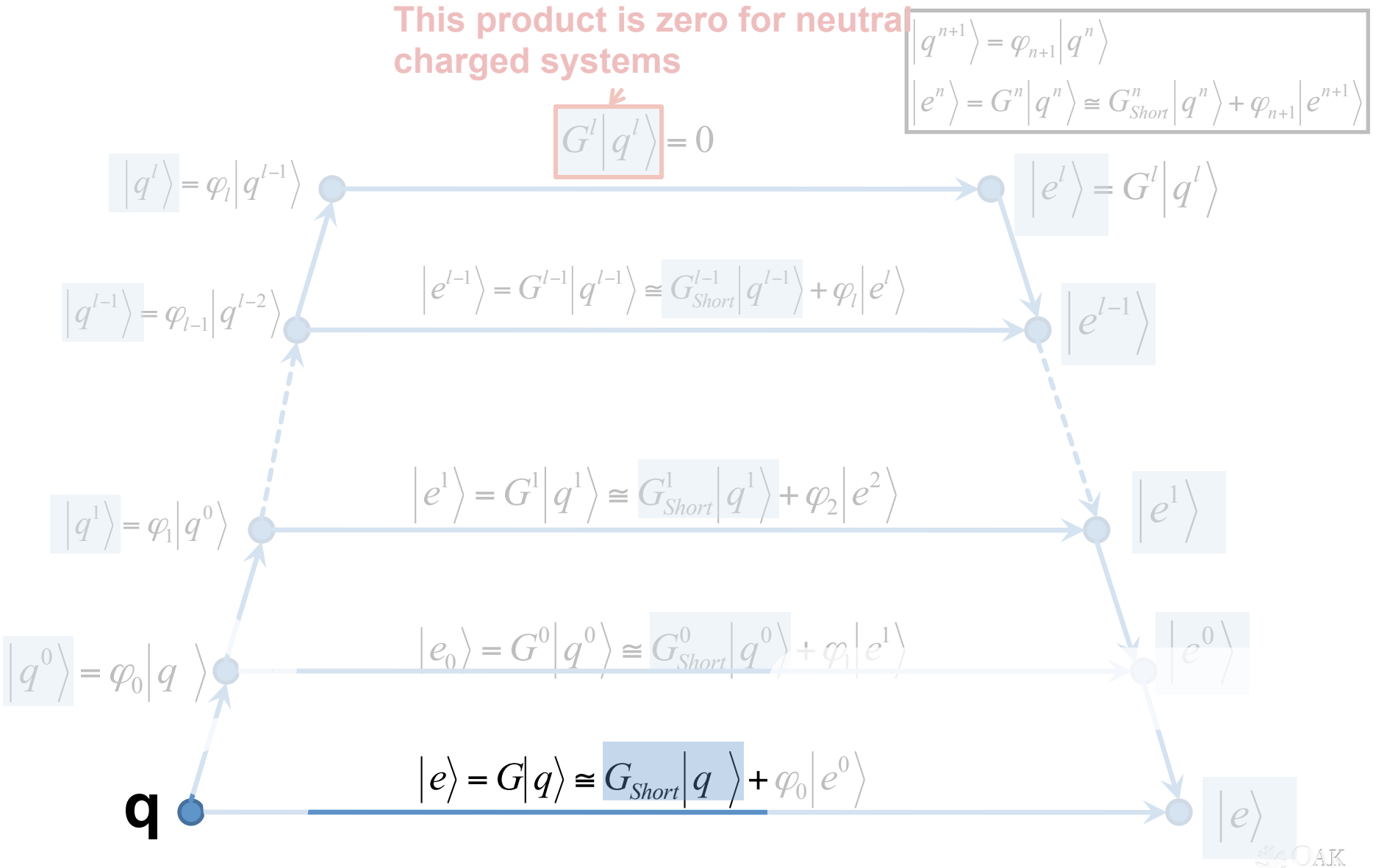
Maximum number of Levels	X grid length	Y grid length	Z grid length	Total Number of grid points	Number of grid points per MPI task
8	3.90625	3.90625	0.12890625	1.68E+07	9.32E+02
9	1.953125	1.953125	0.064453125	1.34E+08	7.46E+03
10	0.9765625	0.9765625	0.032226563	1.07E+09	5.97E+04
11	0.48828125	0.48828125	0.016113281	8.59E+09	4.77E+05
12	0.244140625	0.244140625	0.008056641	6.87E+10	3.82E+06

For the finest grid only

Data Transfer Between Host and Device (cont.)

Maximum number of levels	Bytes per MPI task		Bytes per MPI task		Bytes per MPI task		Bytes per MPI task	
	Level 0		Level 1		Level 2		Level 3	
	Data in	Data out	Data in	Data out	Data in	Data out	Data in	Data out
8	29.8E+3	7.5E+3	3.7E+3	932.1E+0	466.0E+0	116.5E+0	58.3E+0	14.6E+0
9	238.6E+3	59.7E+3	29.8E+3	7.5E+3	3.7E+3	932.1E+0	466.0E+0	116.5E+0
10	1.9E+6	477.2E+3	238.6E+3	59.7E+3	29.8E+3	7.5E+3	3.7E+3	932.1E+0
11	15.3E+6	3.8E+6	1.9E+6	477.2E+3	238.6E+3	59.7E+3	29.8E+3	7.5E+3
12	122.2E+6	30.5E+6	15.3E+6	3.8E+6	1.9E+6	477.2E+3	238.6E+3	59.7E+3

MSM Short Range



MSM Short Range

$$\vec{r}_{ij} = [x_j - x_i, y_j - y_i, z_j - z_i]$$

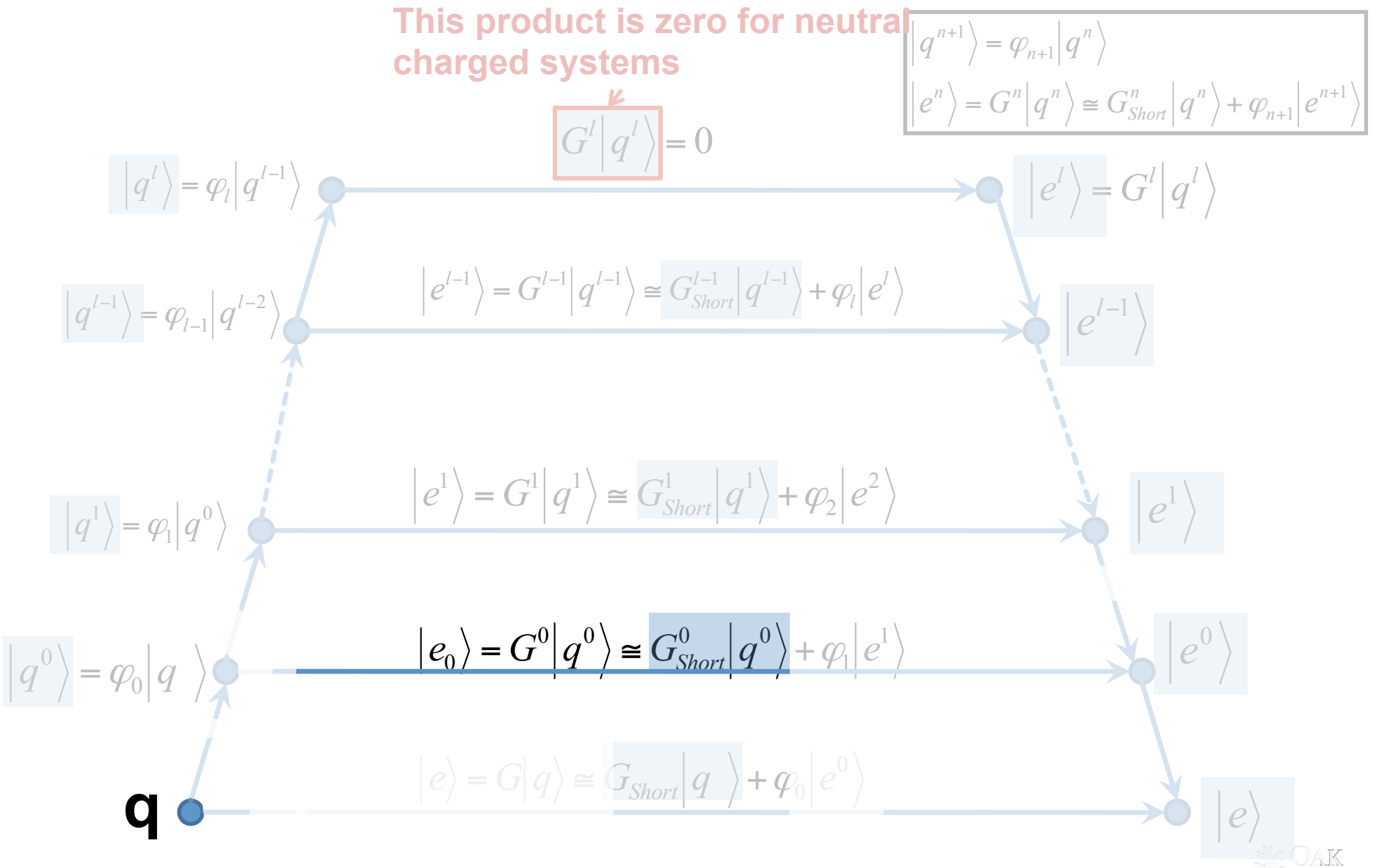
$$r^2 = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$$

if $(r^2 < a^2)$ **then**

$$U^{Short} += q_i q_j \left(\frac{1}{r} - \frac{1}{a} \gamma\left(\frac{r}{a} \right) \right) \frac{\vec{r}_{ij}}{r}$$

$$f_j^{Short} -= q_i q_j \left(\frac{1}{r^2} - \frac{1}{a^2} \gamma'\left(\frac{r}{a} \right) \right) \frac{\vec{r}_{ij}}{r}$$

MSM Direct



MSM Direct

$$\textbf{for } (i_c, j_c, k_c) \in \{\Omega^n \, indices\}$$

$$e^{n,short}[i_c, j_c, k_c] = 0$$

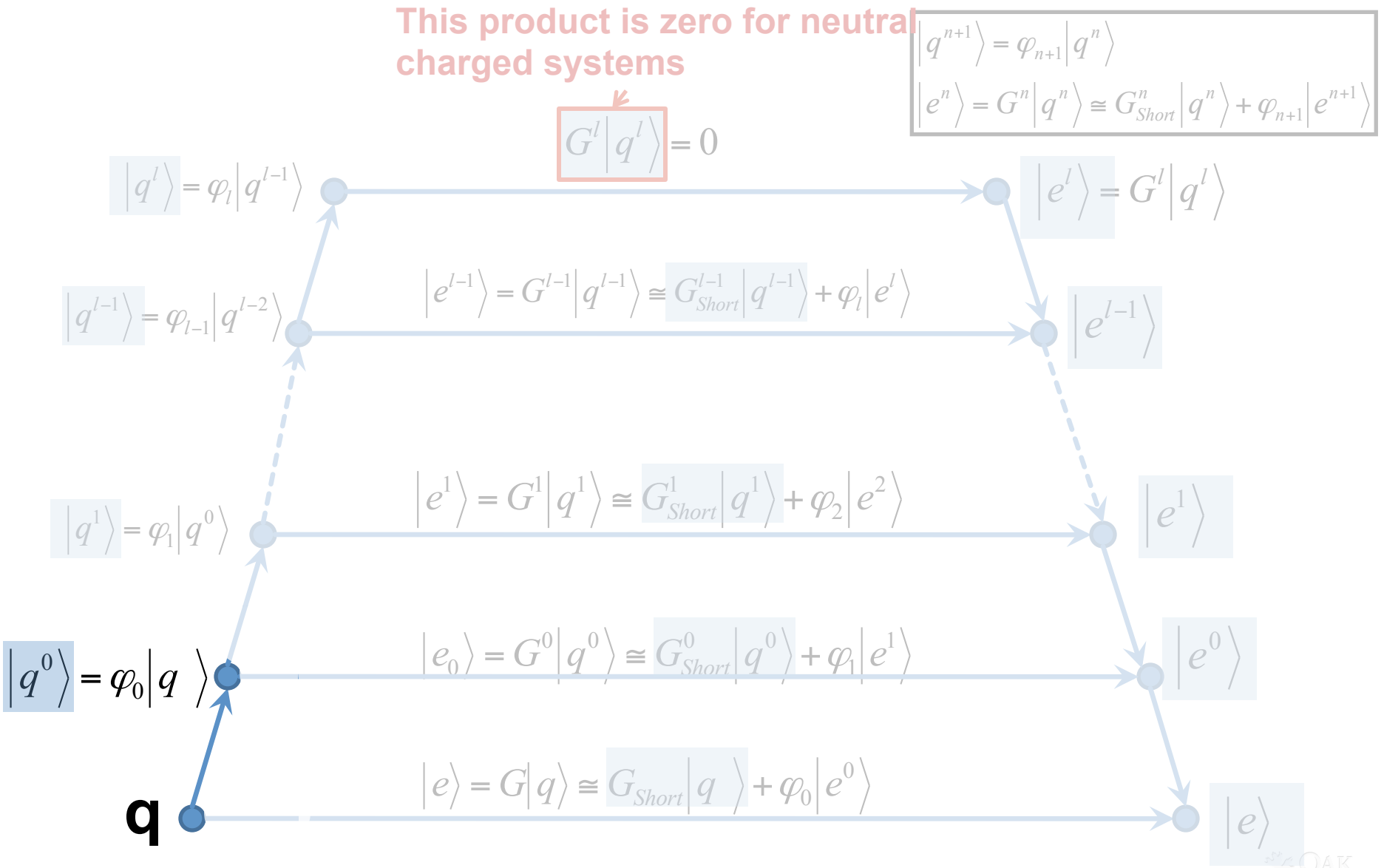
$$\textbf{for } k = \left\lfloor -\frac{2a}{h_z} \right\rfloor, \ldots, \left\lfloor \frac{2a}{h_z} \right\rfloor$$

$$\textbf{for } j = \left\lfloor -\frac{2a}{h_y} \right\rfloor, \ldots, \left\lfloor \frac{2a}{h_y} \right\rfloor$$

$$\textbf{for } i = \left\lfloor -\frac{2a}{h_x} \right\rfloor, \ldots, \left\lfloor \frac{2a}{h_x} \right\rfloor$$

$$e^{n,short}[i_c, j_c, k_c] += \left(g^n\left(\mathbf{0}, 2^n h(i,j,k)\right) - g^{n+1}\left(\mathbf{0}, 2^n h(i,j,k)\right)\right) \times q^n[i_c + i, j_c + j, k_c + k]$$

MSM Anterpolation



Anterpolation

for $n = 1,\ldots,N$

find the lowest grid index of surrounding stencil (x_0, y_0, z_0)

$$i_0 = \left\lfloor \frac{x - x_0}{h_x} \right\rfloor - \frac{p-1}{2}$$

$$j_0 = \left\lfloor \frac{y - y_0}{h_y} \right\rfloor - \frac{p-1}{2}$$

$$k_0 = \left\lfloor \frac{z - z_0}{h_z} \right\rfloor - \frac{p-1}{2}$$

for $\nu = 0,\ldots,p$

$$\varphi_x[\nu] = \Phi\left(\frac{x - x_{i_0+\nu}}{h_x}\right)$$

$$\lambda_x(\nu) = \nu + i_0$$

$$\varphi_y[\nu] = \Phi\left(\frac{y - y_{j_0+\nu}}{h_y}\right)$$

$$\lambda_y(\nu) = \nu + j_0$$

$$\varphi_z[\nu] = \Phi\left(\frac{z - z_{k_0+\nu}}{h_z}\right)$$

$$\lambda_z(\nu) = \nu + k_0$$

for $\mathbf{N}_{\text{z\_stencil}} = 0,\ldots,p$

$$|q_z\rangle = \varphi_z[\mathbf{N}_{\text{z\_stencil}}]$$

$$z_{grid} = \lambda_z(\nu)$$

for $\mathbf{N}_{\text{y\_stencil}} = 0,\ldots,p$

$$|q_{yz}\rangle = |q_z\rangle\varphi_y[\mathbf{N}_{\text{y\_stencil}}]$$

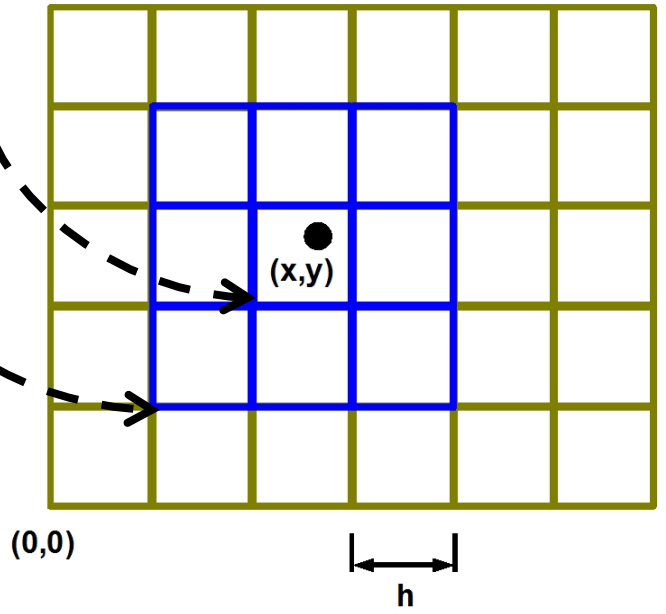
$$y_{grid} = \lambda_y(\nu)$$

for $\mathbf{N}_{\text{x\_stencil}} = 0,\ldots,p$

$$x_{grid} = \lambda_x(\nu)$$

$$q_0[x_{grid}, y_{grid}, z_{grid}] += |q_{yz}\rangle\varphi_x[\mathbf{N}_{\text{x\_stencil}}]q_n$$

$(x_{i_0}, y_{j_0}, z_{k_0})$



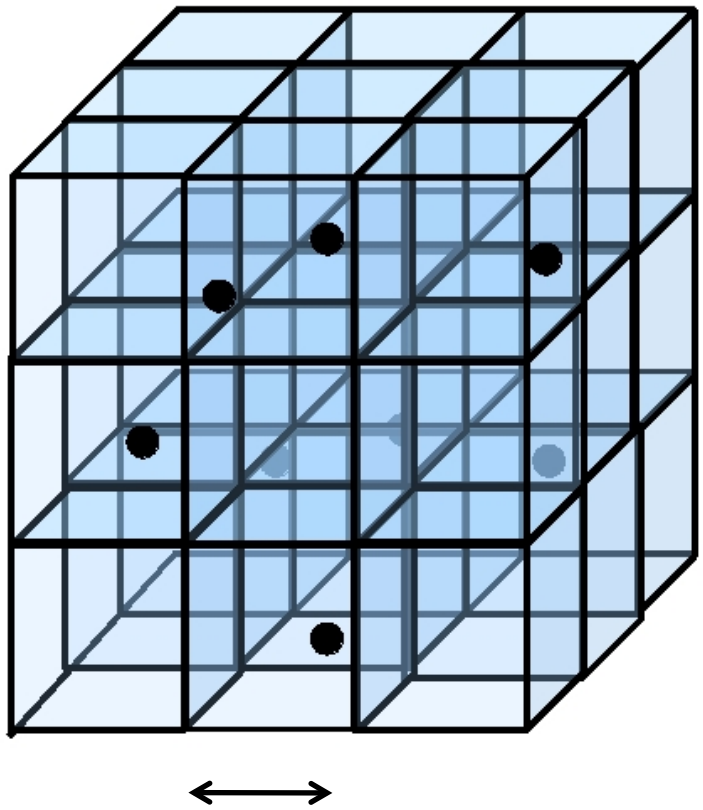
Nodal basis function of degree p=3

$$\Phi^{cubic}(\xi) = \begin{cases} (1-|\xi|)(1+|\xi|-\frac{3}{2}\xi^2), & for\ |\xi| \le 1 \\ -\frac{1}{2}(|\xi|-1)(2-|\xi|)^2, & for\ 1 \le |\xi| \le 2 \\ 0, & otherwise \end{cases}$$

Anterpolation

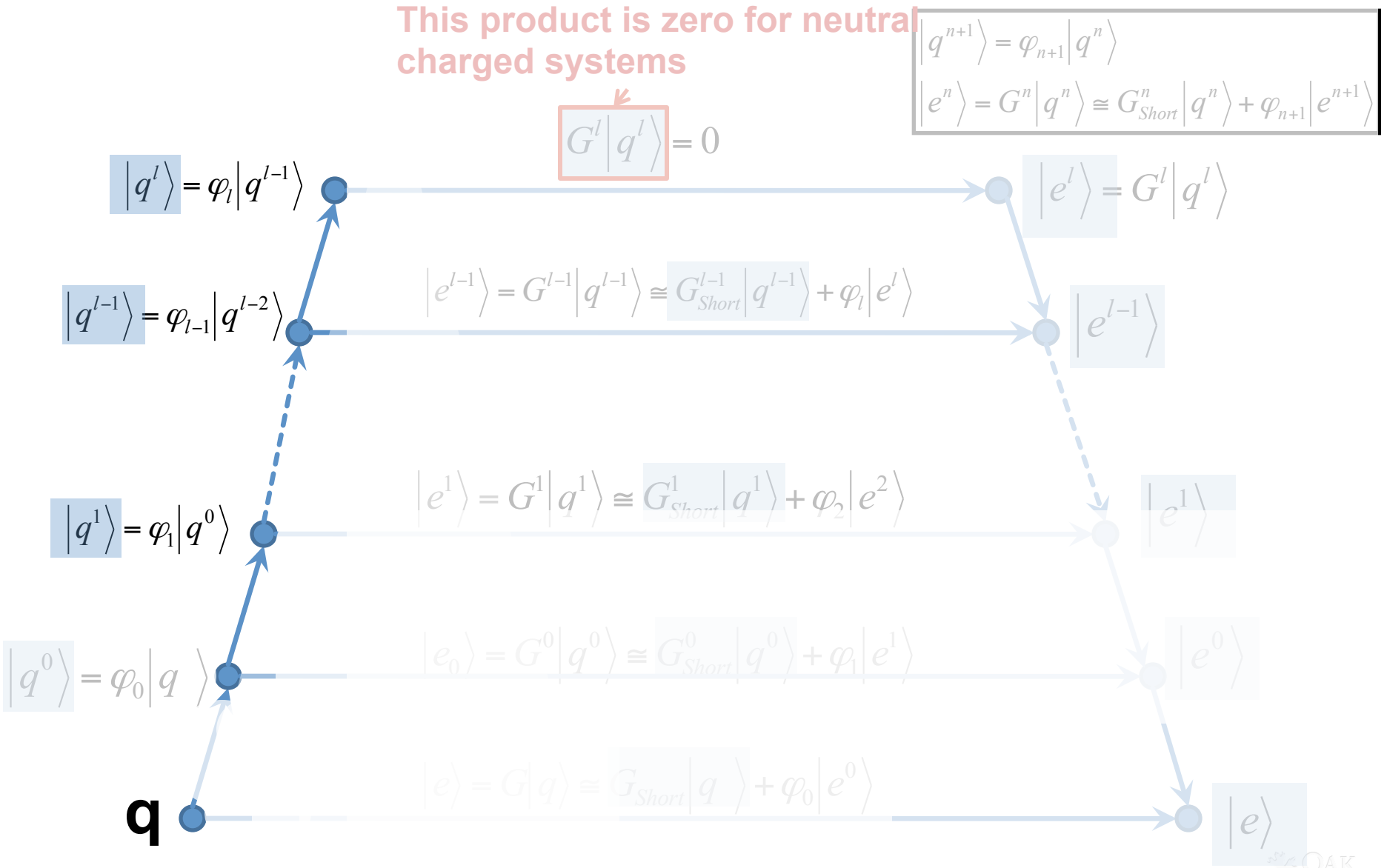
Node *i*

Particle level (k=0)



```
for i=-3,3
    for j=-3,3
        for k=-3,3
            Q(m)=Q(m)+dQ(ijk)
```

Restriction



MSM Restriction

given grid Ω^N with charge q^n

$$i_0 = \left\lfloor \frac{x_0^{n+1} - x_0^n}{2^n h} \right\rfloor, j_0 = \left\lfloor \frac{x_0^{n+1} - x_0^n}{2^n h} \right\rfloor, k_0 = \left\lfloor \frac{x_0^{n+1} - x_0^n}{2^n h} \right\rfloor$$

$$for \quad \nu = -p, \ldots, p+1$$

$$\phi_{(x)}[\nu] = \Phi\left(\frac{x^n_{i_{offset}+\nu} - x_0^{n+1}}{2^{n+1} h_x} \right)$$

$$\phi_{(y)}[\nu] = \Phi\left(\frac{y^n_{i_{offset}+\nu} - y_0^{n+1}}{2^{n+1} h_y} \right)$$

$$\phi_{(z)}[\nu] = \Phi\left(\frac{z^n_{i_{offset}+\nu} - z_0^{n+1}}{2^{n+1} h_z} \right)$$

$$for \quad k' \in \left\{ \left(\Omega^{n+1}\right)_z \textbf{ indices} \right\}$$

$$for \quad j \in \left\{ \left(\Omega^{n}\right)_y \textbf{ indices} \right\}$$

$$for \quad i \in \left\{ \left(\Omega^{n}\right)_x \textbf{ indices} \right\}$$

$$\mathbf{Q}^{n+\frac{1}{3}}[i,j] = 0$$

$$for \quad \nu = -p, \cdots, p+1$$

$$\mathbf{Q}^{n+\frac{1}{3}}[i,j] \mathrel{+}= \varphi_{(z)}[\nu] \cdot q^n[i,j,2k' + k_0 + \nu]$$

$$for \quad j' \in \left\{ \left(\Omega^{n+1}\right)_y \textbf{ indices} \right\}$$

$$for \quad i \in \left\{ \left(\Omega^{n}\right)_x \textbf{ indices} \right\}$$

$$\mathbf{Q}^{n+\frac{2}{3}}[i] = 0$$

$$for \quad \nu = -p, \cdots, p+1$$

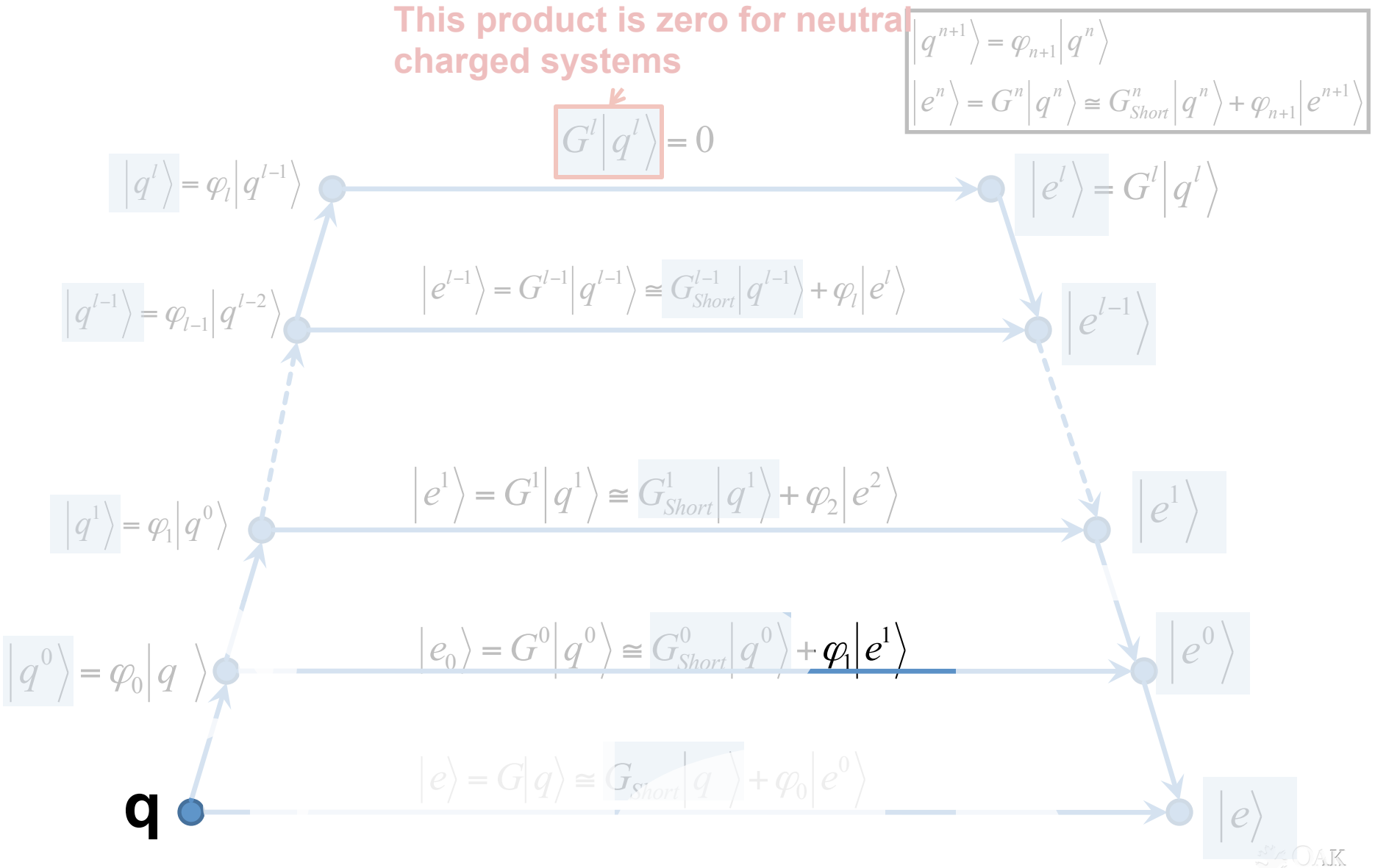
$$\mathbf{Q}^{n+\frac{2}{3}}[i] \mathrel{+}= \varphi_{(y)}[\nu] \cdot \mathbf{Q}^{n+\frac{1}{3}}[i, 2j' + j_0 + \nu]$$

$$for \quad i' \in \left\{ \left(\Omega^{n+1}\right)_x \textbf{ indices} \right\}$$

$$for \quad \nu = -p, \cdots, p+1$$

$$q^{n+1}[i', j', k'] \mathrel{+}= \varphi_{(x)}[\nu] \cdot \mathbf{Q}^{n+\frac{2}{3}}[2i' + i_0 + \nu]$$

MSM Prolongation



Prolongation

given grid Ω^N with charge q^n

$$i_0 = \left\lfloor \frac{x_0^{n+1} - x_0^n}{2^n h} \right\rfloor, j_0 = \left\lfloor \frac{x_0^{n+1} - x_0^n}{2^n h} \right\rfloor, k_0 = \left\lfloor \frac{x_0^{n+1} - x_0^n}{2^n h} \right\rfloor$$

for $\nu = -p, \ldots, p+1$

$$\phi_{(x)}[\nu] = \Phi\left(\frac{x^n_{i_{offset}+\nu} - x_0^{n+1}}{2^{n+1} h_x} \right)$$

$$\phi_{(y)}[\nu] = \Phi\left(\frac{y^n_{i_{offset}+\nu} - y_0^{n+1}}{2^{n+1} h_y} \right)$$

$$\phi_{(z)}[\nu] = \Phi\left(\frac{z^n_{i_{offset}+\nu} - z_0^{n+1}}{2^{n+1} h_z} \right)$$

for $k' \in \left\{ \left(\Omega^{n+1}\right)_z \textbf{ indices} \right\}$

reset $\mathbf{E}^{n+\frac{1}{3}}$ **to 0**

for $j' \in \left\{ \left(\Omega^{n+1}\right)_y \textbf{ indices} \right\}$

reset $\mathbf{E}^{n+\frac{2}{3}}$ **to 0**

for $i' \in \left\{ \left(\Omega^{n+1}\right)_x \textbf{ indices} \right\}$

for $\nu = -p, \cdots, p+1$

$$\mathbf{E}^{n+\frac{2}{3}}[2i' + i_0 + \nu] \mathrel{+}= \varphi_{(x)}[\nu] \cdot e^{n+1}[i', j', k']$$

for $i \in \left\{ \left(\Omega^{n}\right)_x \textbf{ indices} \right\}$

for $\nu = -p, \cdots, p+1$

$$\mathbf{E}^{n+\frac{1}{3}}[i, 2j' + j_0 + \nu] \mathrel{+}= \varphi_{(y)}[\nu] \cdot \mathbf{E}^{n+\frac{2}{3}}[i']$$

for $j \in \left\{ \left(\Omega^{n}\right)_y \textbf{ indices} \right\}$

for $i \in \left\{ \left(\Omega^{n}\right)_x \textbf{ indices} \right\}$

for $\nu = -p, \cdots, p+1$

$$e^{n,long}[i, j, 2k' + k_0 + \nu] \mathrel{+}= \varphi_{(y)}[\nu] \cdot \mathbf{E}^{n+\frac{1}{3}}[i, j]$$