

CMSI配信講義B

第10回

大規模MD並列化の技術1

名古屋大学 工学研究科

CMSI 重点研究員

安藤 嘉倫

- **分子動力学計算**を例に, **三次元トラスネットワーク**上での**高並列対応ソフトウェアのコーディング技術**を紹介
 - 京コンピュータ, FX10 などで有効な技術
 - おそらく他ソフトにも有効な技術. 講義内容にインスパイアされたコードを自作, 公表した場合は文献[1]を引用下さい.
- **MODYLAS^[1]**での実装例をもとに, **並列 (MPI, OpenMP, SIMD)**の効率化技術をメインに, **演算の効率化技術**についても紹介
- **コードの詳細には極力立ち入らずコーディング概念を説明**
 - 詳細は **ライセンスに同意の上** www.modylas.org での公開ソースコードをダウンロードして確認下さい.

[1] Y.Andoh, N.Yoshii, K. Fujimoto, K.Mizutani, H.Kojima, A.Yamada, S.Okazaki, K.Kawaguchi, H.Nagao, K.Iwahashi, F. Mizutani, K.Minami, S. Ichikawa, H.Komatsu, S. Ishizuki, Y.Takeda, and M.Fukushima, *J. Chem. Theory Comput.*, 9, 3201 (2013).



目次

- 分子動力学 (MD) 法
- 分子動力学計算の並列化特性
- 並列化技術 1 データ構造

- 並列化技術 2 MPI
- 並列化技術 3 OpenMP, SIMD

第一回

第二回

分子動力学法(1) 基礎方程式

ニュートンの運動方程式

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i = - \frac{\partial}{\partial \mathbf{r}_i} \Phi(\overbrace{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_i, \dots, \mathbf{r}_N}^{\mathbf{r}^N}) \quad i = 1, \dots, N$$

時間に対する座標の二階常微分方程式

t : 時刻

\mathbf{r}_i : 原子 i の座標

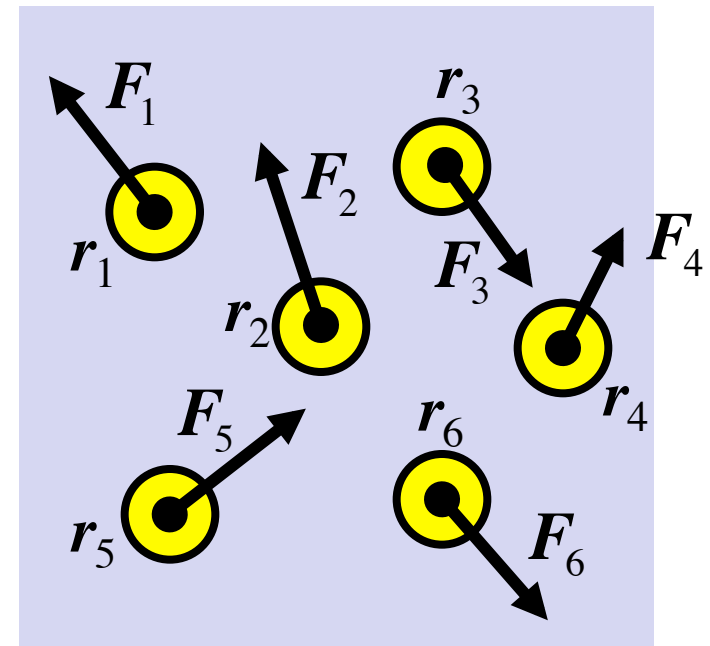
m_i : 原子 i の質量

\mathbf{F}_i : 原子 i に作用する力

$\Phi(\mathbf{r}^N)$: 系のポテンシャルエネルギー

N : 系に含まれる原子数

$N = 6$



分子動力学法(2) 運動方程式の数値積分

ベルレ法による差分化

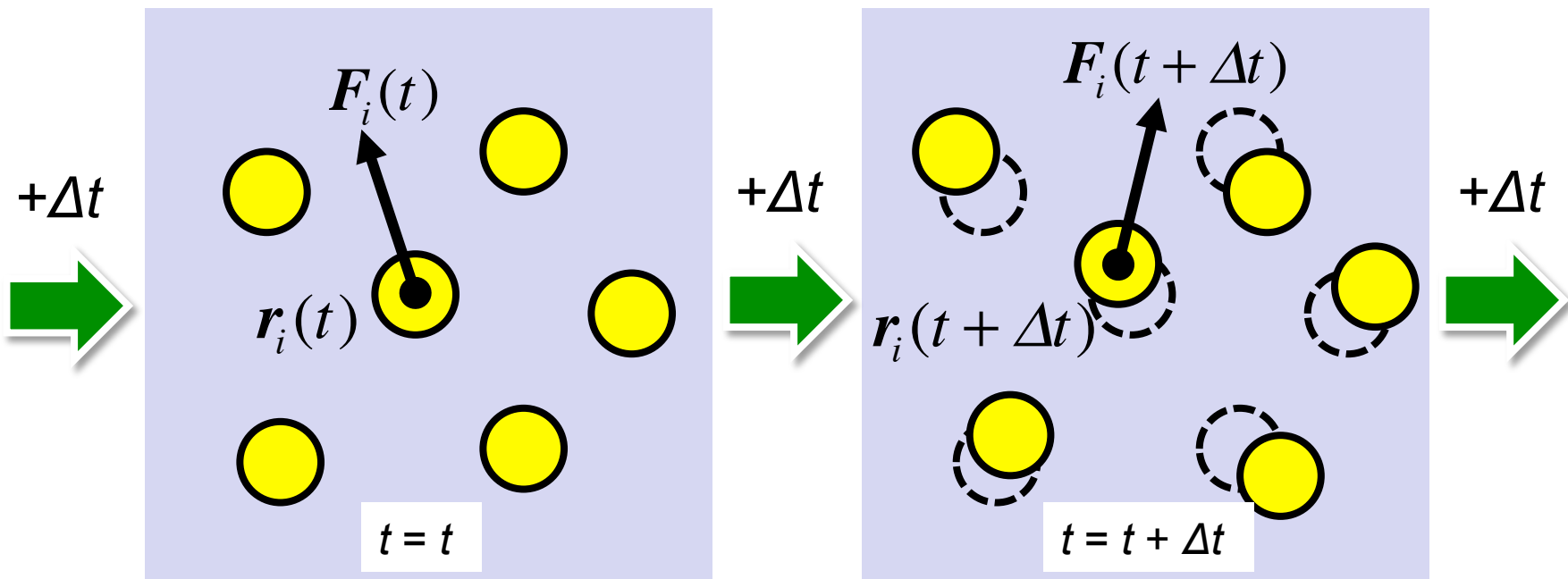
$$r_i(t + \Delta t) = r_i(t) + \frac{dr_i(t)}{dt} \Delta t + \frac{1}{2!} \frac{d^2 r_i(t)}{dt^2} \Delta t^2 + \dots = r_i(t) + v_i(t) \Delta t + \frac{1}{2!} \frac{F_i(t)}{m} \Delta t^2 + \dots \quad \text{Taylor 展開 } +\Delta t$$

$$+ \left. \begin{aligned} r_i(t - \Delta t) &= r_i(t) - \frac{dr_i(t)}{dt} \Delta t + \frac{1}{2!} \frac{d^2 r_i(t)}{dt^2} \Delta t^2 - \dots = r_i(t) - v_i(t) \Delta t + \frac{1}{2!} \frac{F_i(t)}{m} \Delta t^2 - \dots \quad \text{Taylor 展開 } -\Delta t \end{aligned} \right\}$$

$$r_i(t + \Delta t) = 2r_i(t) + r_i(t - \Delta t) + \frac{F_i(t)}{m} \Delta t^2 + O(\Delta t^4) \quad \longrightarrow \quad \text{現在は「速度ベルレ法」}$$

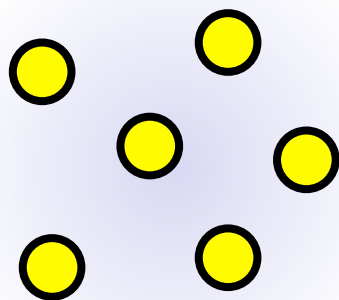
ベルレの差分式

による差分化が標準的



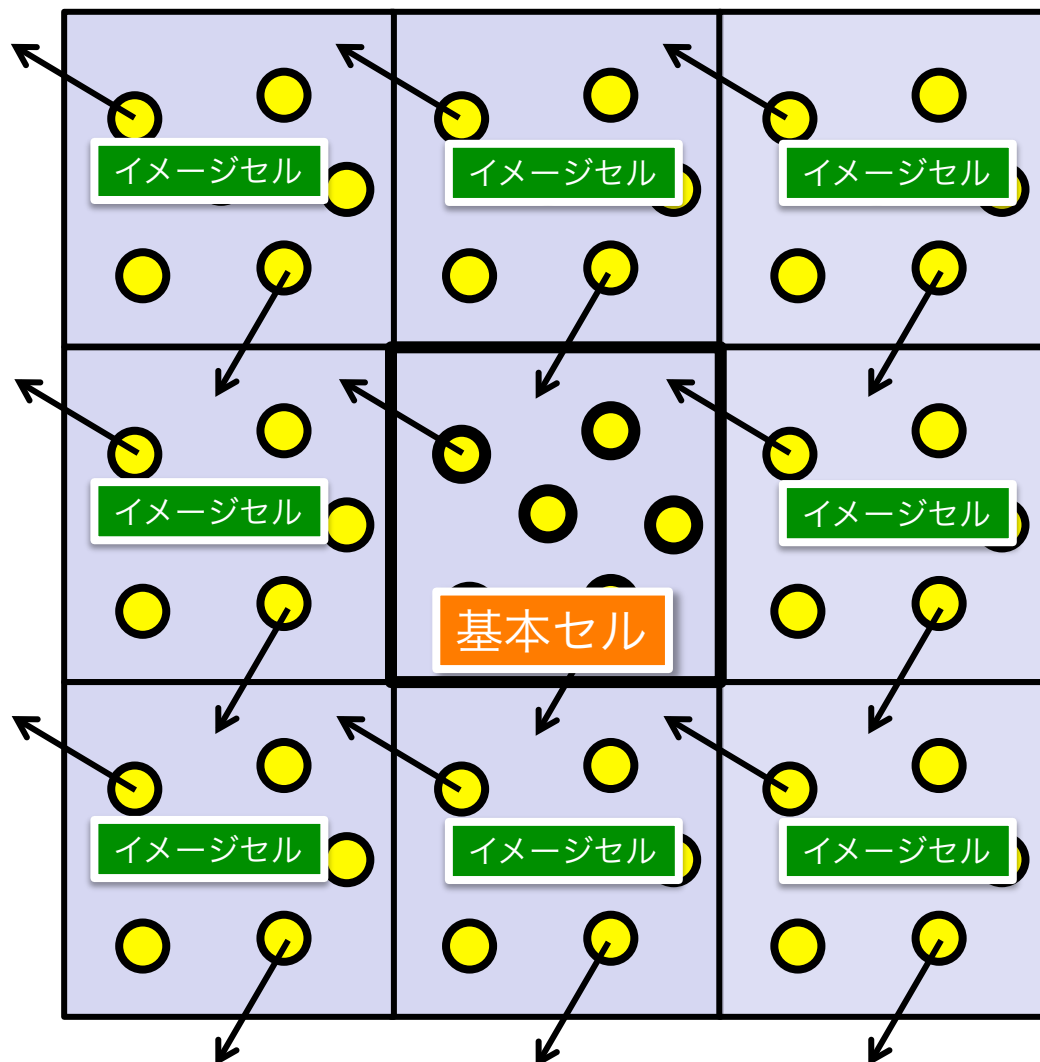
分子動力学法(3) 座標に対する境界条件

自由境界条件



真空中での孤立系 (NVE)

周期境界条件



バルクでの孤立系 (NVE)

分子動力学法(4) 原子間相互作用

古典分子動力学法

古典近似された原子間相互作用関数 (力場) $\Phi(\mathbf{r}^N)$

各項は以降のスライド参照

$$\begin{aligned}
 \Phi(\mathbf{r}^N) = & \sum_{\text{bonds}} K_b (b(\mathbf{r}_i, \mathbf{r}_j) - b_0)^2 + \sum_{\text{angles}} K_\theta (\theta(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) - \theta_0)^2 \\
 & + \sum_{\text{dihedrals}} K_\phi [1 + \cos(n\phi(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) - \delta)] + \sum_{\text{impropers}} K_\psi (\psi(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) - \psi_0)^2 \\
 & + \sum_{\text{nonbonds}} \left[4\varepsilon \left\{ \left(\frac{\sigma}{r(\mathbf{r}_i, \mathbf{r}_j)} \right)^{12} - \left(\frac{\sigma}{r(\mathbf{r}_i, \mathbf{r}_j)} \right)^6 \right\} + \frac{q_i q_j}{r(\mathbf{r}_i, \mathbf{r}_j)} \right] \\
 & \qquad \qquad \qquad \text{Lennard-Jones} \qquad \qquad \qquad \text{Coulomb}
 \end{aligned}$$

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i = - \frac{\partial}{\partial \mathbf{r}_i} \Phi(\mathbf{r}^N)$$

分子動力学法(4) 原子間相互作用

古典分子動力学法

古典近似された原子間相互作用関数 (力場) $\Phi(\mathbf{r}^N)$

各項は以降のスライド参照

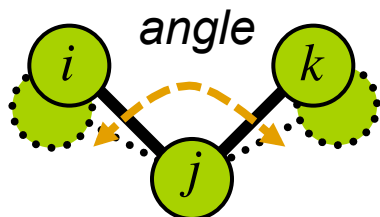
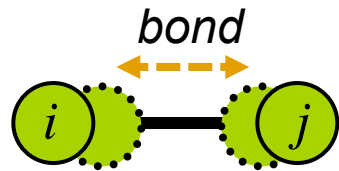
$$\begin{aligned}
 \Phi(\mathbf{r}^N) = & \sum_{\text{bonds}} K_b (b(\mathbf{r}_i, \mathbf{r}_j) - b_0)^2 + \sum_{\text{angles}} K_\theta (\theta(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) - \theta_0)^2 \\
 & + \sum_{\text{dihedrals}} K_\phi [1 + \cos(n\phi(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) - \delta)] + \sum_{\text{impropers}} K_\psi (\psi(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) - \psi_0)^2 \\
 & + \sum_{\text{nonbonds}} \left[4\varepsilon \left\{ \left(\frac{\sigma}{r(\mathbf{r}_i, \mathbf{r}_j)} \right)^{12} - \left(\frac{\sigma}{r(\mathbf{r}_i, \mathbf{r}_j)} \right)^6 \right\} + \frac{q_i q_j}{r(\mathbf{r}_i, \mathbf{r}_j)} \right]
 \end{aligned}$$

Lennard-Jones
Coulomb

分子内 (intra-molecule, bonded) 相互作用 +
 分子間 (inter-molecule, nonbonded) 相互作用

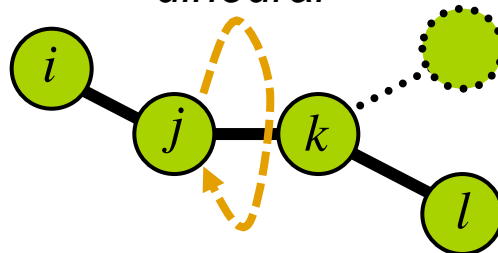
分子動力学法(5) 分子内相互作用

$$K_b (b(\mathbf{r}_i, \mathbf{r}_j) - b_0)^2$$



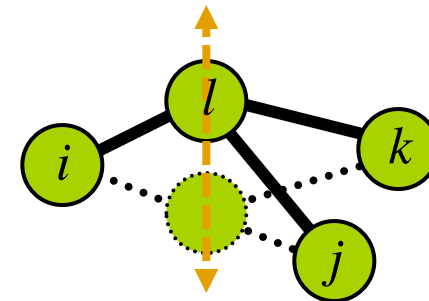
$$K_\theta (\theta(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) - \theta_0)^2$$

dihedral



$$K_\phi [1 + \cos(n\phi(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) - \delta)]$$

improper dihedral



$$K_\psi (\psi(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) - \psi_0)^2$$

コーディングイメージ

do n=1, nbonds

b=b(ri, rj)

$\phi_{bond,ij}$ の計算

$\phi_{bond} = \phi_{bond} + \phi_{bond,ij}$

Fi, Fj の計算

f(i)=f(i)+Fi

f(j)=f(j)+Fj

enddo

do n=1, nangles

theta=theta(ri, rj, rk)

$\phi_{angle,ijk}$ の計算

$\phi_{angle} = \phi_{angle} + \phi_{angle,ijk}$

Fi, Fj, Fk の計算

f(i)=f(i)+Fi

f(j)=f(j)+Fj

f(k)=f(k)+Fk

enddo

do n=1, ndihedrals

phi=phi(ri, rj, rk, rl)

$\phi_{dihedral,ijkl}$ の計算

$\phi_{dihedral} = \phi_{dihedral} + \phi_{dihedral,ij}$

Fi, Fj, Fk, Fl の計算

f(i)=f(i)+Fi

f(j)=f(j)+Fj

f(k)=f(k)+Fk

f(l)=f(l)+Fl

enddo

do n=1, nimpropers

psi=psi(ri, rj, rk, rl)

$\phi_{improper,ijkl}$ の計算

$\phi_{improper} = \phi_{improper} + \phi_{improper,ijkl}$

Fi, Fj, Fk, Fl の計算

f(i)=f(i)+Fi

f(j)=f(j)+Fj

f(k)=f(k)+Fk

f(l)=f(l)+Fl

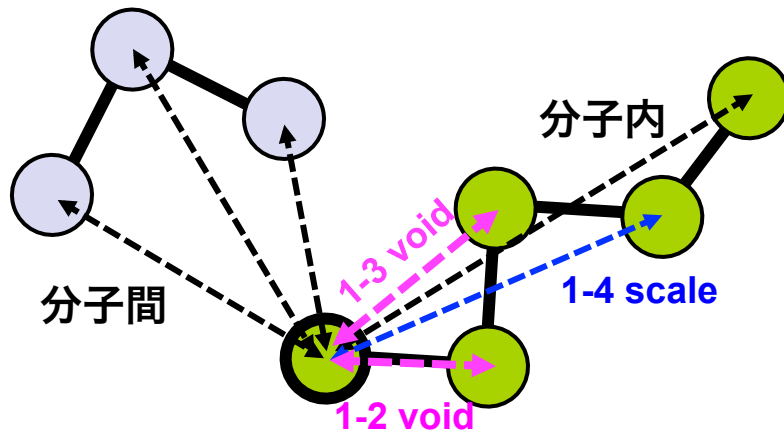
enddo

分子動力学法(6) nonbonded 相互作用

コーディングイメージ

$nonbonded = Lennard-Jones(LJ) + Coulomb(CL)$

$$4\epsilon \left\{ \left(\frac{\sigma}{r(r_i, r_j)} \right)^{12} - \left(\frac{\sigma}{r(r_i, r_j)} \right)^6 \right\} + \frac{q_i q_j}{r(r_i, r_j)}$$



分子内 *nonbond* 項計算の注意点:

- ・1番目および2番目の隣接原子とは相互作用しない (1-2, 1-3 void)
- ・3番目の隣接原子との相互作用は因子 s でスケールする (1-4 scale) [s は LJ, Coulomb べつ]
- ・4番目以降の隣接原子とは通常の相互作用

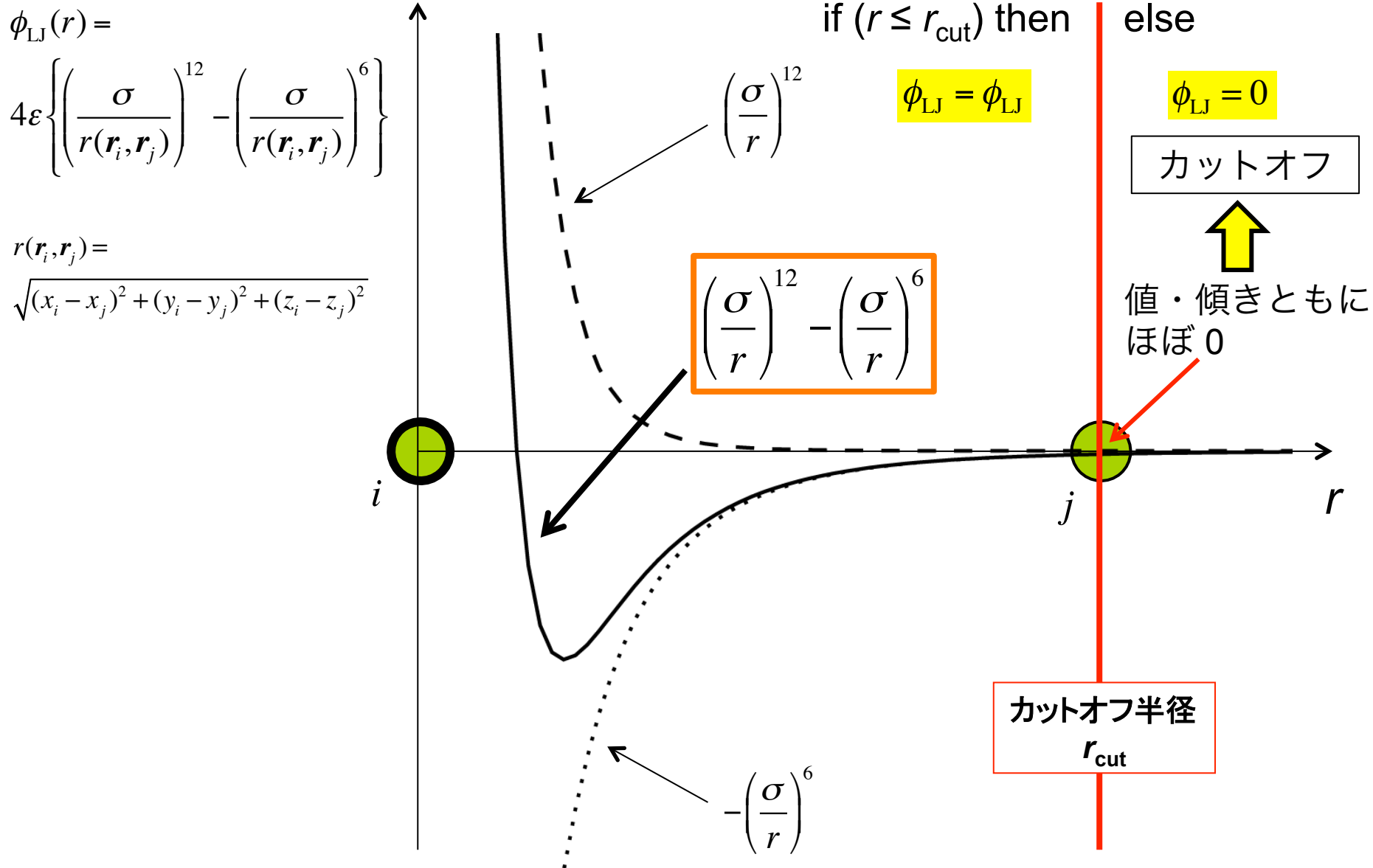
分子間

```
do imol=1,nmol-1
do jmol=imol+1,nmol
do i=1,natom(imol)
do j=1,natom(jmol)
rij=rij(ri, rj)
 $\phi_{ij}$  の計算
 $\phi_{nonbond} = \phi_{nonbond} + \phi_{ij}$ 
 $F_i, F_j$  の計算
 $f(i) = f(i) + F_i$ 
 $f(j) = f(j) + F_j$ 
enddo
enddo
enddo
enddo
```

分子内

```
do imol=1,nmol
do i=1,natom(imol)-1
do j=i+1,natom(imol)
rij=rij(ri, rj)
 $x = \begin{cases} 0 & \text{if 1-2, -3 void} \\ s & \text{if 1-4 scale} \\ 1 & \text{else} \end{cases}$ 
 $\phi_{ij}$  の計算
 $\phi_{nonbond} = \phi_{nonbond} + x * \phi_{ij}$ 
 $F_i, F_j$  の計算
 $f(i) = f(i) + x * F_i$ 
 $f(j) = f(j) + x * F_j$ 
enddo
enddo
enddo
```

分子動力学法(7) LJ 相互作用のカットオフ



分子動力学法(7) LJ相互作用のカットオフ

LJ相互作用を計算
しない別の分子内
原子

コーディングイメージ
(分子間)

```
do imol=1,nmol-1
do jmol=imol+1,nmol
do i=1,natom(imol)
do j=1,natom(jmol)
```

```
rij=rij(ri, rj)
```

```
if(rij > rcut) cycle
```

ϕ_{ij} の計算

```
 $\phi_{nonbond} = \phi_{nonbond} + \phi_{ij}$ 
```

F_i, F_j の計算

```
f(i)=f(i)+F_i
```

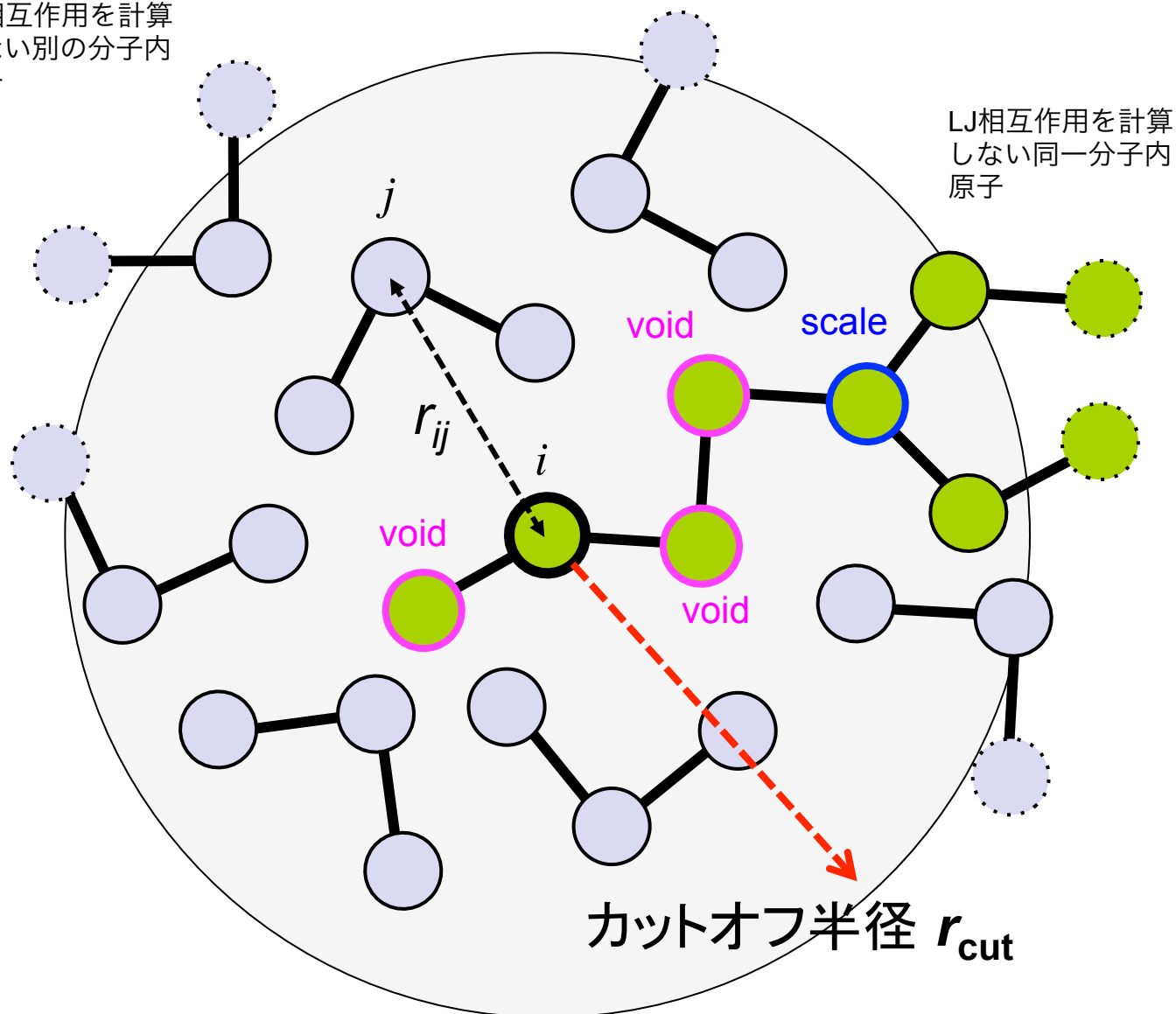
```
f(j)=f(j)+F_j
```

```
enddo
```

```
enddo
```

```
enddo
```

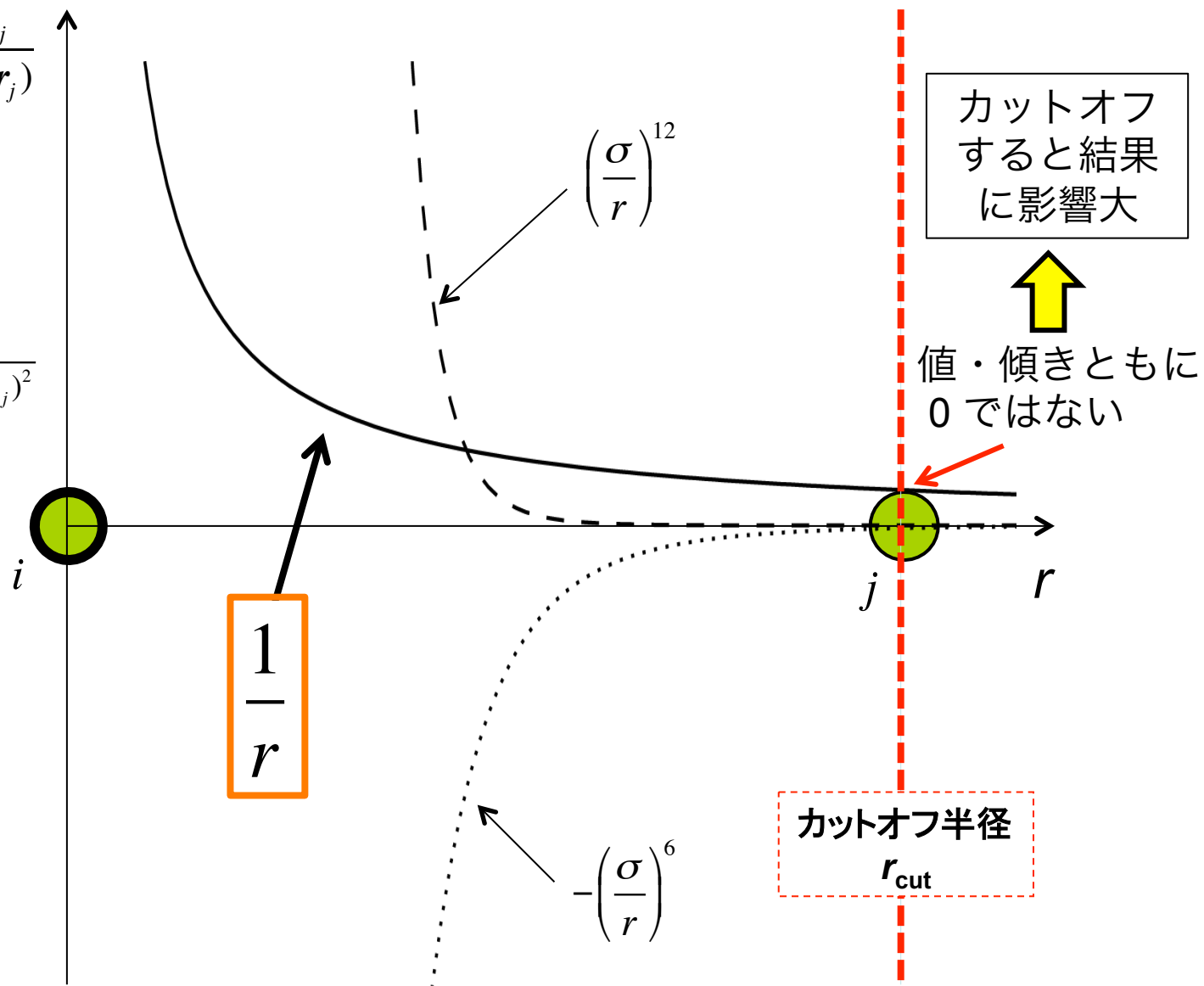
```
enddo
```



分子動力学法(8) Coulomb 相互作用

$$\phi_{\text{CL}}(r) = \frac{q_i q_j}{r(\mathbf{r}_i, \mathbf{r}_j)}$$

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



分子動力学法(8) Coulomb 相互作用

カットオフしない Coulomb 相互作用の計算法

- 自由境界条件下

すべての i, j 原子対を計算

多重極展開法, および高速多重極展開法 (FMM)


- 周期境界条件下

Ewald 法

Particle Mesh Ewald (PME) 法

FMM + 多重極子のEwald法

今回の講義ではこの
並列化を説明



FMM:

方法論の詳細は「CMSI計算科学技術特論A」第12, 13回 古典分子動力学法の高速度化1,2 (吉井) を参照

分子動力学法(8) Coulomb 相互作用

FMM での階層的セル分割

- ・ 基本セルは立方体 (一辺 L)
- ・ 各辺を 2^n 回に等分割

本講義では簡単のためこの2つを前提

また説明の都合上, レベルの数を
「計算科学技術特論A」(吉井) と逆に
記述しているので注意.

用語の定義:

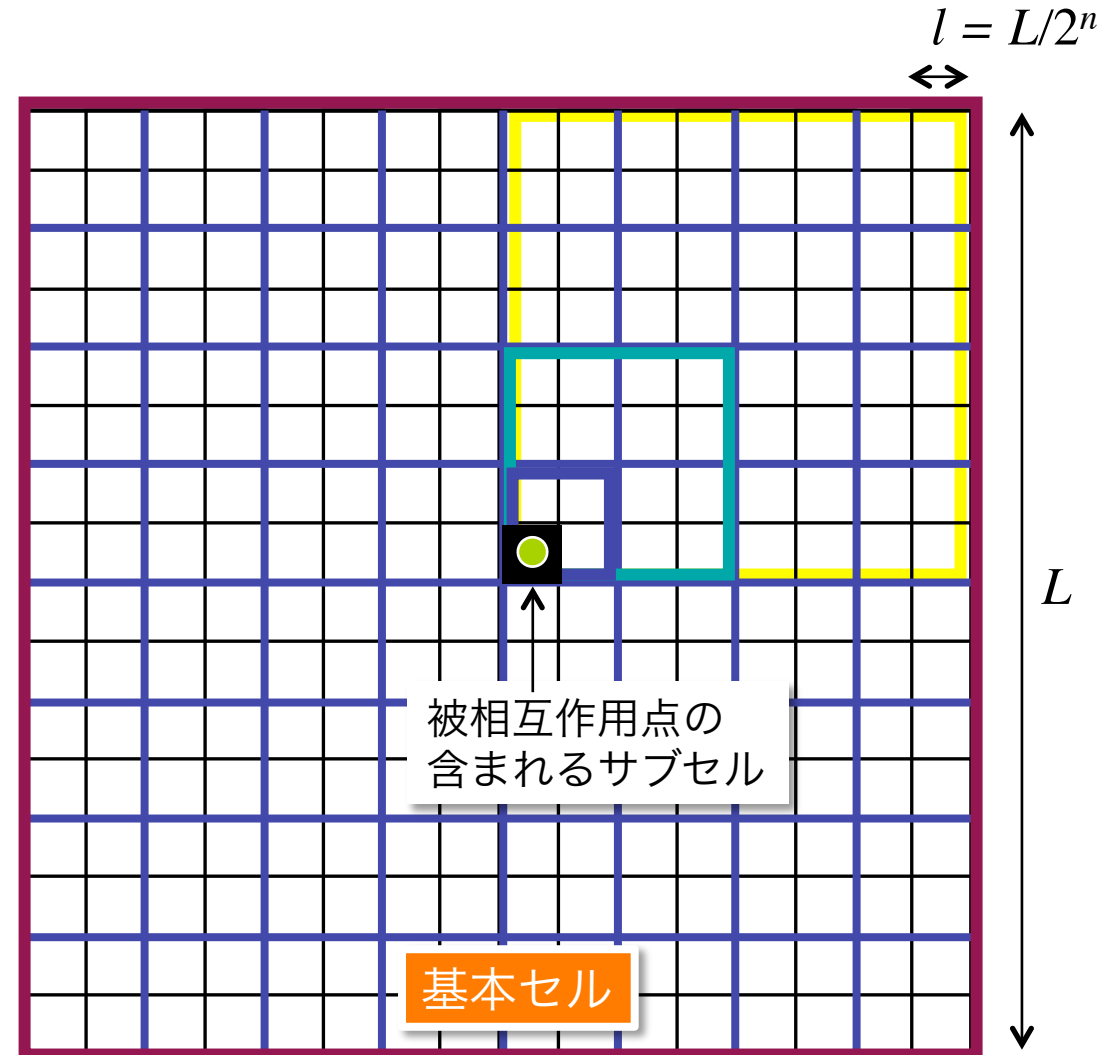
サブセル:

レベル0の分割セル (一辺 $l = L/2^n$)

スーパーセル:

レベル m の分割セル (一辺 $l \cdot 2^m$)

系には計 8^n 個のサブセル, レベル m には
 8^{n-m} 個のスーパーセル.





分子動力学法(8) Coulomb 相互作用

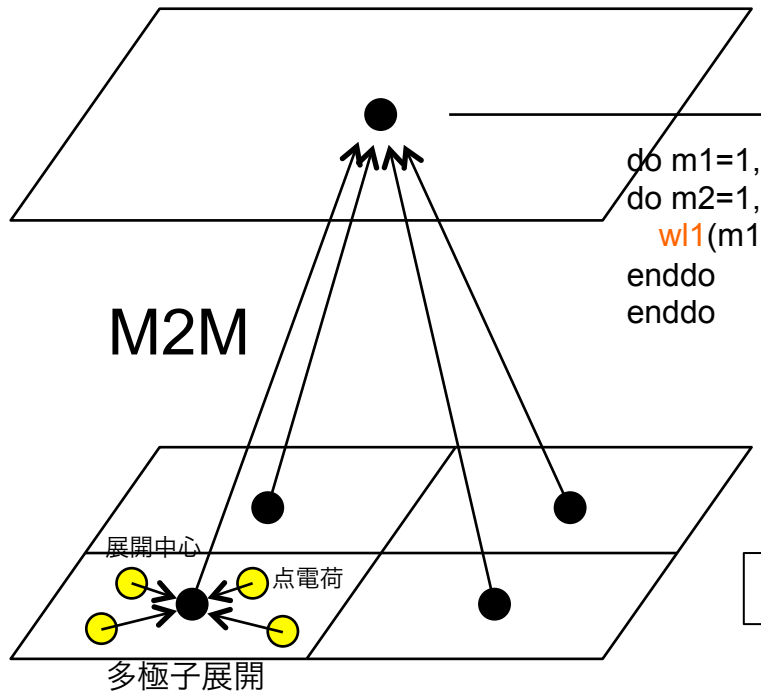
多極子(M) 局所展開係数(L)

FMM の主要 3 演算

complex*8 $wm0, wm1, wl0, wl1$

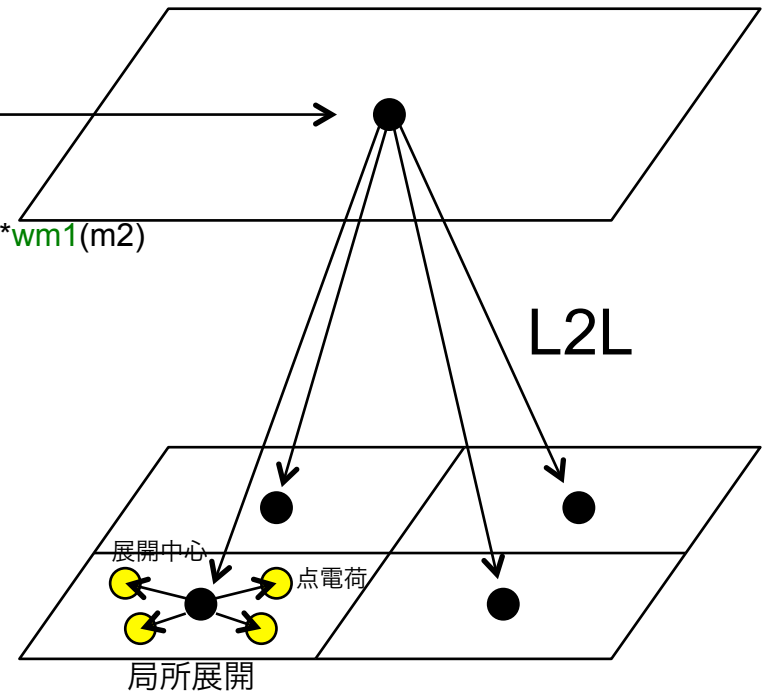
```
allocate(  $wm1((nmax+1)*(nmax+1)), 1, 1$  )
```

```
allocate(  $wl1((nmax+1)*(nmax+1)), 1, 1$  )
```



```
do m1=1,(nmax+1)*(nmax+1)
do m2=1,(nmax+1)*(nmax+1)
 $wl1(m1)=wl1(m1)+m2l(m2,m1)*wm1(m2)$ 
enddo
enddo
```

M2L



$nmax$: 展開次数

```
allocate(  $wm0((nmax+1)*(nmax+1)), 2, 2$  )
```

```
allocate(  $wl0((nmax+1)*(nmax+1)), 2, 2$  )
```

M2M : 多極子展開中心の移動, 足し合わせ

M2L : 局所展開係数の計算

→ **Hot spot**

L2L : 局所展開中心の移動, 足し合わせ

分子動力学法(8) Coulomb 相互作用

相互作用計算の要点

レベル0

- 近傍サブセルに含まれる点電荷と粒子-粒子相互作用を計算
- サブセルAとのM2L計算
- 点電荷への局所展開

Hot spot

レベル1

- スーパーセルBとのM2L, L2L計算

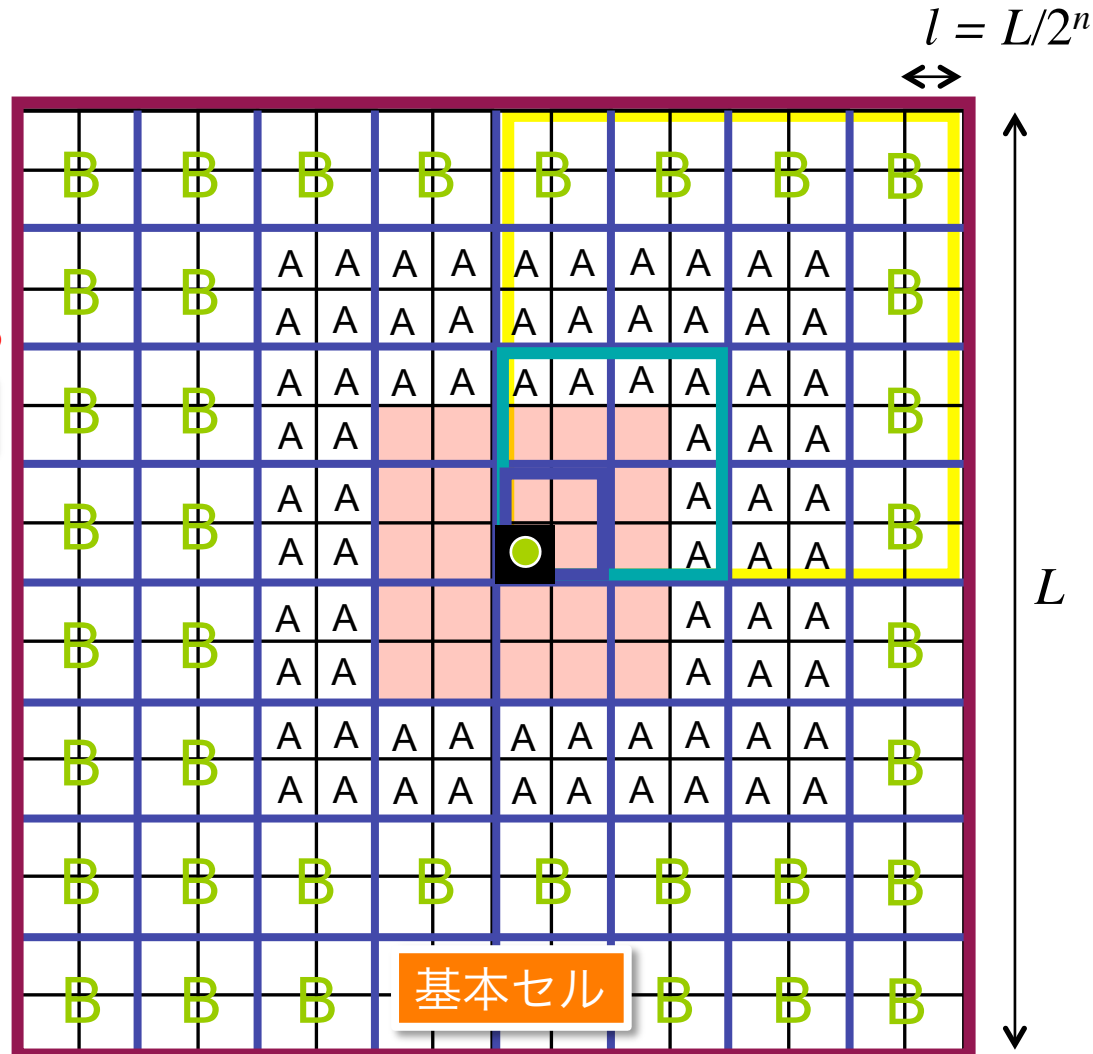
...以下同様 (入れ子構造)

最上レベル

- 多重極子のEwald法, L2L計算

入れ子構造により,

周期境界条件下では各レベル $10^3 - 5^3 = 875$ 個のスーパーセルとM2L計算



□: レベル0

□: レベル1

□: レベル2

□: レベル3

□: レベル4

□: 粒子-粒子相互作用を計算

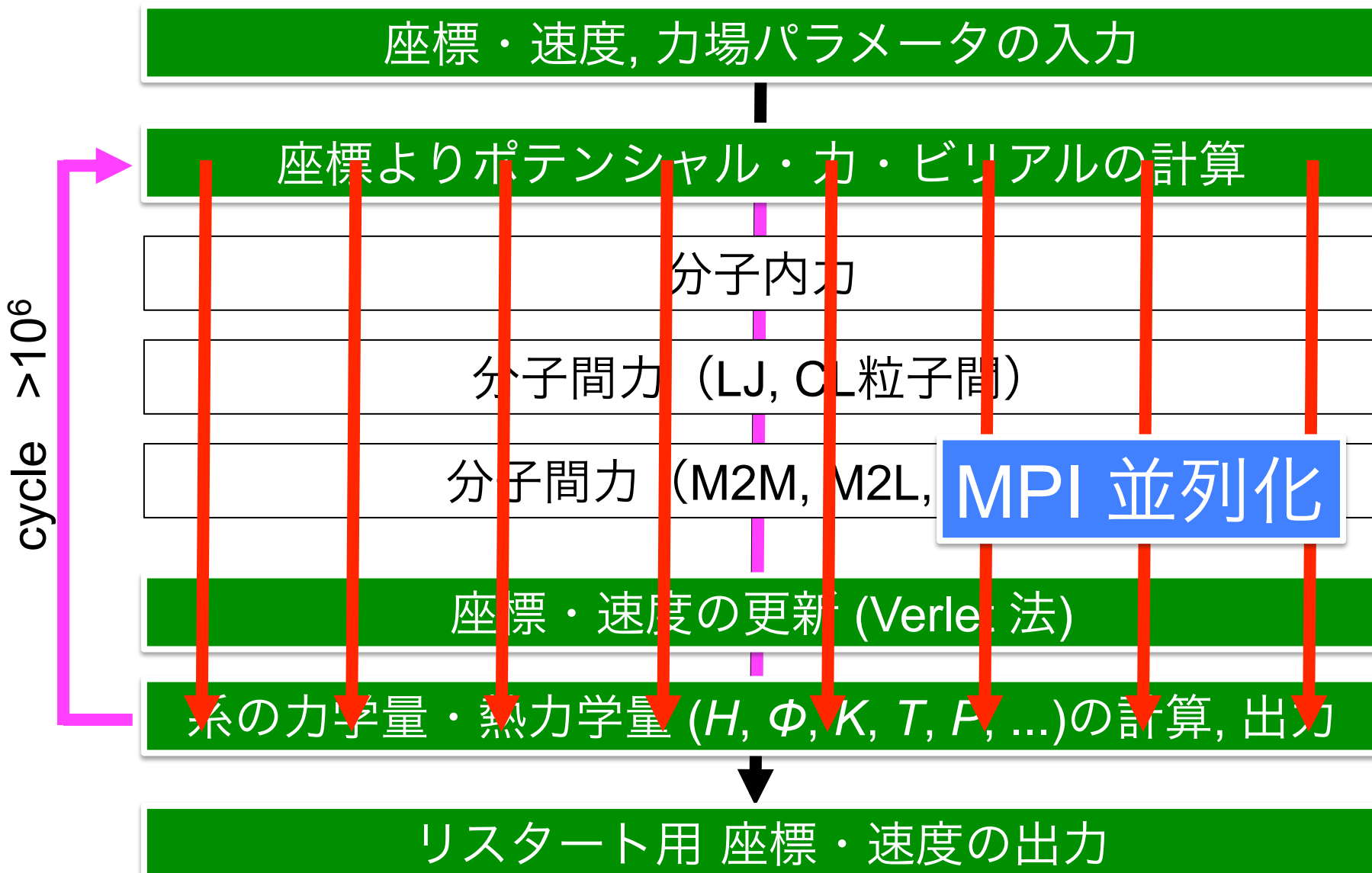
分子動力学計算の並列化特性(1)

MD 計算の処理フロー



分子動力学計算の並列化特性(1)

MD 計算の処理フロー



分子動力学計算の並列化特性(2) プロセス分割

20/170

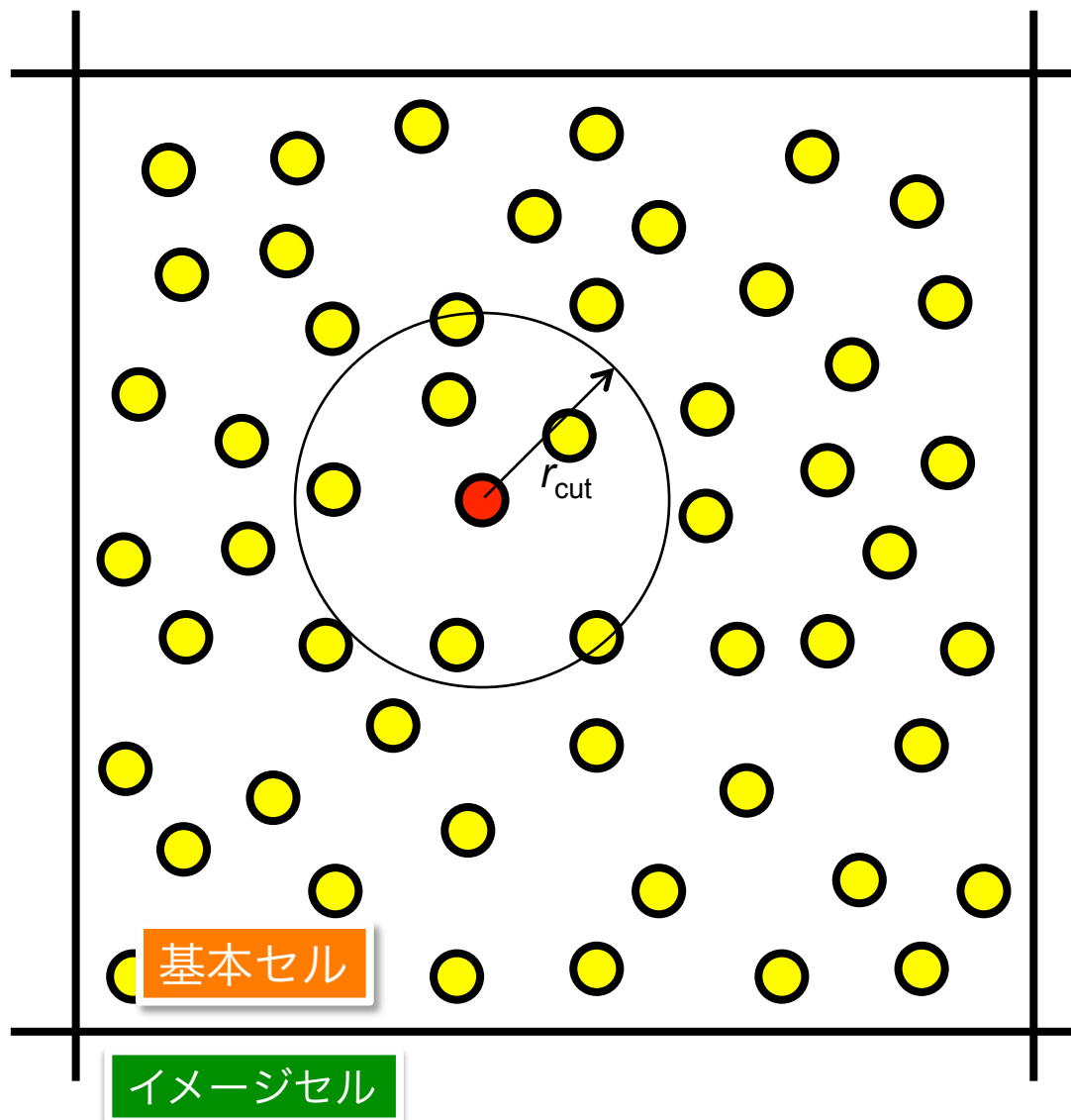
$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i$$

i ごと独立な演算
(粒度 N)

コーディングイメージ

```
do i=1, N
  ポテンシャルの計算
  力の計算
  座標・速度更新
  力学・熱力学量計算
enddo
```

注) あくまで単純化したコーディングイメージ. 実際は i についての do ループは都度閉じています.



分子動力学計算の並列化特性(2) プロセス分割

21 / 70

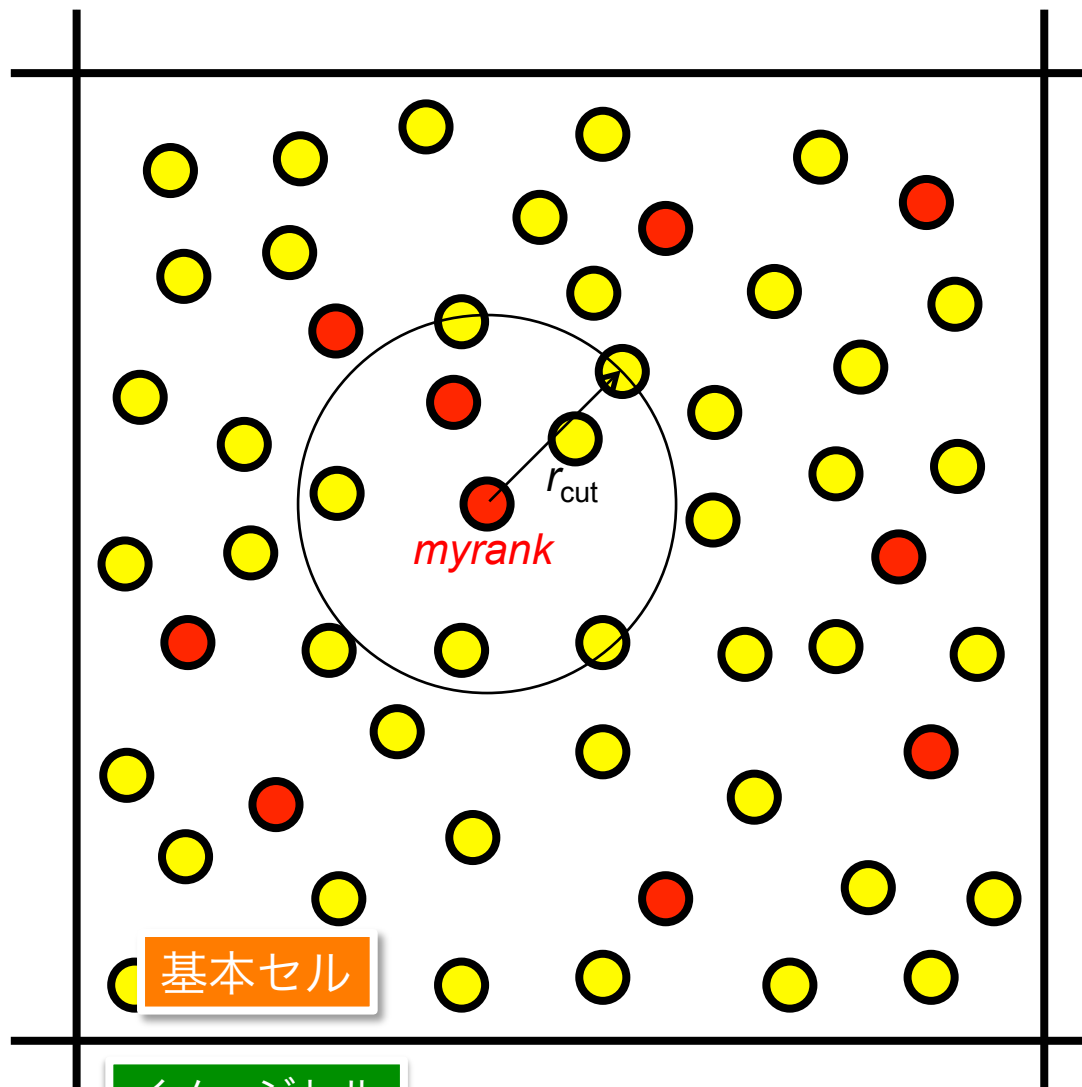
$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i$$

i ごと独立な演算
(粒度 N)

最も単純な MPI 並列化:

```
do i=1+myrank, N, nprocs
  ポテンシャルの計算
  力の計算
  call mpi_barrier
  座標・速度更新
  力学・熱力学量計算
enddo
```

myrank : MPIプロセス番号 (0始まり)
nprocs : MPIプロセス数



分子動力学計算の並列化特性(2) プロセス分割

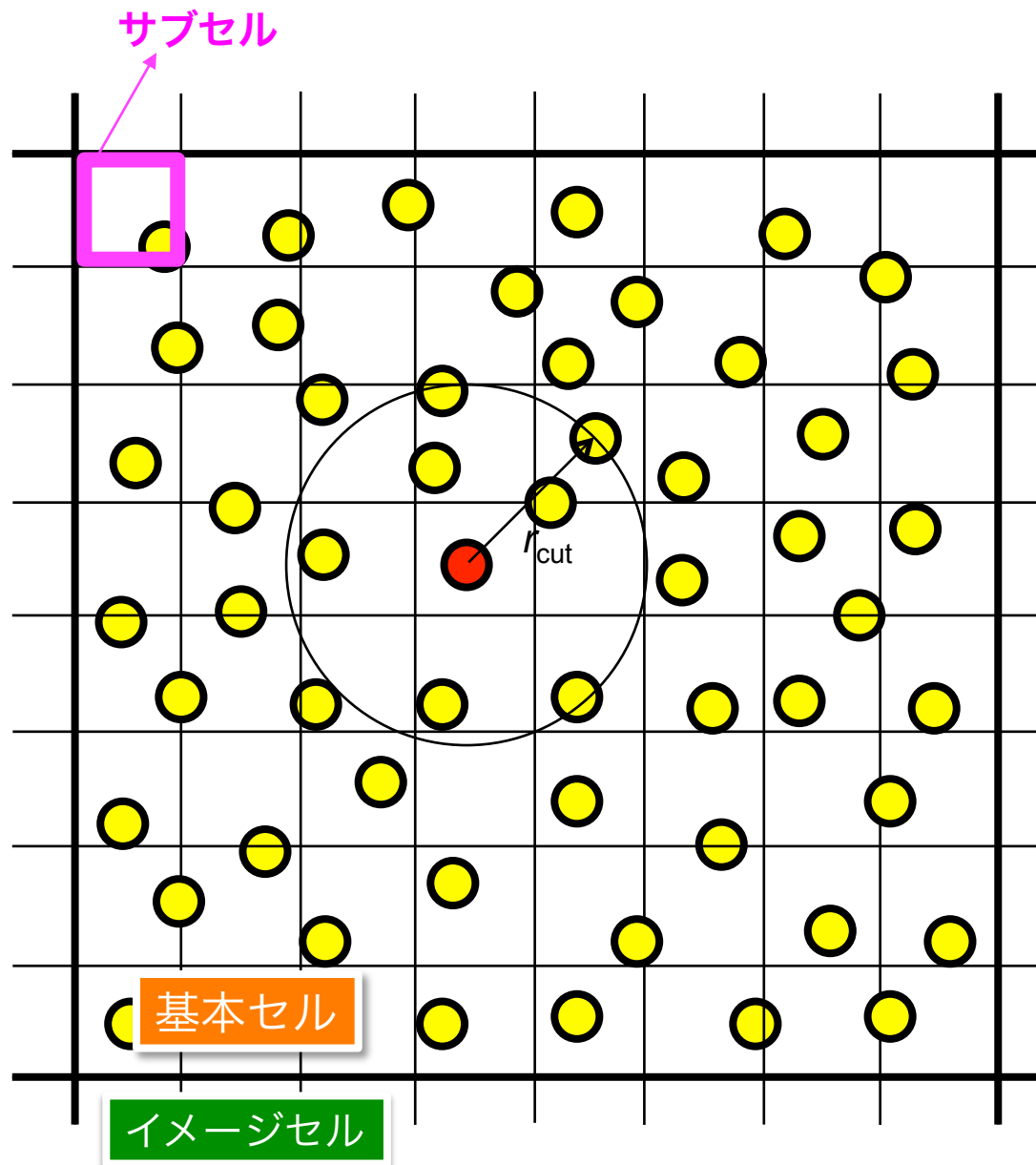
22 / 70

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i$$

i ごと独立な演算
(粒度 N)

空間ドメイン分割による
MPI 並列化:

①基本セルを各辺にそって分割



分子動力学計算の並列化特性(2) プロセス分割

23 / 70

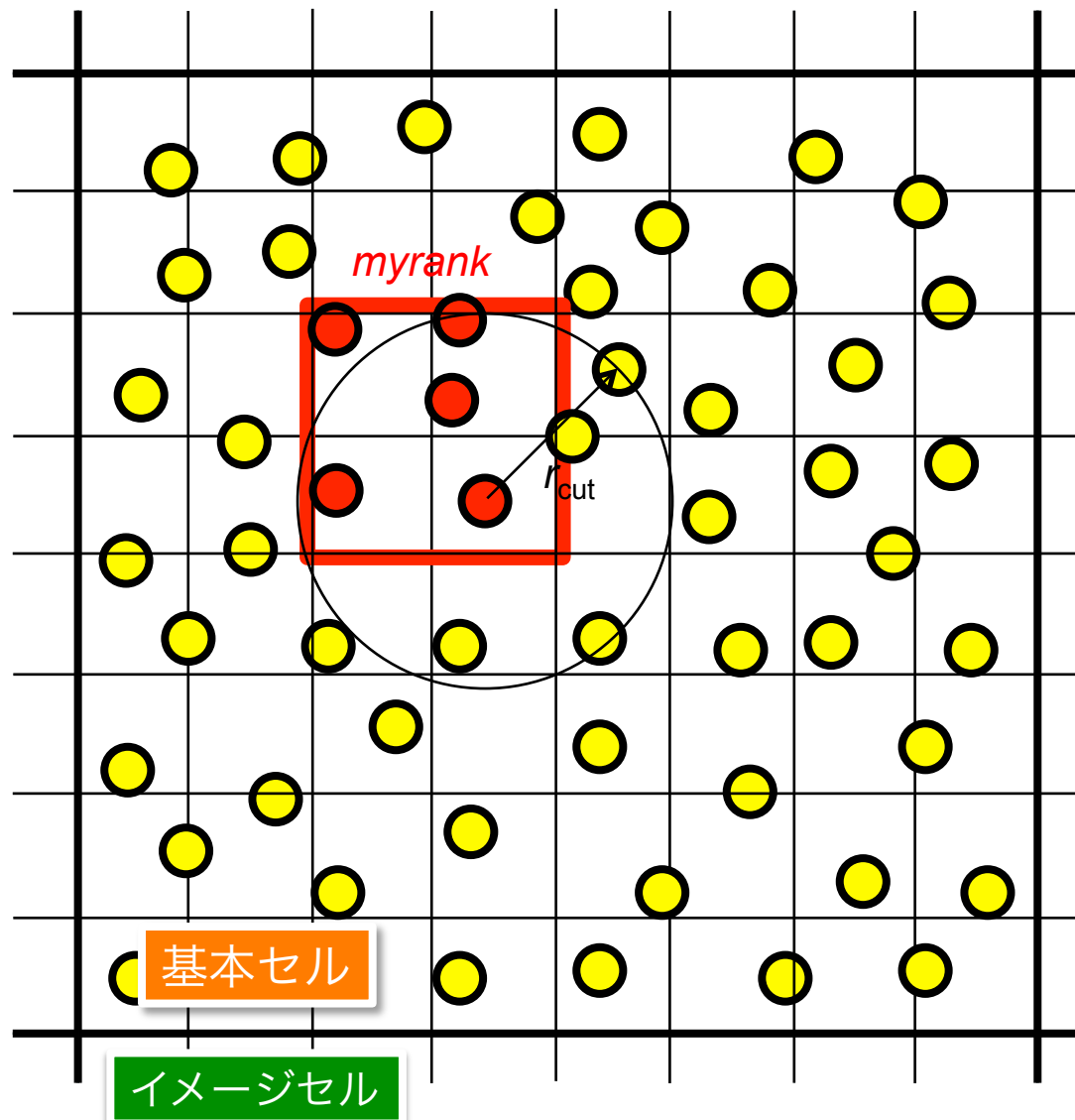
$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i$$

i ごと独立な演算
(粒度 N)

空間ドメイン分割による
MPI 並列化:

- ① 基本セルを各辺にそって分割
- ② 複数のサブセルを各プロセスに均等割り当て

例) プロセスあたり 4 サブセル



分子動力学計算の並列化特性(2) プロセス分割

24 / 70

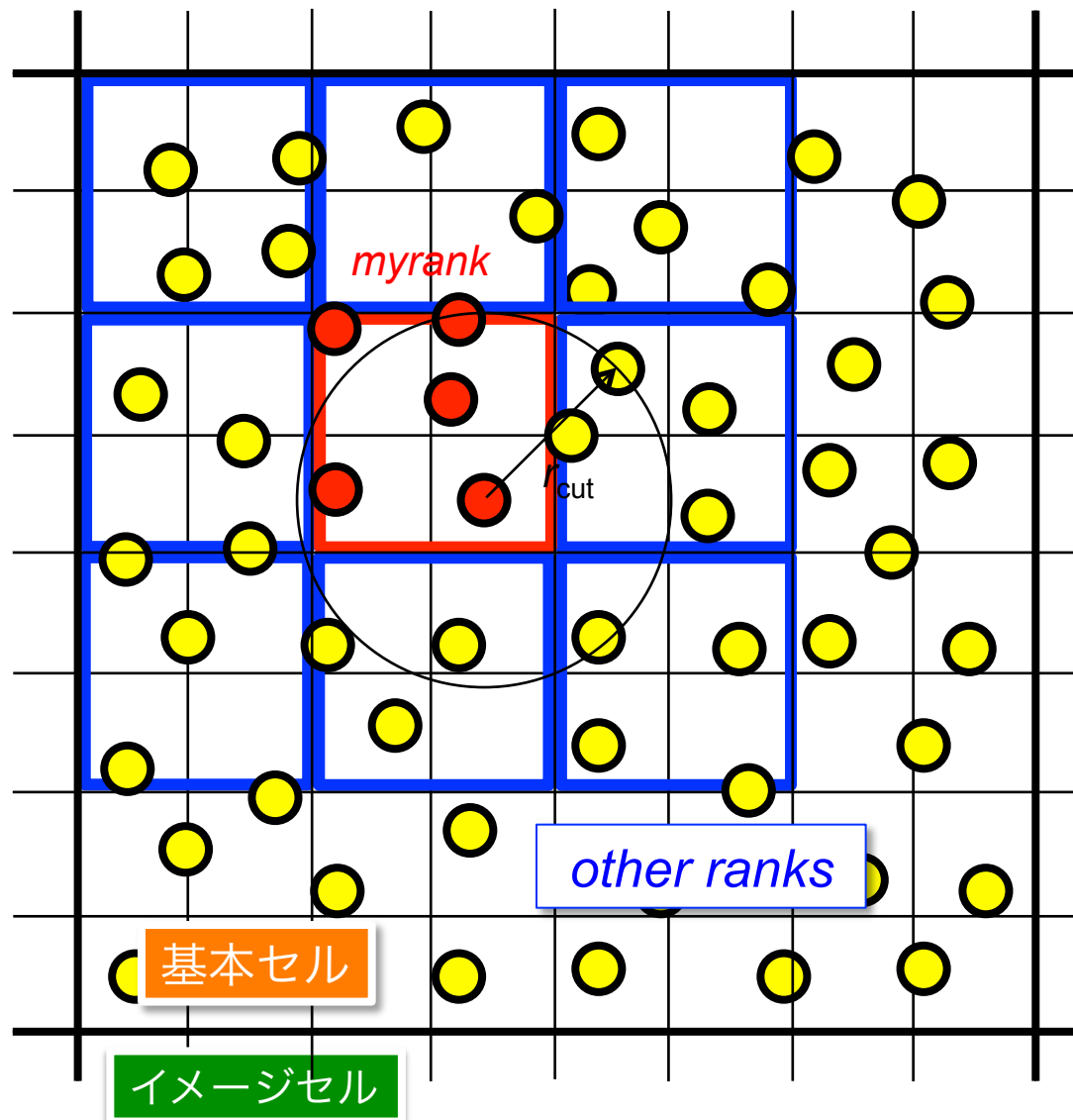
$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i$$

i ごと独立な演算
(粒度 N)

空間ドメイン分割による
MPI 並列化:

- ① 基本セルを各辺にそって分割
- ② 複数のサブセルを各プロセスに均等割り当て
- ③ *myrank* の所持するサブセル内原子と近傍 *other ranks* の所持するサブセル内原子との間で相互作用計算 (カットオフの場合)

例) プロセスあたり 4 サブセル



分子動力学計算の並列化特性(2) プロセス分割

25 / 70

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i$$

i ごと独立な演算
(粒度 N)

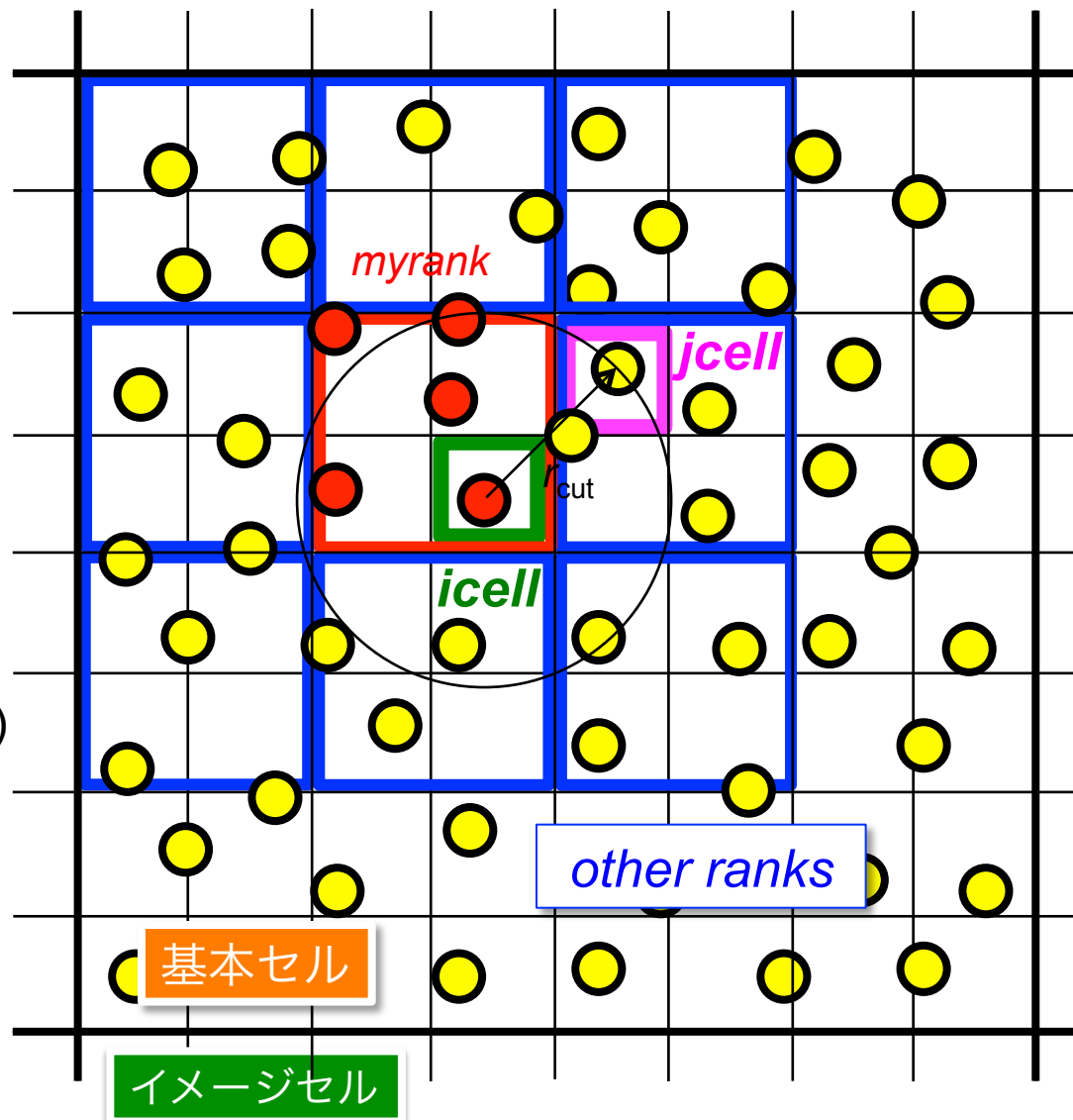
空間ドメイン分割による
MPI 並列化:

コーディングイメージ

```
do icell(myrank)
  do jcell_list(myrank or other ranks)
    ポテンシャルの計算
    力の計算
  enddo
  call mpi_barrier
  座標・速度更新
  力学・熱力学量計算
enddo
```

注) 実際は icell の do
ループは都度閉じて
います。

例) プロセスあたり 4 サブセル



分子動力学計算の並列化特性(2) プロセス分割

26 / 70

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i$$

i ごと独立な演算
(粒度 N)

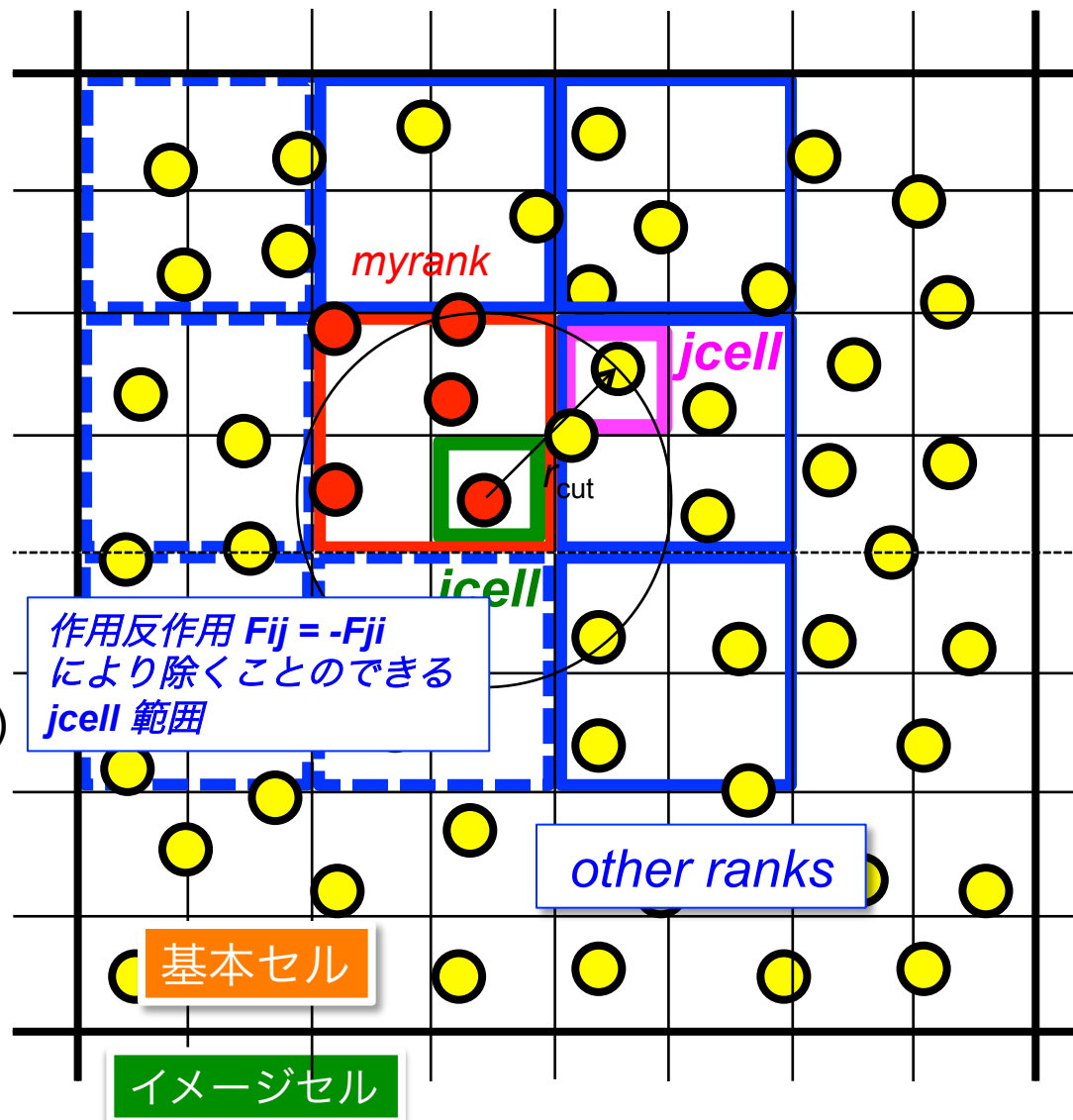
空間ドメイン分割による
MPI 並列化:

コーディングイメージ

```
do icell(myrank)
  do jcell_list(myrank or other ranks)
    ポテンシャルの計算
    力の計算
  enddo
  call mpi_barrier
  座標・速度更新
  力学・熱力学量計算
enddo
```

注) 実際は icell の do
ループは都度閉じて
います。

例) プロセスあたり 4 サブセル

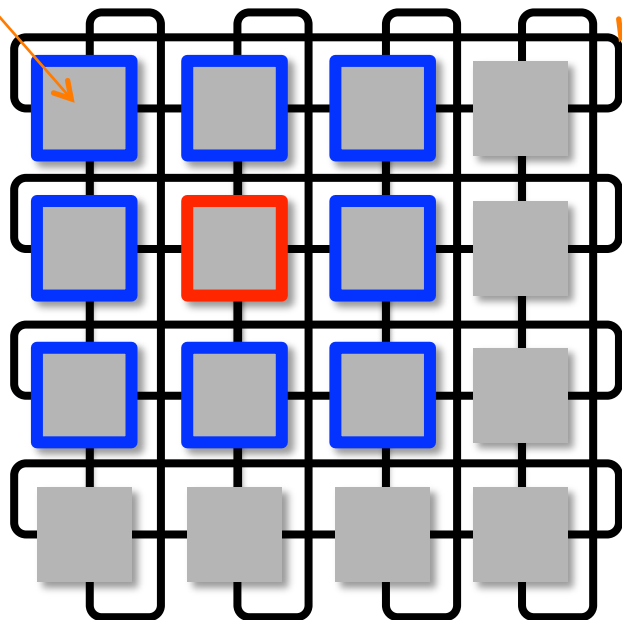


分子動力学計算の並列化特性(2) プロセス分割

27 / 70

計算機 (ノード)

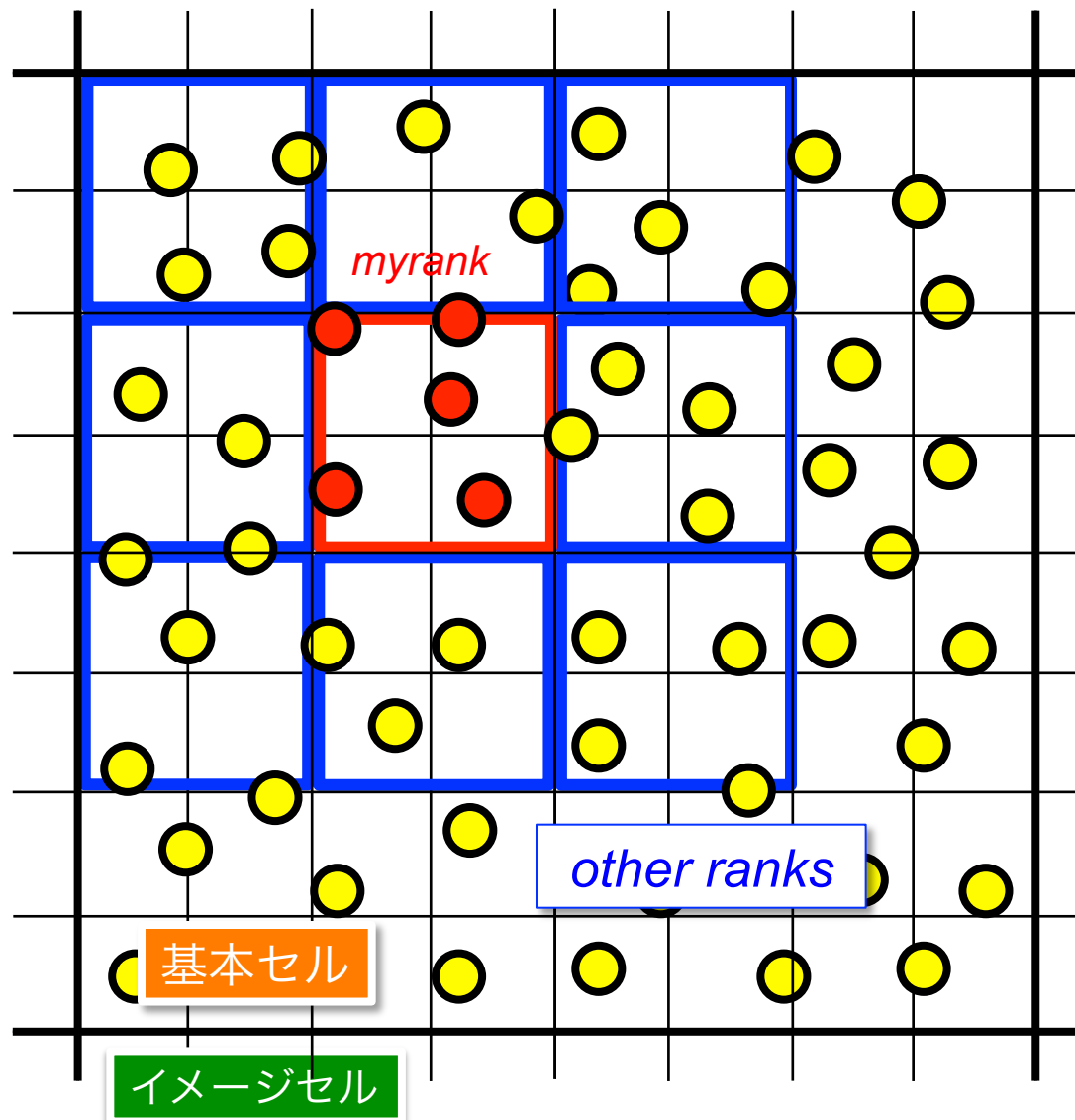
通信ケーブル



トーラスネットワーク

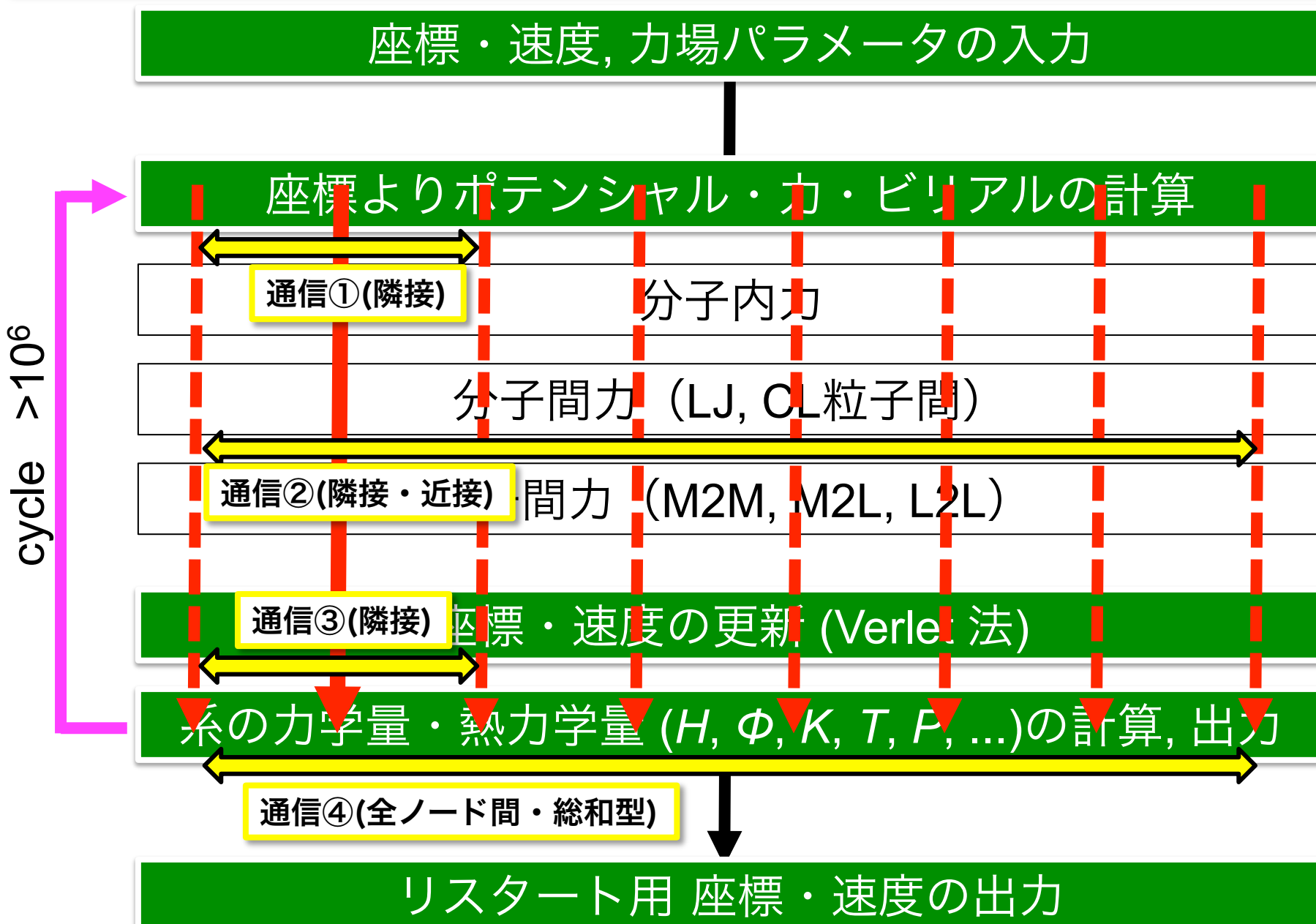
空間ドメイン分割による
MPI 並列化:

ノード形状とプロセス形状とを一致させることを前提に,
トーラスネットワークとの相性が非常に良い



分子動力学計算の並列化特性(3) 通信パターン

28 / 70



分子動力学計算の並列化特性(3) 通信パターン

29 / 70

例) プロセスあたり 4 サブセル

通信①(隣接)

分子内力, 分子間力の計算に必要な相手原子座標 r_j, r_k, r_l の通信.

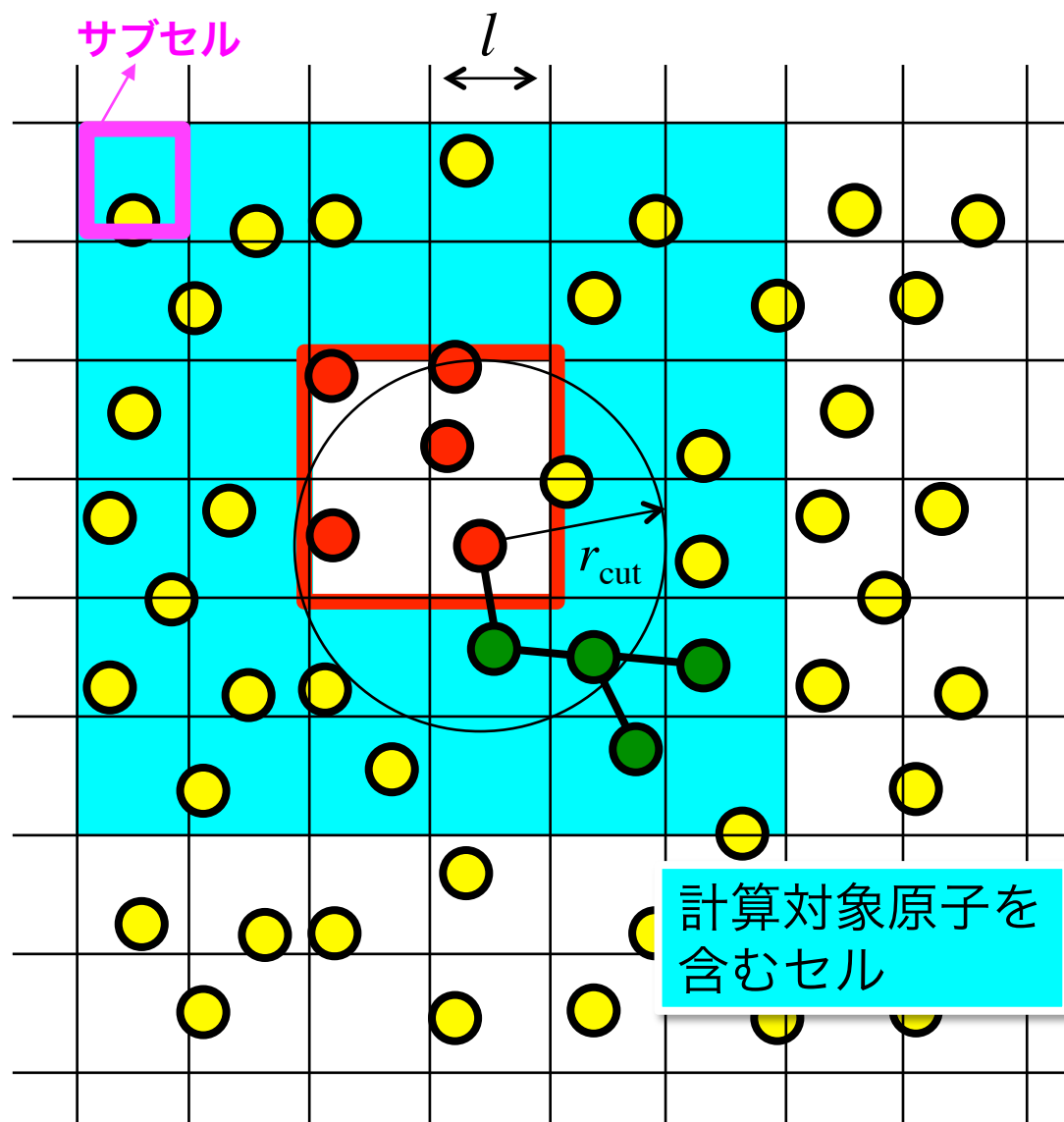
特徴:

- ・ 近傍プロセスとのみ通信
- ・ データサイズは小さい

例) 40 原子/サブセル では,
 $8 \text{ byte}(\text{real} * 8) * 3(\text{xyz}) * 40$
 $= 1 \text{ KB/サブセル}$

注意点:

- ・ $r_{\text{cut}} \leq n * l$ [今回の講義では $n=2$]
- ・ 分子内結合原子 (特に dihedral) を通信範囲内に収める



分子動力学計算の並列化特性(3) 通信パターン

30 / 70

通信①(隣接)

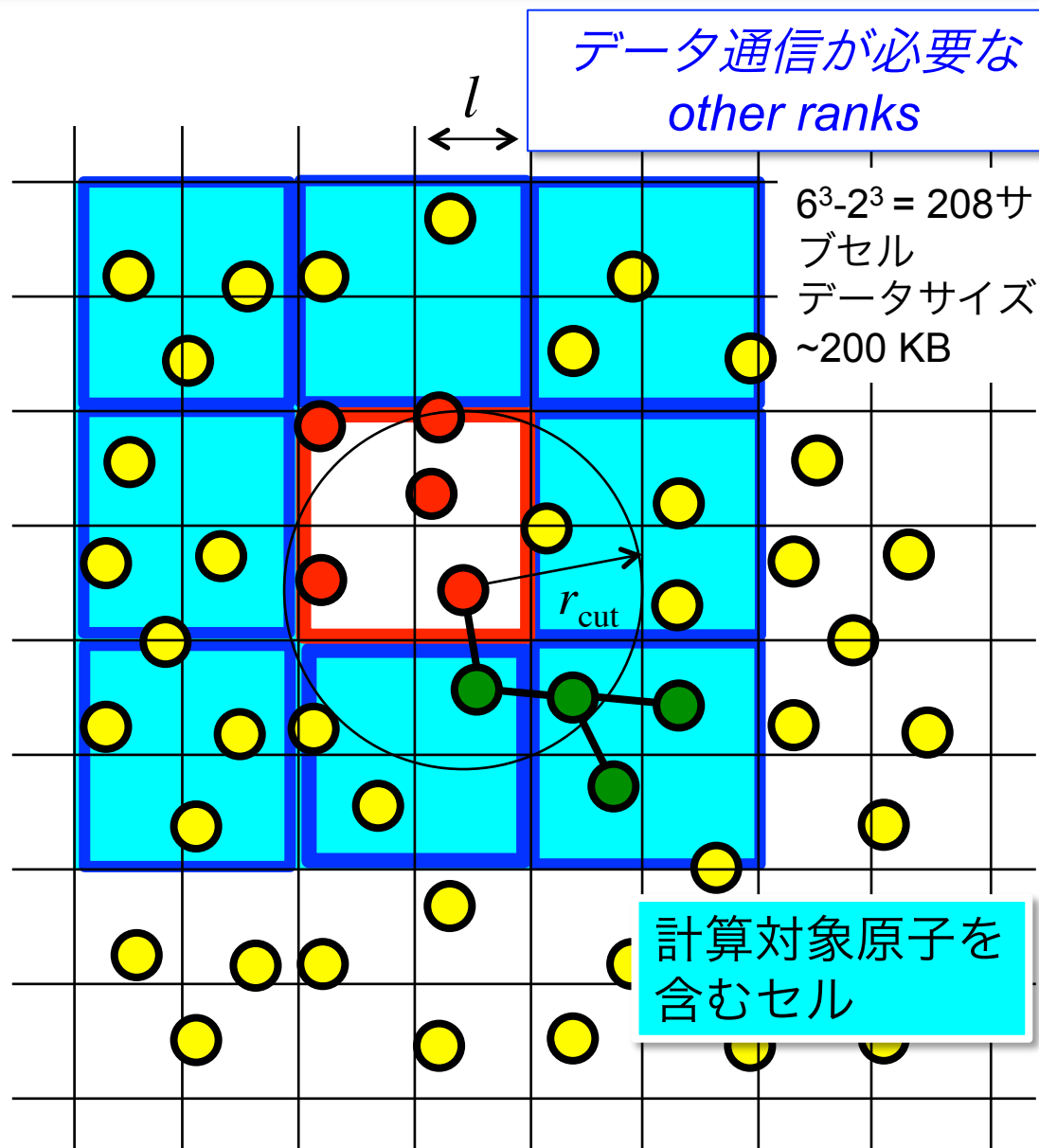
分子内力, 分子間力の計算に必要な相手原子座標 r_j, r_k, r_l の通信.

特徴:

- ・ 近傍プロセスとのみ通信
- ・ データサイズは小さい
例) 40 原子/サブセル では,
8 byte(real*8)*3(xyz)*40
= 1 KB/サブセル

注意点:

- ・ $r_{\text{cut}} \leq n * l$ [今回の講義では $n=2$]
- ・ 分子内結合原子 (特に dihedral) を通信範囲内に収める



分子動力学計算の並列化特性(3) 通信パターン

31 / 70

通信①(隣接)

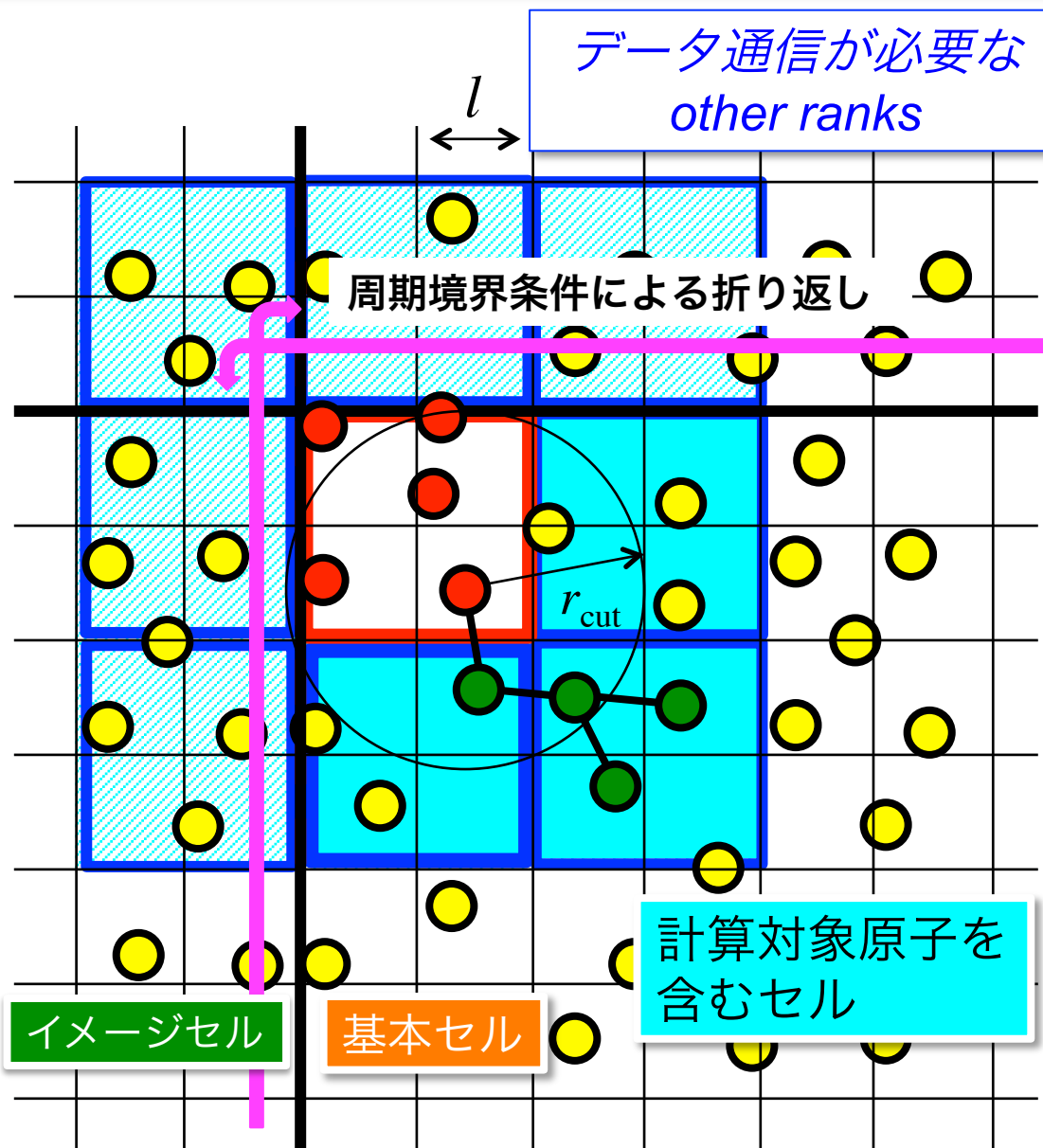
分子内力, 分子間力の計算に必要な相手原子座標 r_j, r_k, r_l の通信.

特徴:

- ・ 近傍プロセスとのみ通信
- ・ データサイズは小さい
例) 40 原子/サブセル では,
8 byte(real*8)*3(xyz)*40
= 1 KB/サブセル

注意点:

- ・ $r_{cut} \leq n * l$ [今回の講義では $n=2$]
- ・ 分子内結合原子 (特に dihedral) を通信範囲内に収める
- ・ 周期境界条件による折り返し



分子動力学計算の並列化特性(3) 通信パターン

32 / 70

レベル 0

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

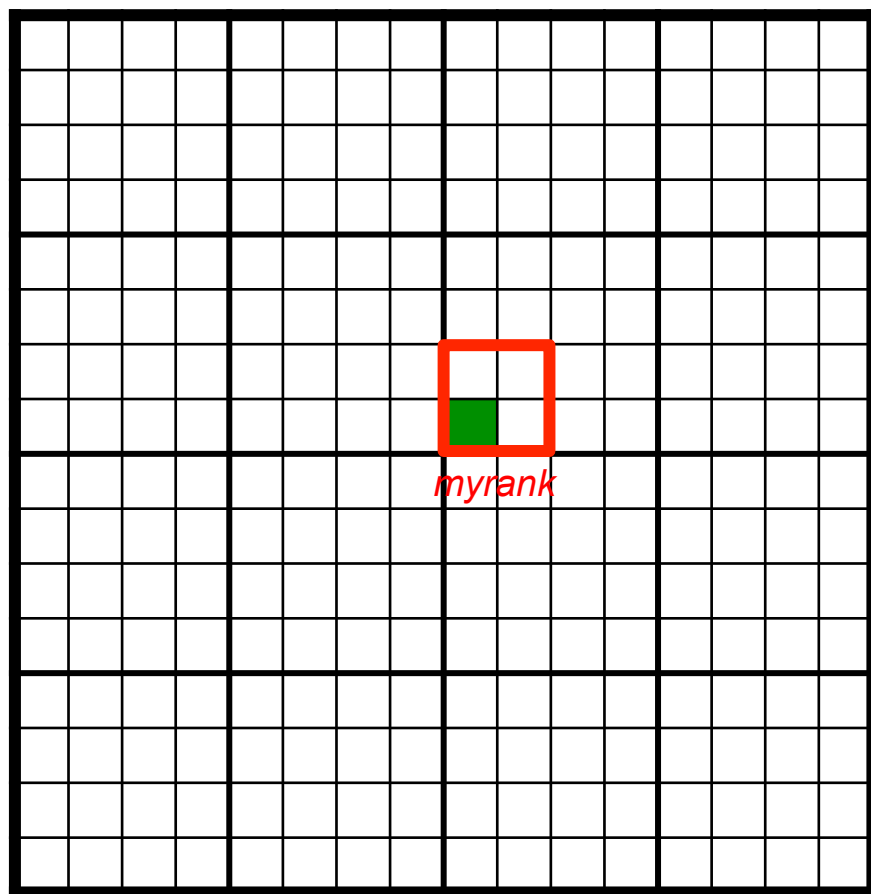
- ・ データサイズ

例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

階層ごと $875 * 0.4 \text{ KB} = 320 \text{ KB}$



分子動力学計算の並列化特性(3) 通信パターン

33 / 70

レベル 0

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

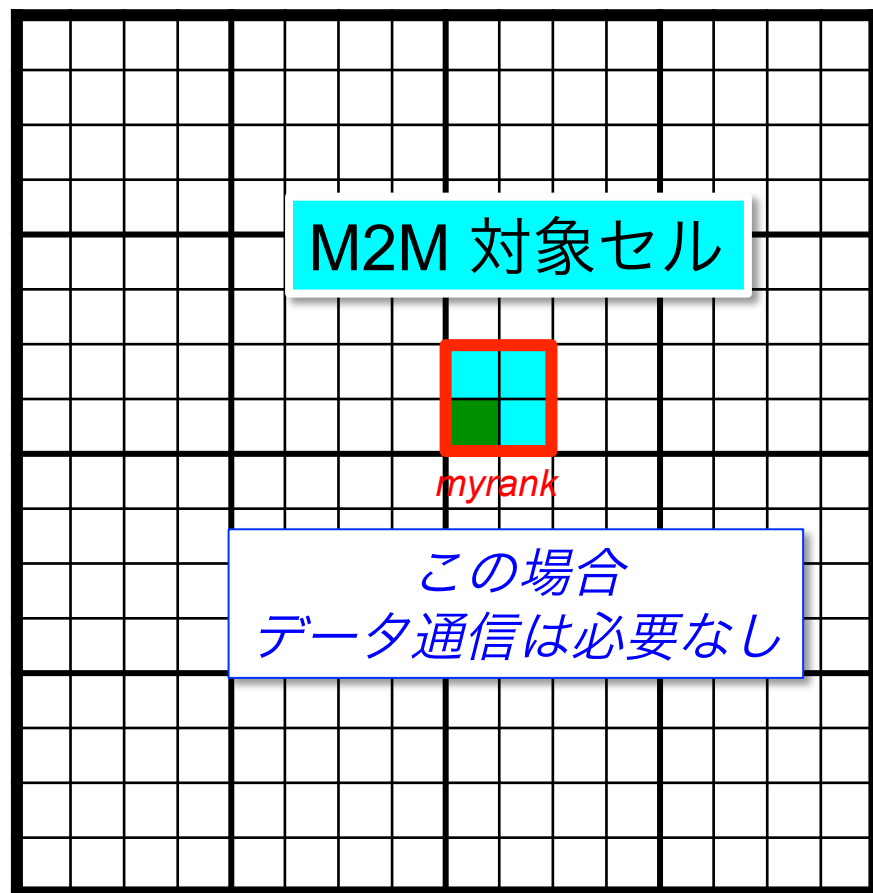
- ・ データサイズ

例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

階層ごと $875 * 0.4 \text{ KB} = 320 \text{ KB}$



分子動力学計算の並列化特性(3) 通信パターン

34 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

- ・ データサイズ

例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

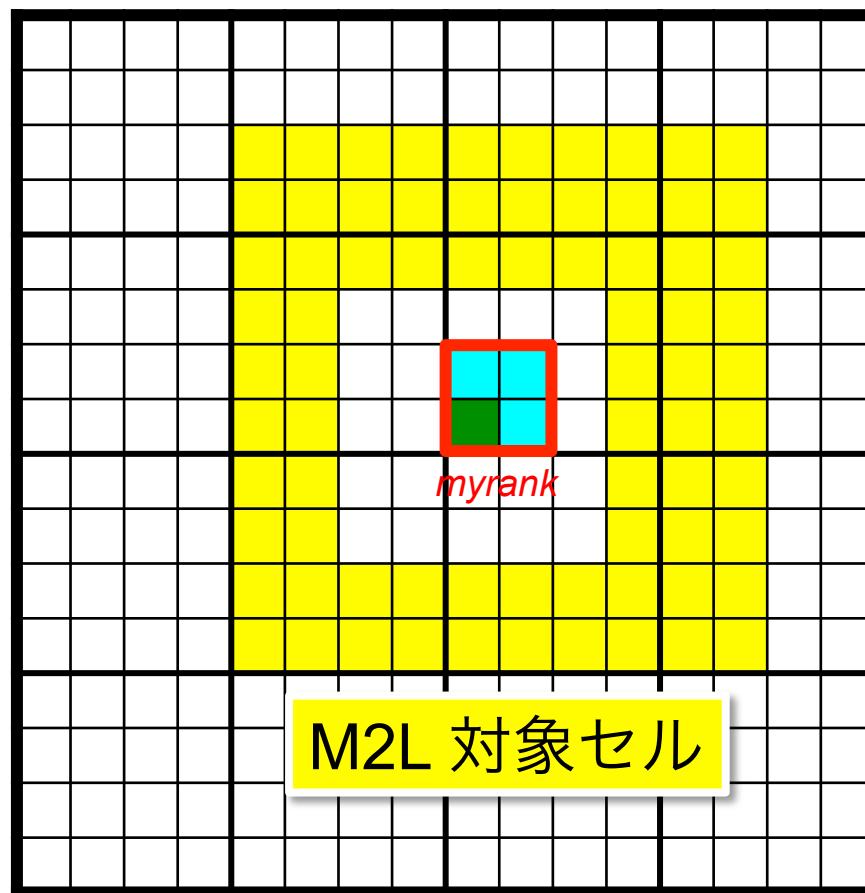
$$= 0.4 \text{ KB/スーパーセル}$$

階層ごと $875 * 0.4 \text{ KB} = 320 \text{ KB}$

レベル 0

注意

プロセス内自セル位置
(0,0), (1,0), (0,1), (1,1)



分子動力学計算の並列化特性(3) 通信パターン

35 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

- ・ データサイズ

例) $n_{max}=4$ では

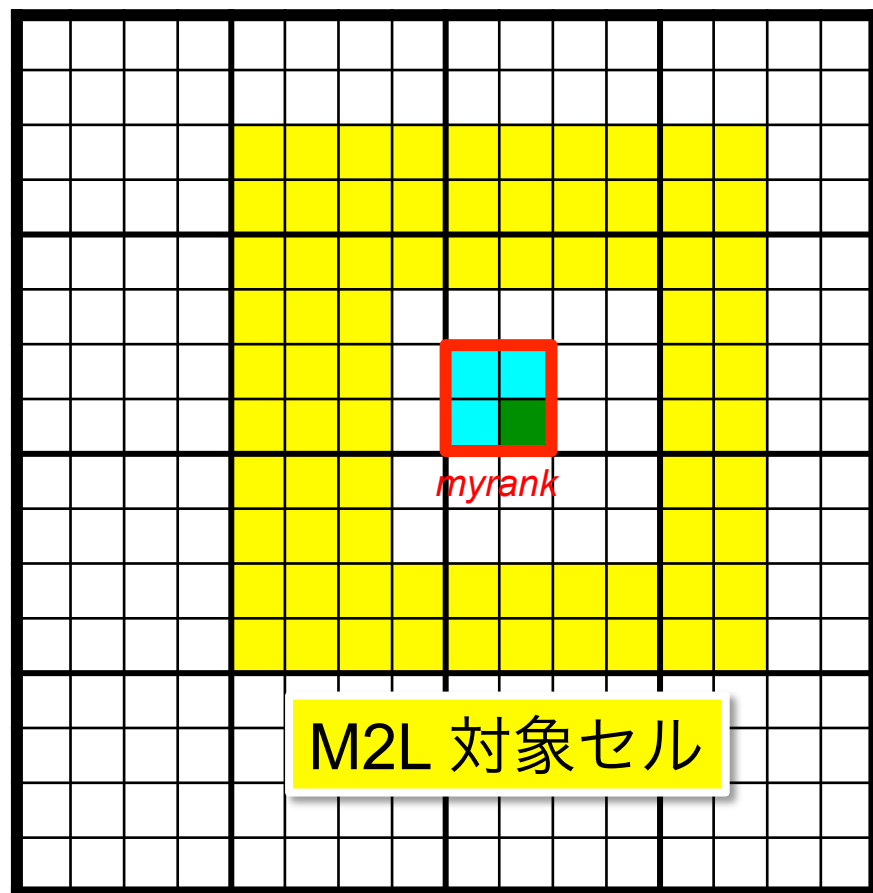
$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

$$\text{階層ごと } 875 * 0.4 \text{ KB} = 320 \text{ KB}$$

レベル 0

プロセス内自セル位置
(0,0), (1,0), (0,1), (1,1)



M2L 対象セル

分子動力学計算の並列化特性(3) 通信パターン

36 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

- ・ データサイズ

例) $n_{max}=4$ では

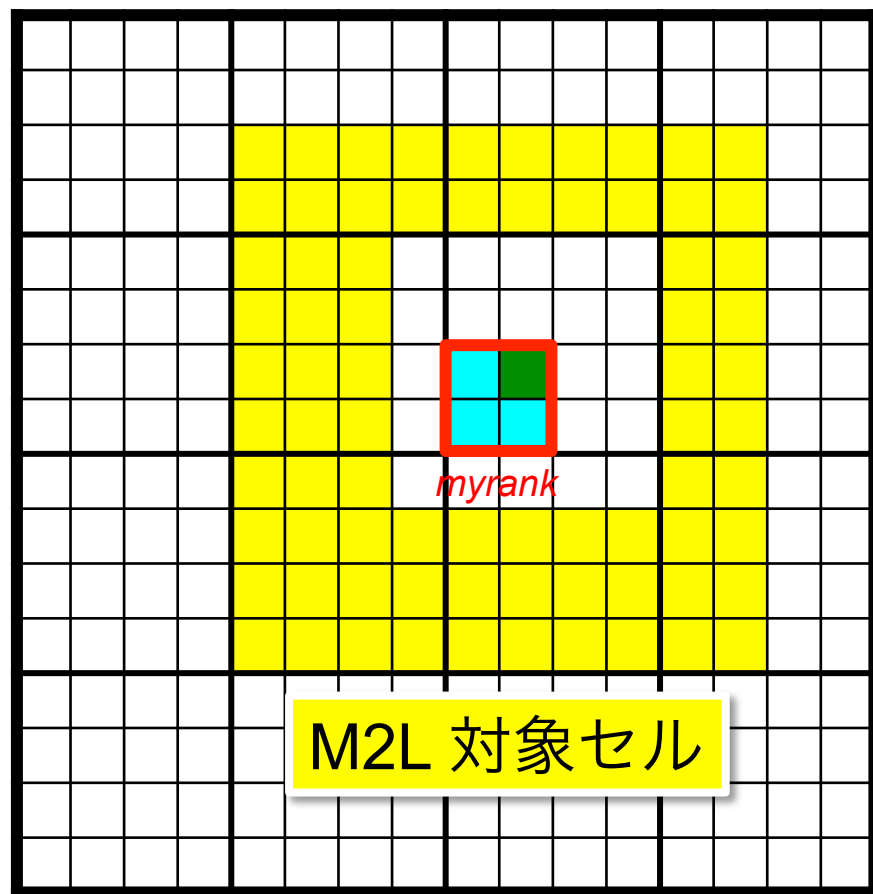
$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

$$\text{階層ごと } 875 * 0.4 \text{ KB} = 320 \text{ KB}$$

レベル 0

プロセス内自セル位置
(0,0), (1,0), (0,1), (1,1)



分子動力学計算の並列化特性(3) 通信パターン

37 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

- データサイズ
例) $n_{max}=4$ では

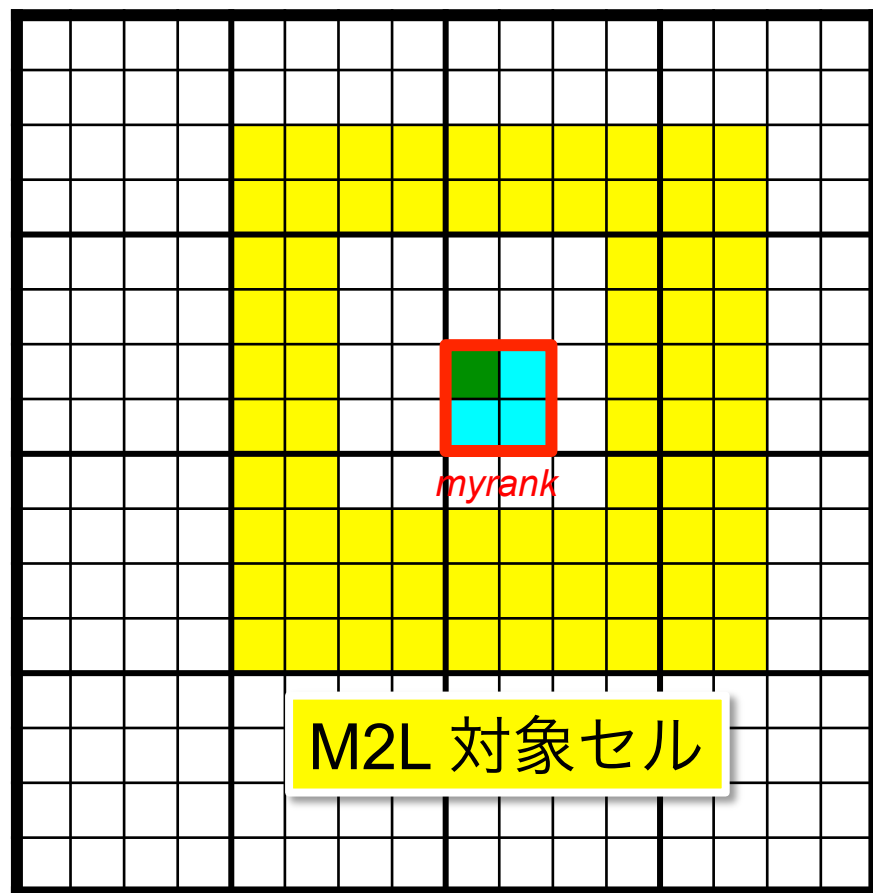
$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

$$\text{階層ごと } 875 * 0.4 \text{ KB} = 320 \text{ KB}$$

レベル 0

プロセス内自セル位置
(0,0), (1,0), (0,1), (1,1)



分子動力学計算の並列化特性(3) 通信パターン

38 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

- ・ データサイズ
例) $n_{max}=4$ では

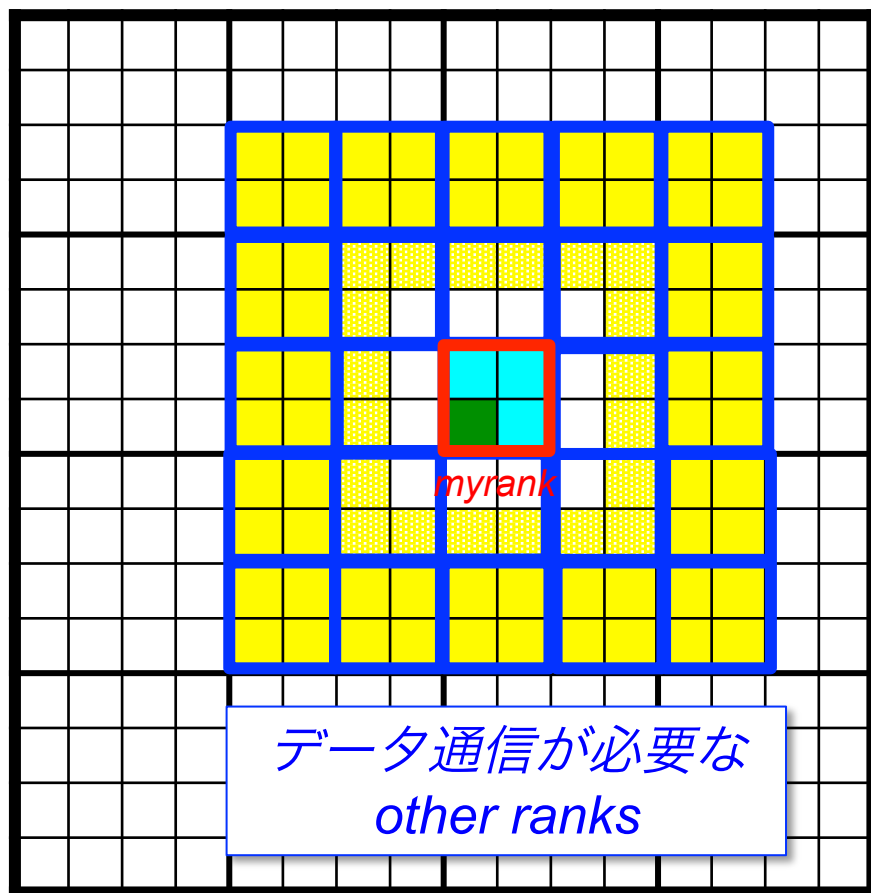
$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

$$\text{階層ごと } 875 * 0.4 \text{ KB} = 320 \text{ KB}$$

レベル 0

プロセス内自セル位置
(0,0), (1,0), (0,1), (1,1)



分子動力学計算の並列化特性(3) 通信パターン

39 / 70

レベル 1

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

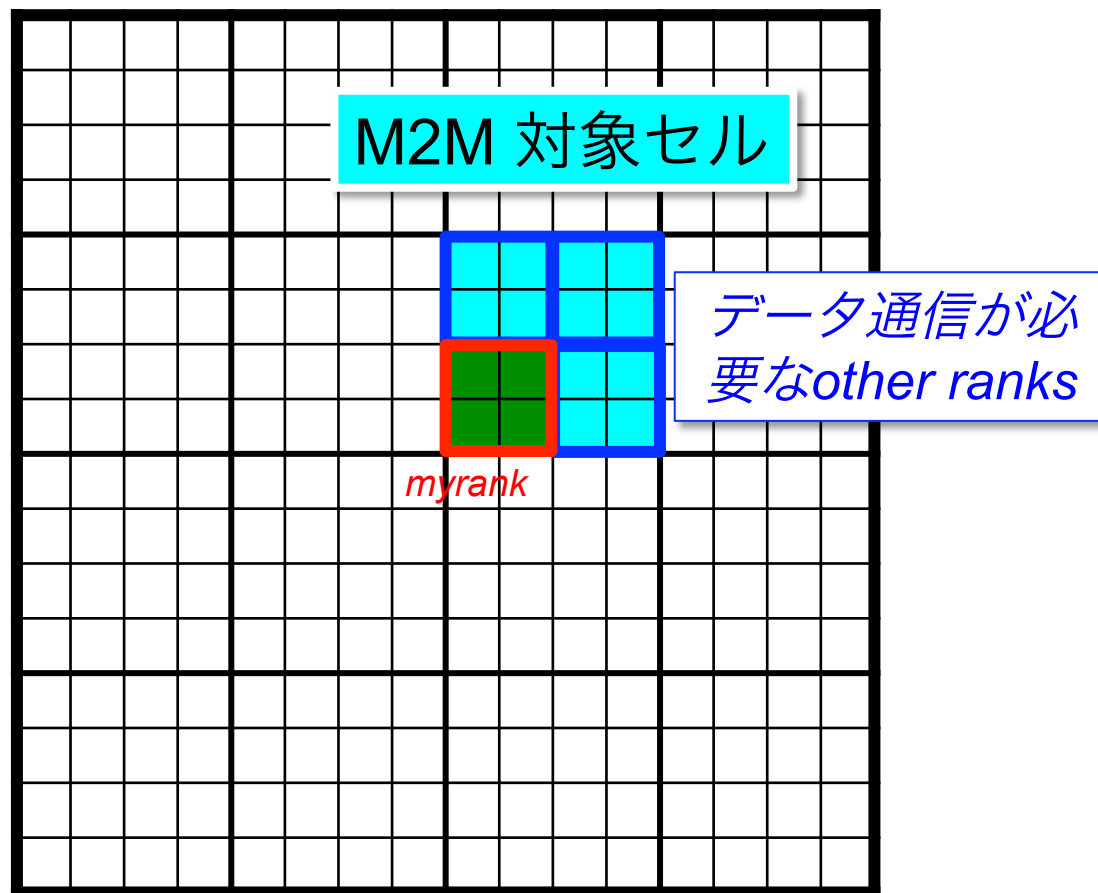
- ・ データサイズ

例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

階層ごと $875 * 0.4 \text{ KB} = 320 \text{ KB}$



分子動力学計算の並列化特性(3) 通信パターン

40 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

- ・ データサイズ

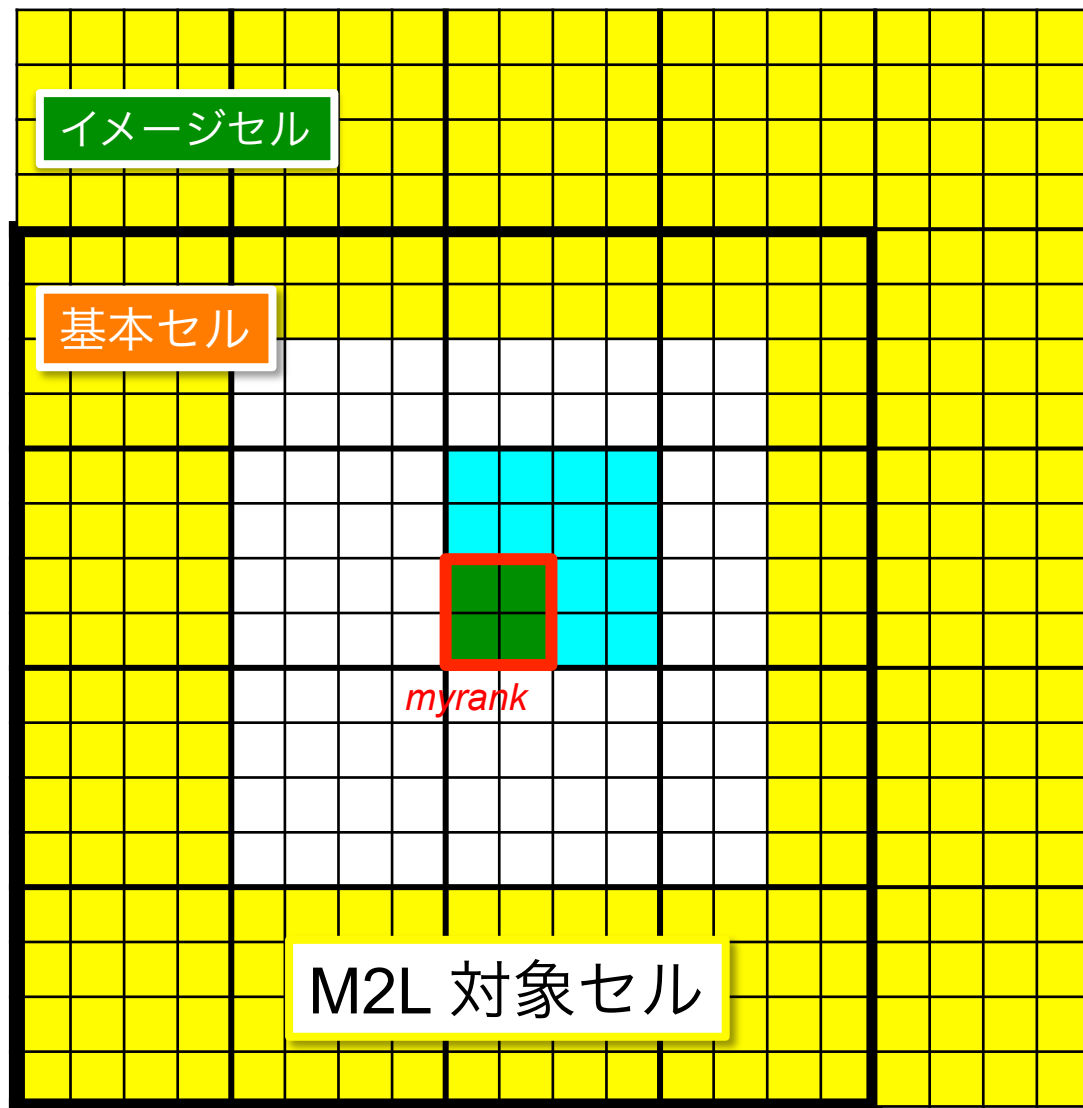
例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

階層ごと $875 * 0.4 \text{ KB} = 320 \text{ KB}$

レベル 1



分子動力学計算の並列化特性(3) 通信パターン

41 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・通信対象範囲が階層ごと異なる.
上位階層ほど遠方.

- ・データサイズ

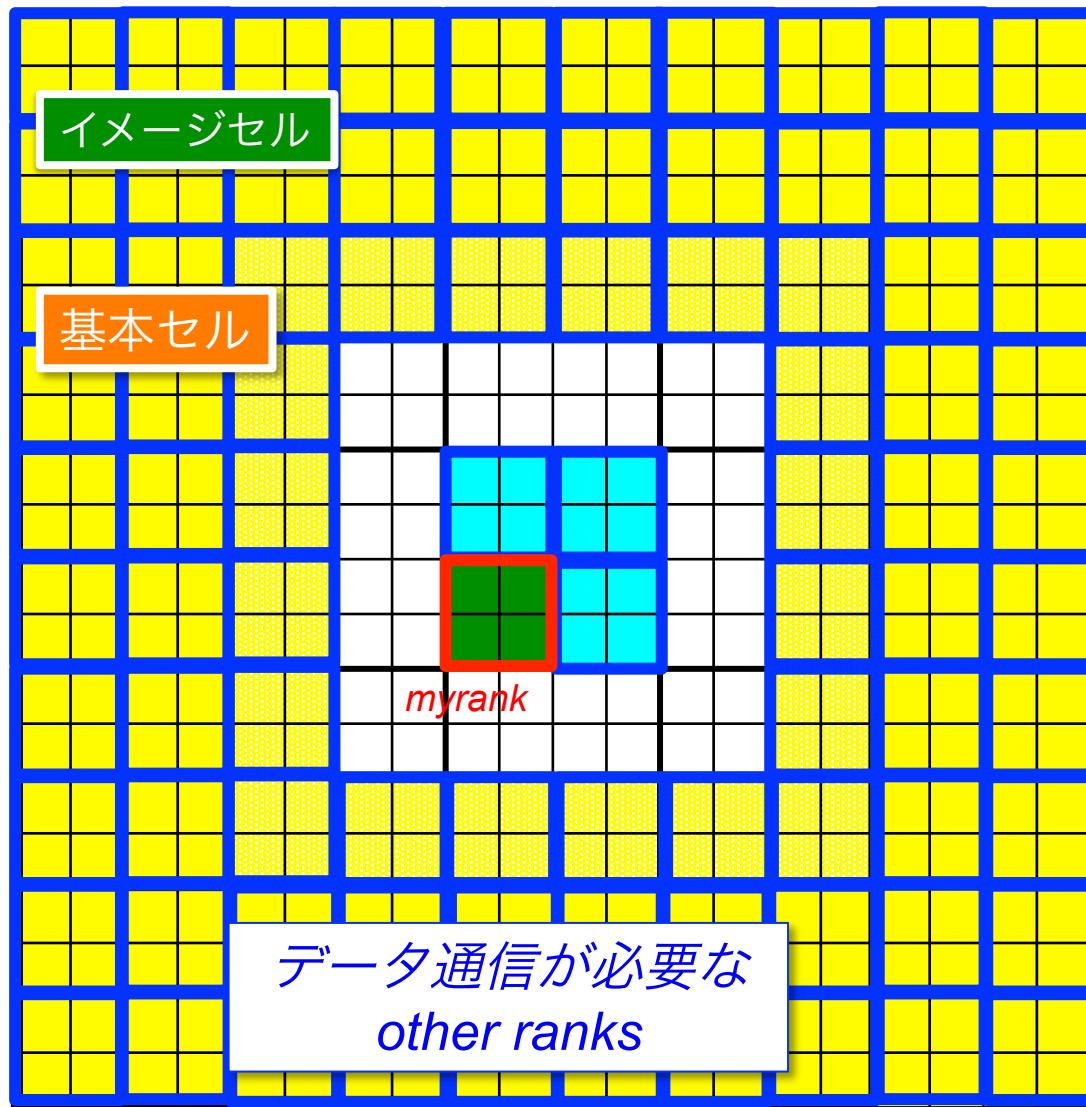
例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

階層ごと $875 * 0.4 \text{ KB} = 320 \text{ KB}$

レベル 1



分子動力学計算の並列化特性(3) 通信パターン

42 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる。
上位階層ほど遠方。
- ・ データサイズ
例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

$$= 0.4 \text{ KB/スーパーセル}$$

$$\text{階層ごと } 875 * 0.4 \text{ KB} = 320 \text{ KB}$$

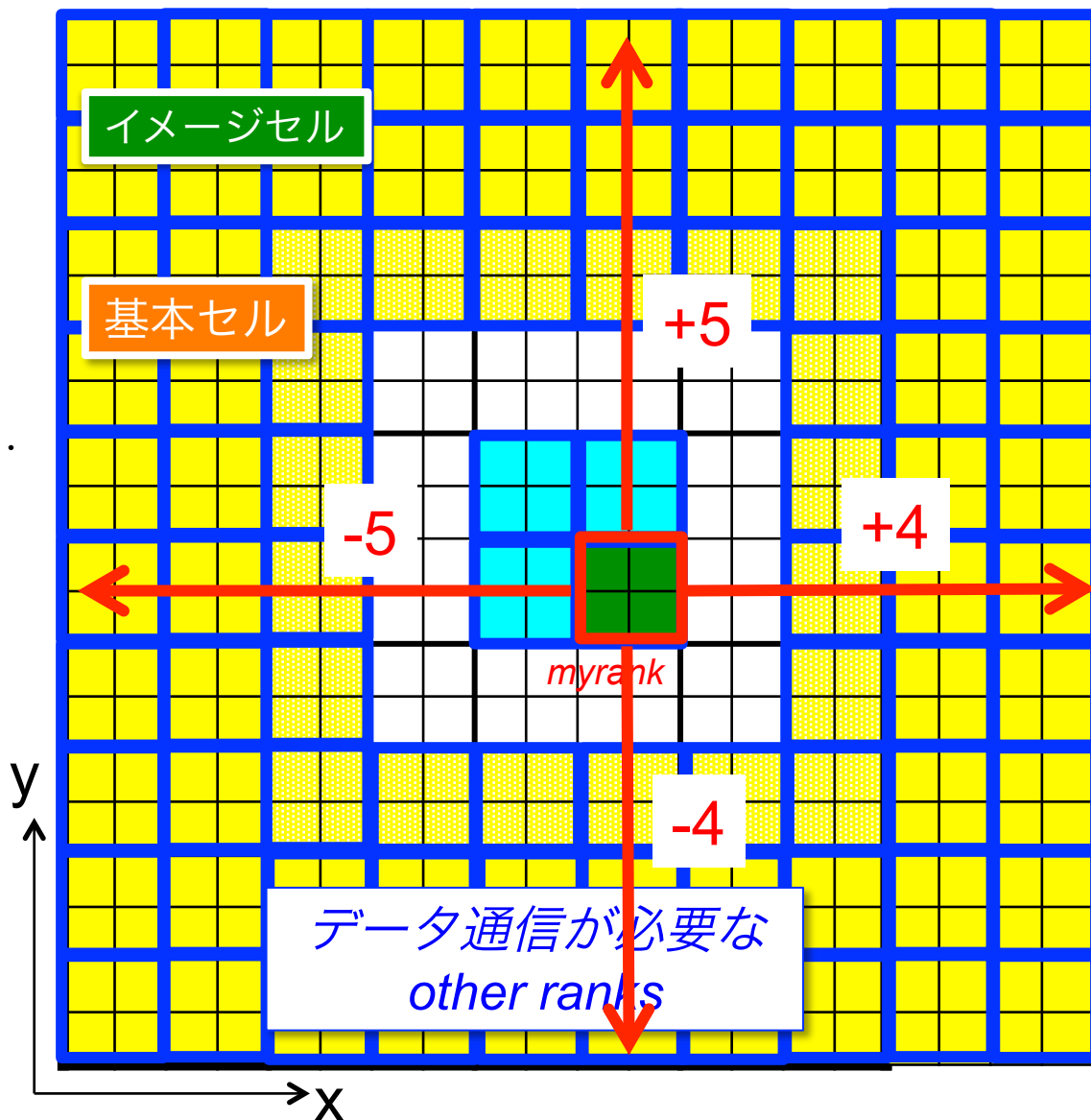
注意点 (上位階層):

- ・ 軸ごとrank参照範囲が異なる

右の例では, x方向 -5 ~ +4

y方向 -4 ~ +5

レベル 1



分子動力学計算の並列化特性(3) 通信パターン

43 / 70

通信②(隣接・近接)

多極子情報の通信

M2M: 周辺 8 スーパーセル

M2L : 周辺 875 スーパーセル

特徴:

- ・ 通信対象範囲が階層ごと異なる
上位階層ほど遠方.

- ・ データサイズ

例) $n_{max}=4$ では

$$16 \text{ byte}(\text{complex} * 8) * (4+1)^2$$

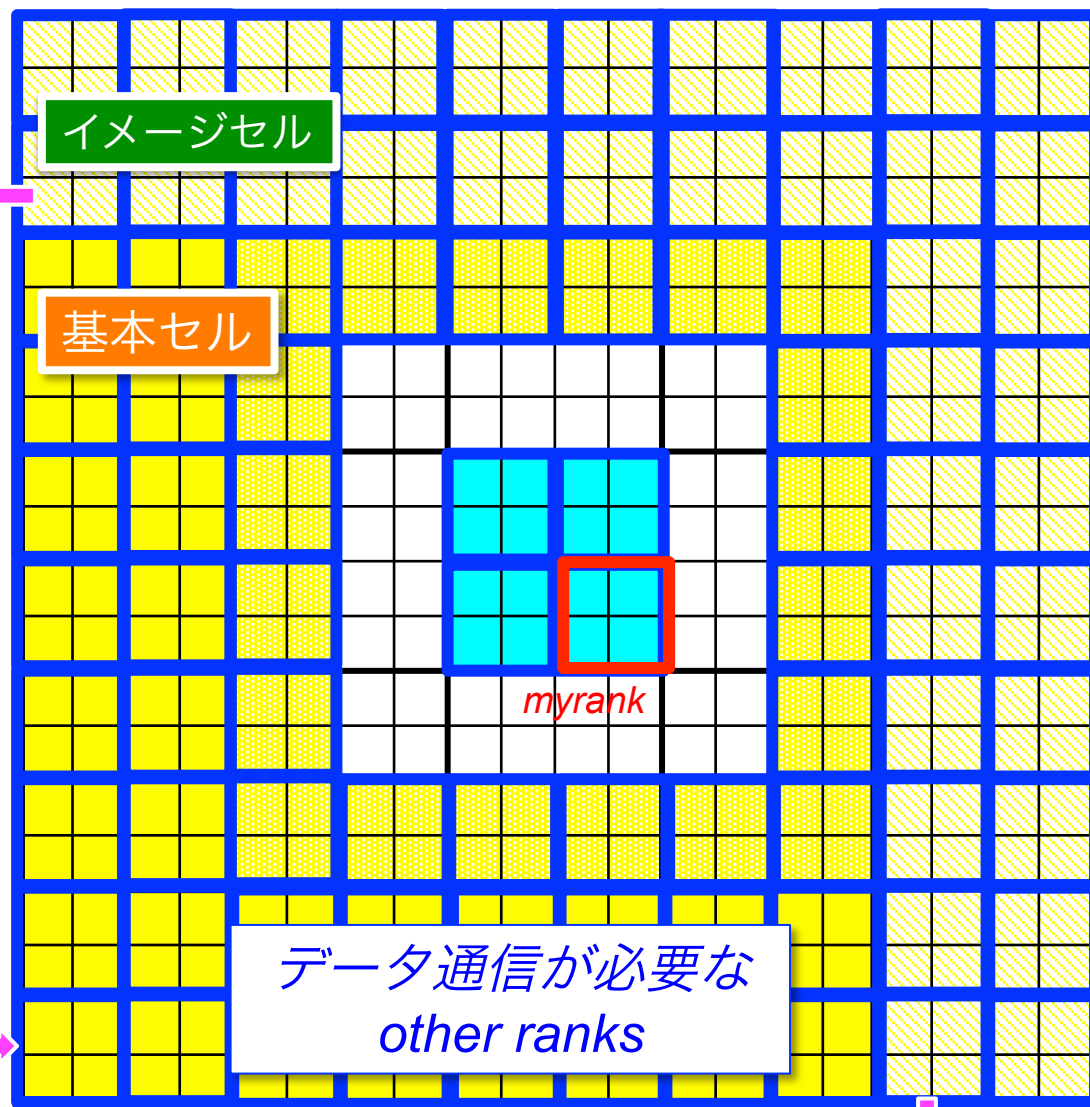
$$= 0.4 \text{ KB/スーパーセル}$$

階層ごと $875 * 0.4 \text{ KB} = 320 \text{ KB}$

注意点:

- ・ 軸ごとrank参照範囲が異なる
- ・ 周期境界条件による折り返し

レベル 1



周期境界条件による折り返し

分子動力学計算の並列化特性(3) 通信パターン

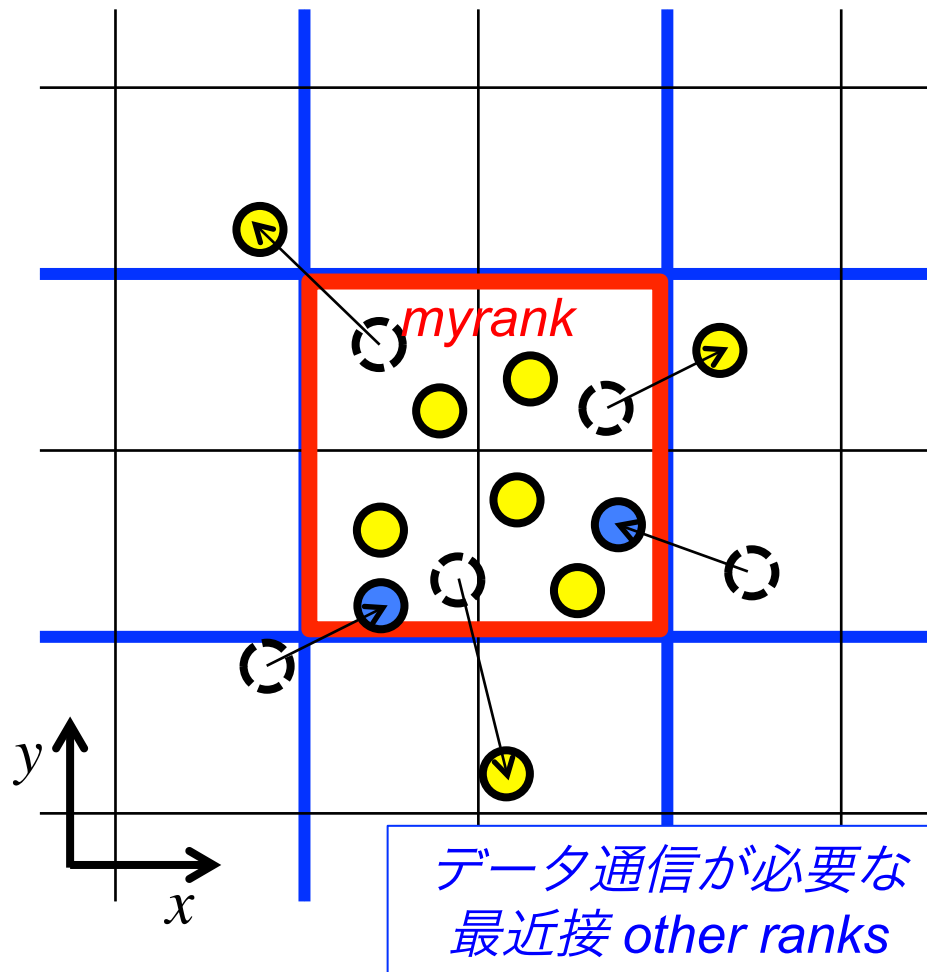
44 / 70

通信③(隣接)

運動方程式の数値積分の結果, 原子が移動. これを各プロセスに再帰属させるための座標通信.

特徴:

- ・ 最近接 $3^3-1=26$ プロセスと通信
- ・ データサイズは非常に小さい
- ・ myrank から出る粒子, myrank に入ってくる粒子, 両方ある



分子動力学計算の並列化特性(3) 通信パターン

45 / 70

通信③(隣接)

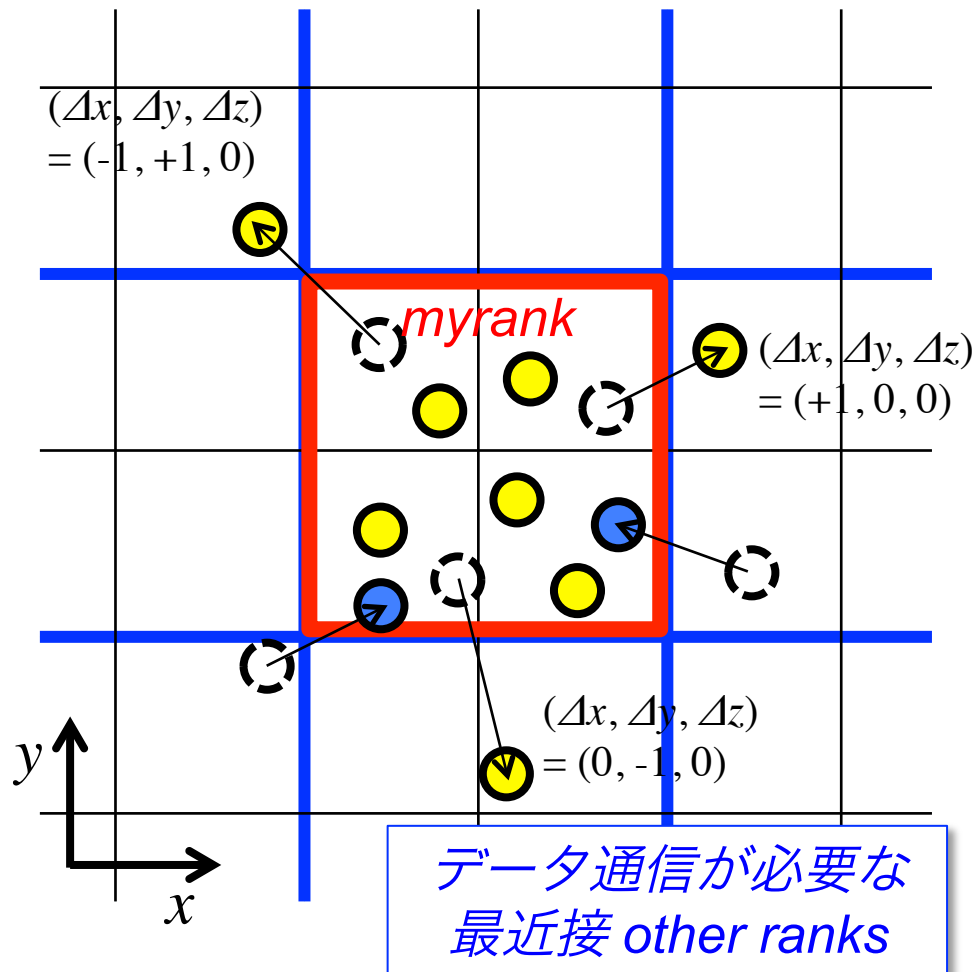
運動方程式の数値積分の結果, 原子が移動. これを各プロセスに再帰属させるための座標通信.

特徴:

- ・ 最近接 $3^3-1=26$ プロセスと通信
- ・ データサイズは非常に小さい
- ・ myrank から出る粒子, myrank に入ってくる粒子, 両方ある

注意点:

- ・ x, y, z 軸方向への移動に加え, 斜め方向への移動が発生



分子動力学計算の並列化特性(3) 通信パターン

46 / 70

通信③(隣接)

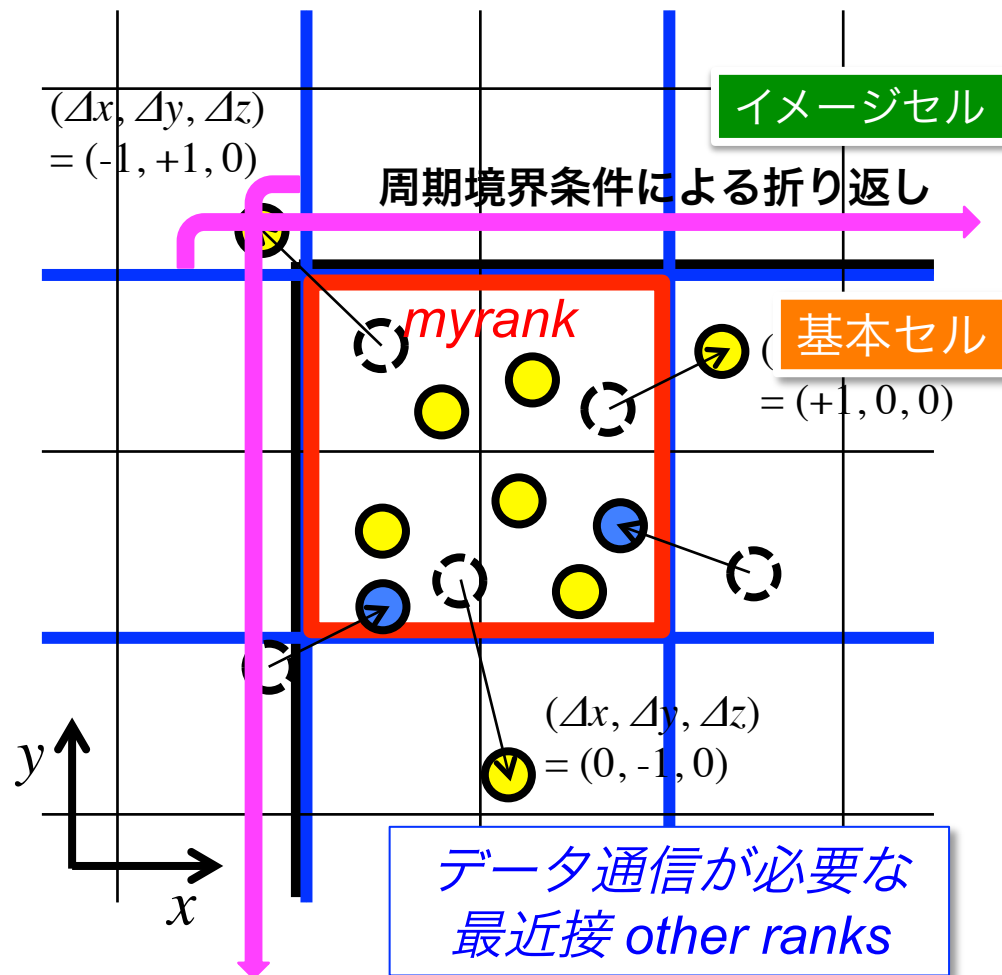
運動方程式の数値積分の結果, 原子が移動. これを各プロセスに再帰属させるための座標通信.

特徴:

- ・最近接 $3^3-1=26$ プロセスと通信
- ・データサイズは非常に小さい
- ・myrank から出る粒子, myrank に入ってくる粒子, 両方ある

注意点:

- ・x, y, z 軸方向への移動に加え, 斜め方向への移動が発生
- ・周期境界条件による折り返し



分子動力学計算の並列化特性(3) 通信パターン

47 / 70

通信④(全ノード間・総和型)

系全体での量（ハミルトニアン, 温度, 圧力など）を計算するための通信

特徴:

- ・ データサイズは, real*8 の 1 変数
- ・ 全プロセス間で総和をとる (allreduce)

→ハードウェア機能の利用で高速化

例えば, 京の TBI (Tofu barrier interface)

~10,000ノード reduce 演算 10 μ s

安島ら, FUJITSU. 63, 3, p. 260-264(05, 2012)

例) 運動エネルギー

$$K = \sum_{i=1}^N \frac{1}{2} m_i v_i^2 = \sum_{k=1}^{nprocs} \sum_{i=1}^{N_k} \frac{1}{2} m_i v_i^2$$

kene プロセスごとの和
wk_kene

```
call mpi_allreduce(wk_kene,kene,1,
  mpi_double_precision,mpi_sum,
  mpi_comm_world,ierr)
```

分子動力学計算の並列化特性(3) 通信パターン

48 / 70

通信④(全ノード間・総和型)

系全体での量（ハミルトニアン, 温度, 圧力など）を計算するための通信

特徴:

- ・ データサイズは, real*8 の 1 変数
- ・ 全プロセス間で総和をとる (allreduce)

→ハードウェア機能の利用で高速化

例えば, 京の TBI (Tofu barrier interface)

~10,000ノード reduce 演算 10 μ s

安島ら, FUJITSU. 63, 3, p. 260-264(05, 2012)

注意点:

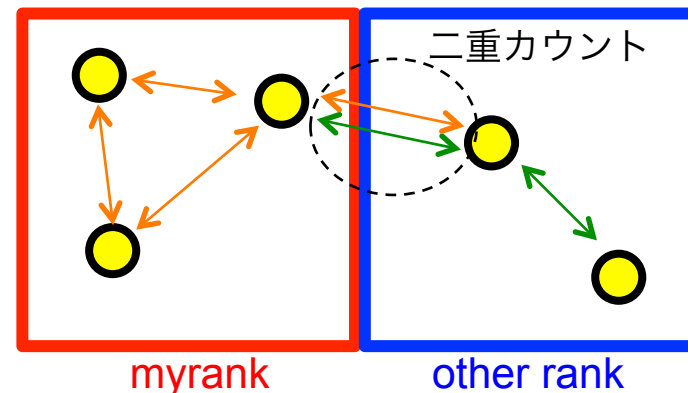
- ・ rank 間での二重カウント
特にポテンシャルエネルギーとビリアル

例) 運動エネルギー

$$K = \sum_{i=1}^N \frac{1}{2} m_i v_i^2 = \sum_{k=1}^{nprocs} \sum_{i=1}^{N_k} \frac{1}{2} m_i v_i^2$$

kene プロセスごとの和
wk_kene

```
call mpi_allreduce(wk_kene,kene,1,
  mpi_double_precision,mpi_sum,
  mpi_comm_world,ierr)
```





目次

- **分子動力学 (MD) 法**
- **分子動力学計算の並列化特性**
- **並列化技術 1 データ構造**

- **並列化技術 2 MPI**
- **並列化技術 3 OpenMP, SIMD**

第一回

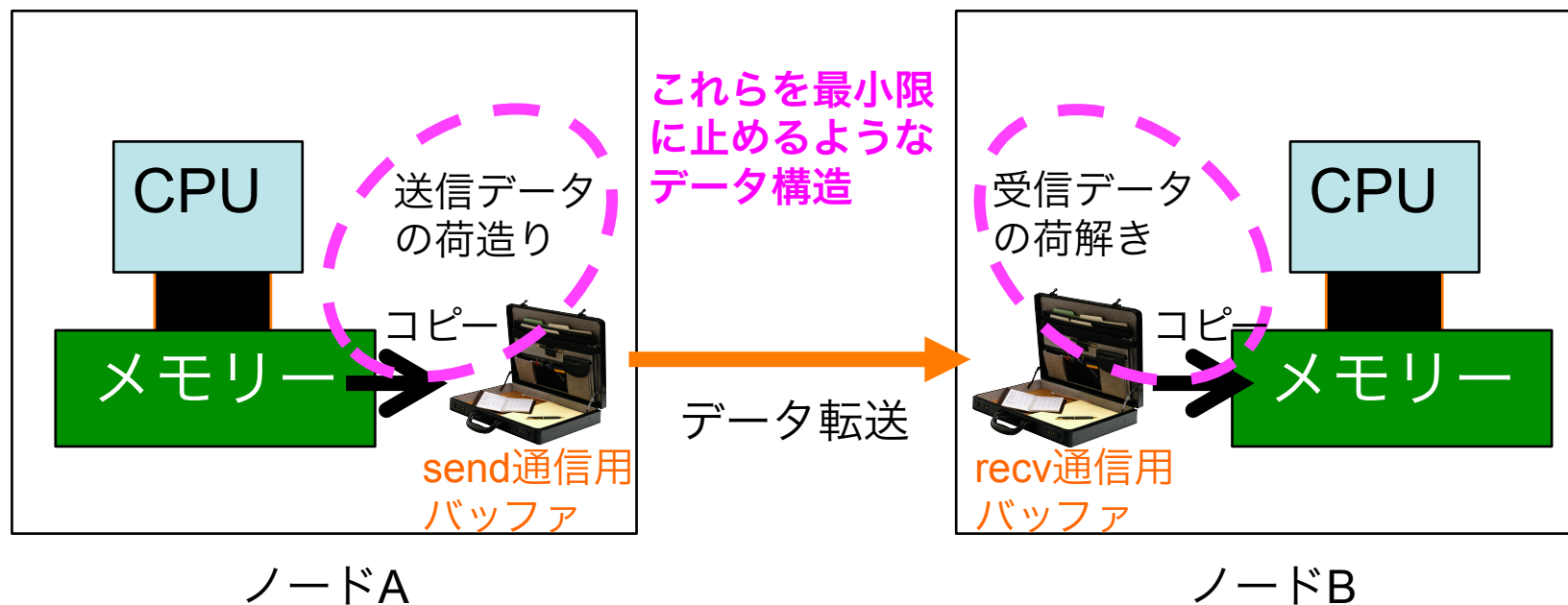
第二回

MPI 並列化技術: データ構造

データ構造: MPI 並列性能 (および演算性能) に直結

MD 計算のように 1 サイクル当たりの経過時間が小さい [$\leq O(10^{-3} \text{ sec})$] 場合に特に重要.

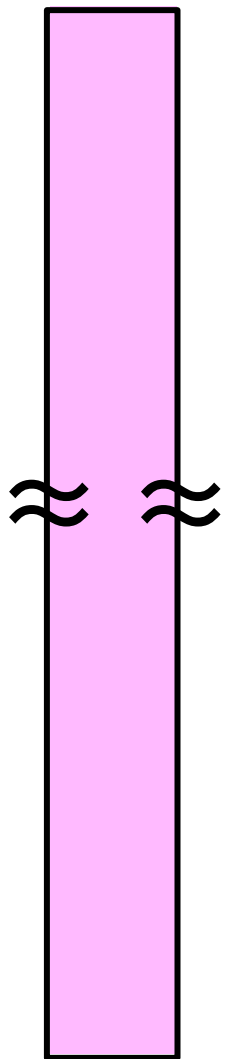
システムが大規模・複雑化する将来, アプリによらず重要になる (?).



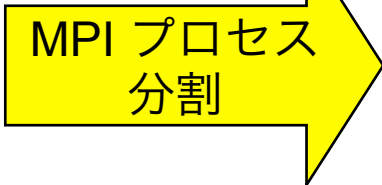
[理想] プログラムを書く前に, ネットワーク構造に最適なデータ構造を考案する
 [現実] 途中で問題に直面して, 既存のプログラムをイチから書き直す

オリジナル配列

allocate(x(N))



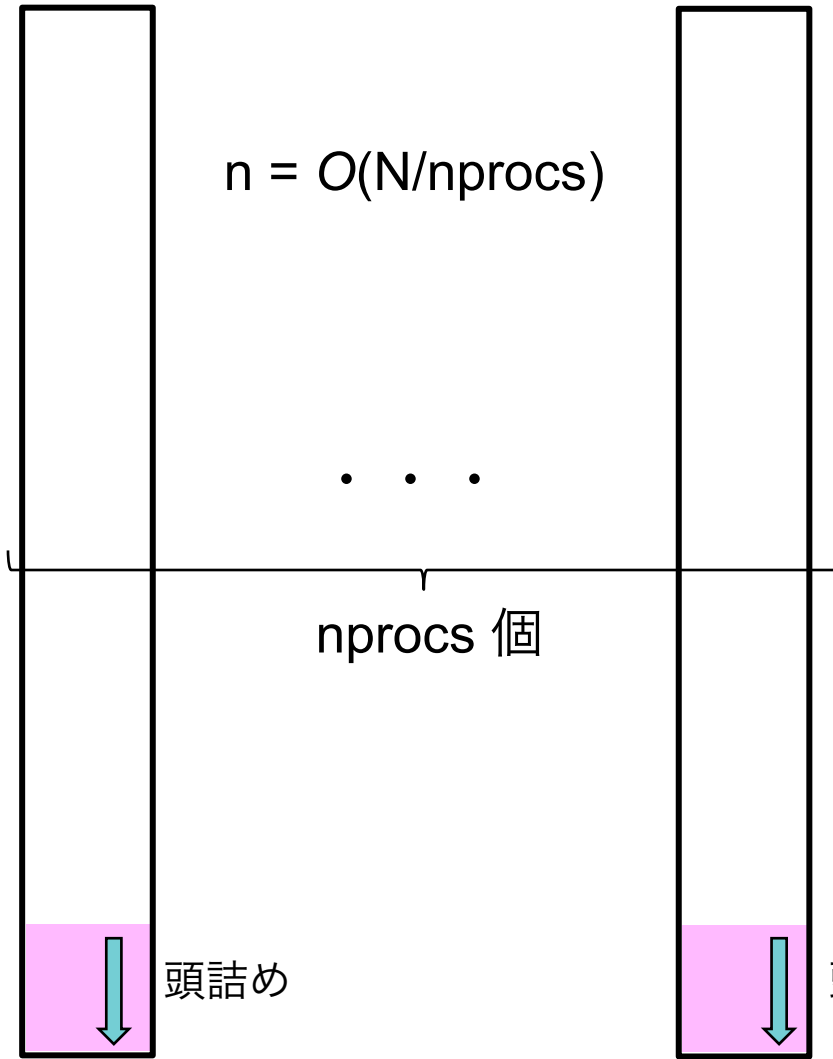
N: 原子数



プロセス分割した配列

allocate(wk_x(n))

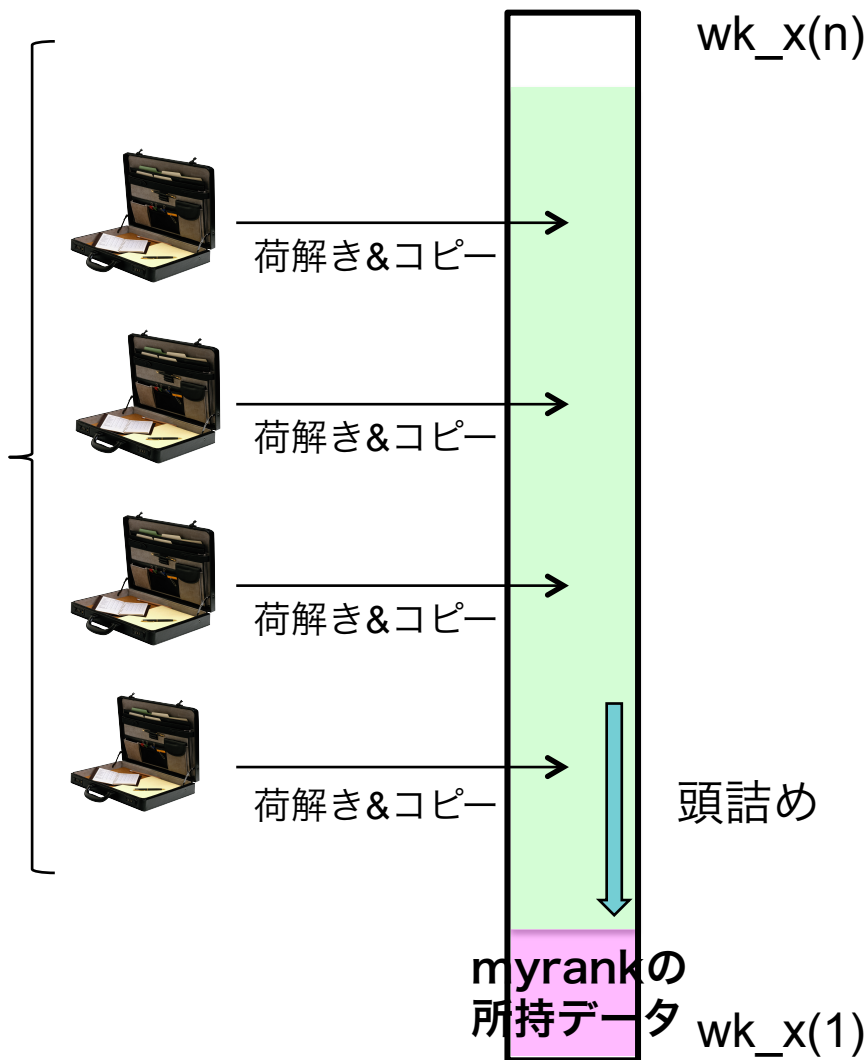
注) 実際には,
allocate(wk_xyz(3,n))





転送された座標データの格納形態

other ranksからの
受信データ





新しく考案した座標のデータ構造: **メタデータ構造**

メタデータ: あるデータのための
情報を持ったデータ

ここでは,
あるデータ: 原子座標
情報: 原子の所属するサブセルの
相対的な空間配置

myrankの
所持データ

meta_x(n')

$n' \sim n * \text{margin}$

allocate(meta_x(n'))

+

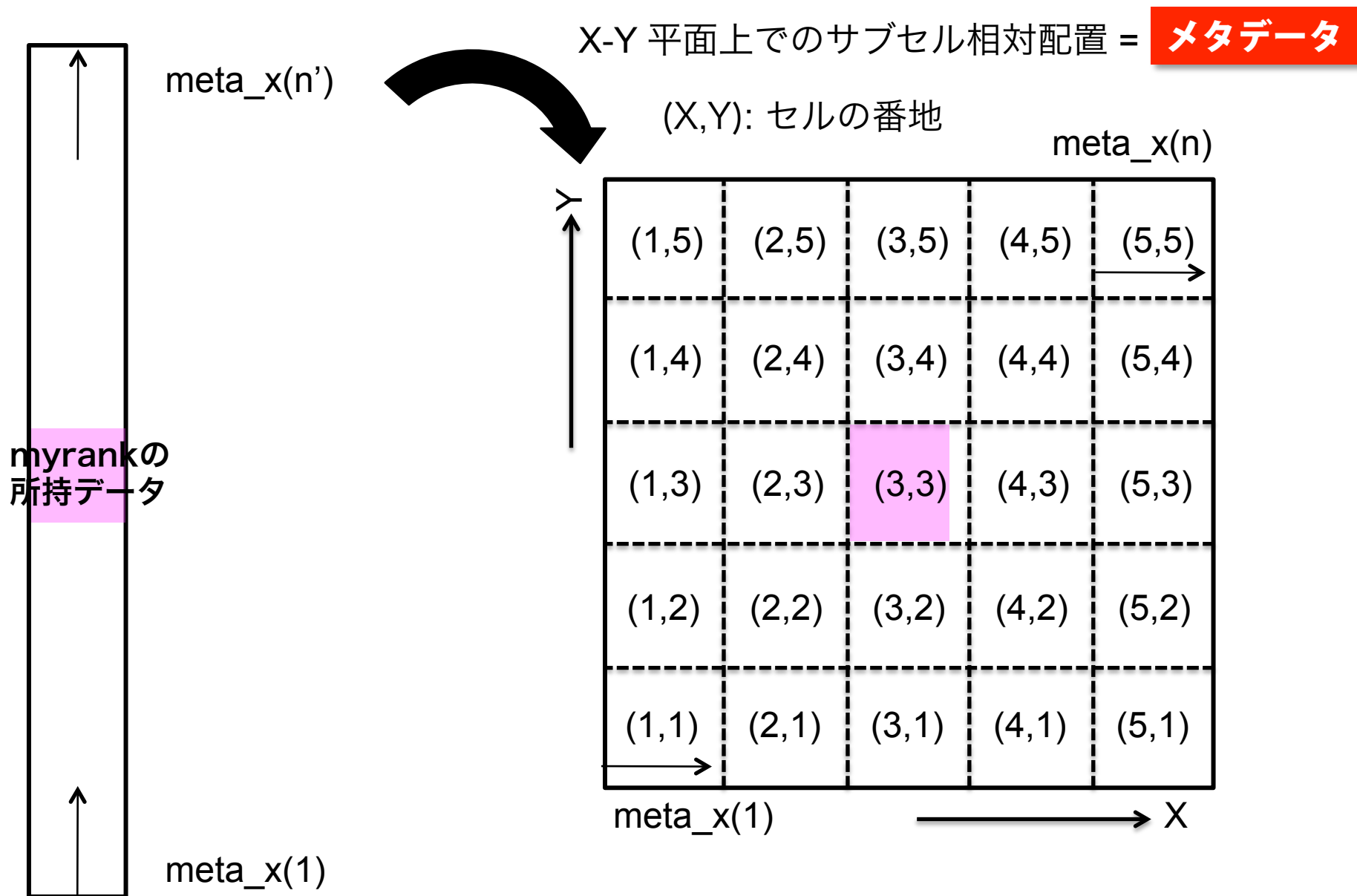
メタデータ管理配列

allocate(tag(n_{sbc} , n_{sbc}),
na_per_cell(n_{sbc} , n_{sbc}))

n_{sbc} : 軸あたりのサブセル数

注) 3次元では,
allocate(tag(n_{sbc} , n_{sbc} , n_{sbc}))

meta_x(1)



メタデータ管理配列

サブセルの先頭原子番号:

`allocate(tag(1:5,1:5))`

サブセルの格納原子数:

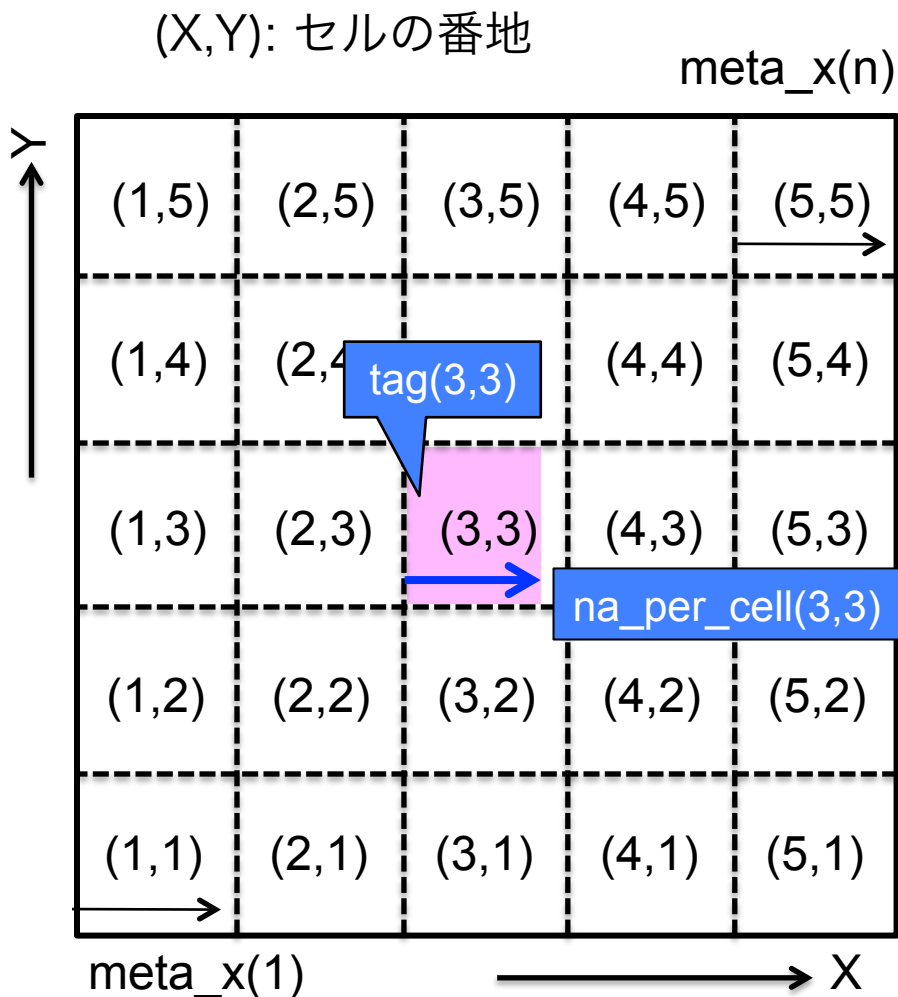
`allocate(na_per_cell(1:5,1:5))`

データへのアクセス方法

myrank所持データ:

`tag(3,3) ~`

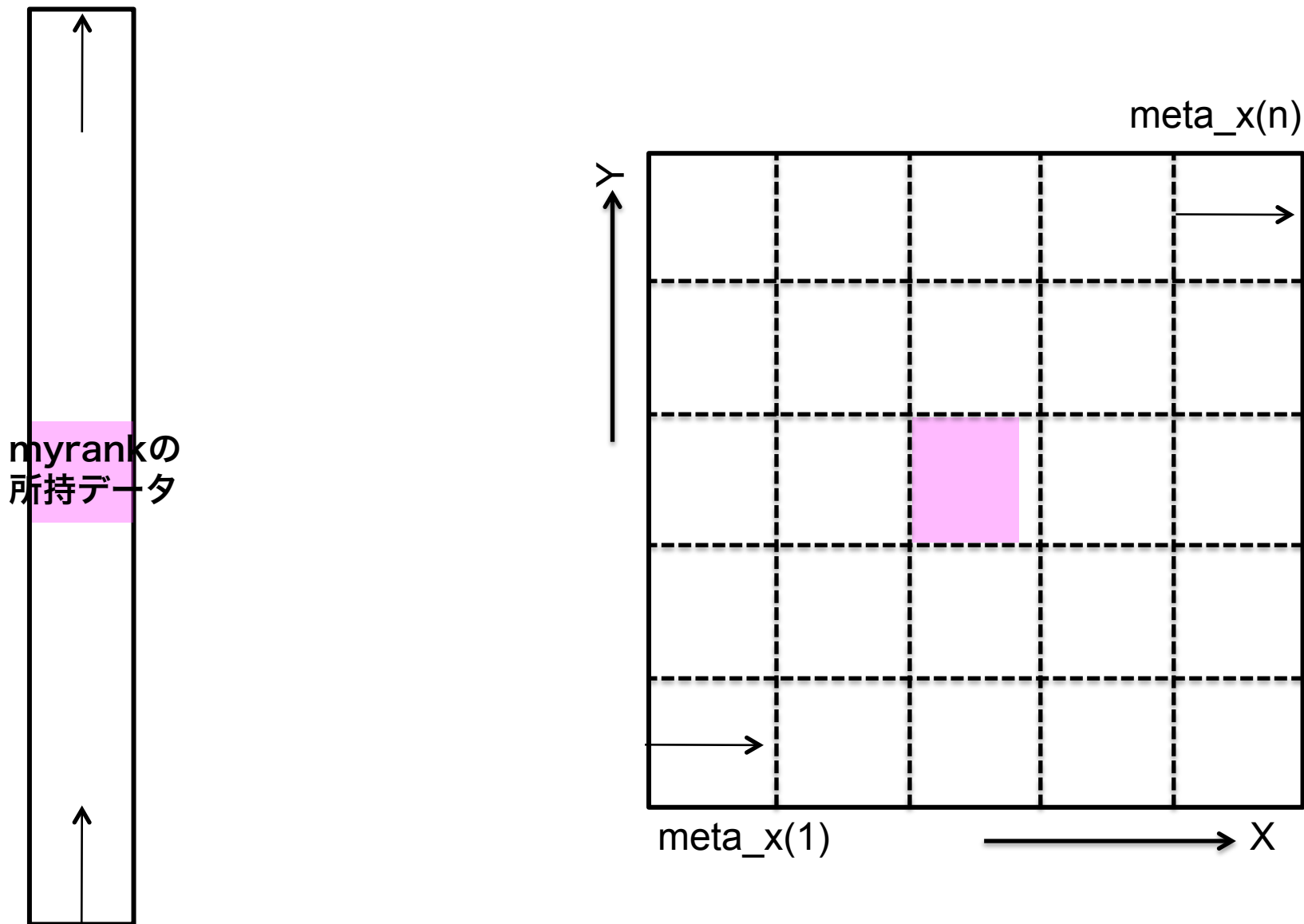
`tag(3,3)+na_per_cell(3,3)-1`





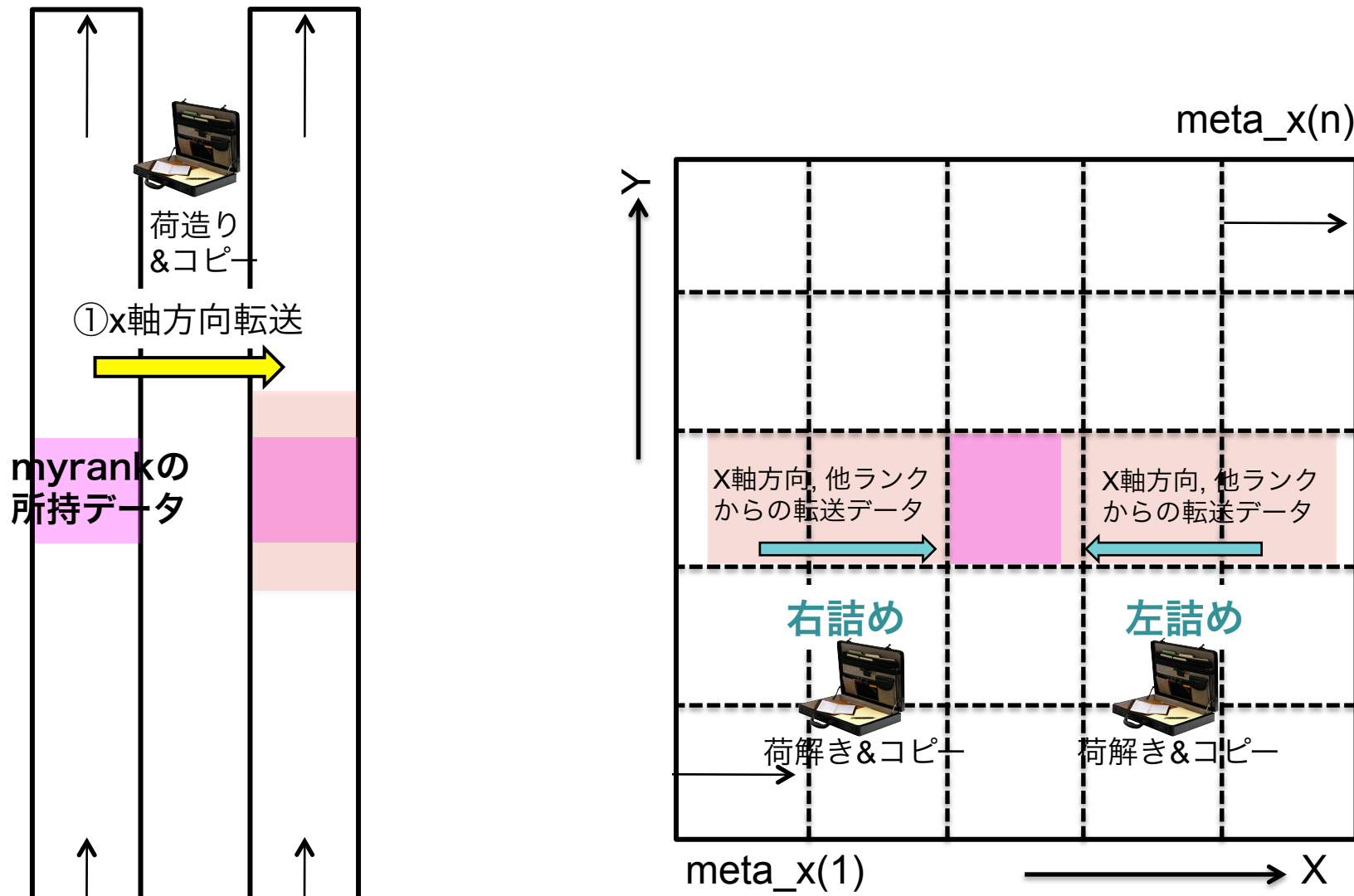
転送された座標データの格納形態

メタデータ構造



転送された座標データの格納形態

メタデータ構造



データへのアクセス方法

myrank所持データ:

$\text{tag}(3,3) \sim$
 $\text{tag}(3,3) + \text{na_per_cell}(3,3) - 1$

帯の部分のデータ:

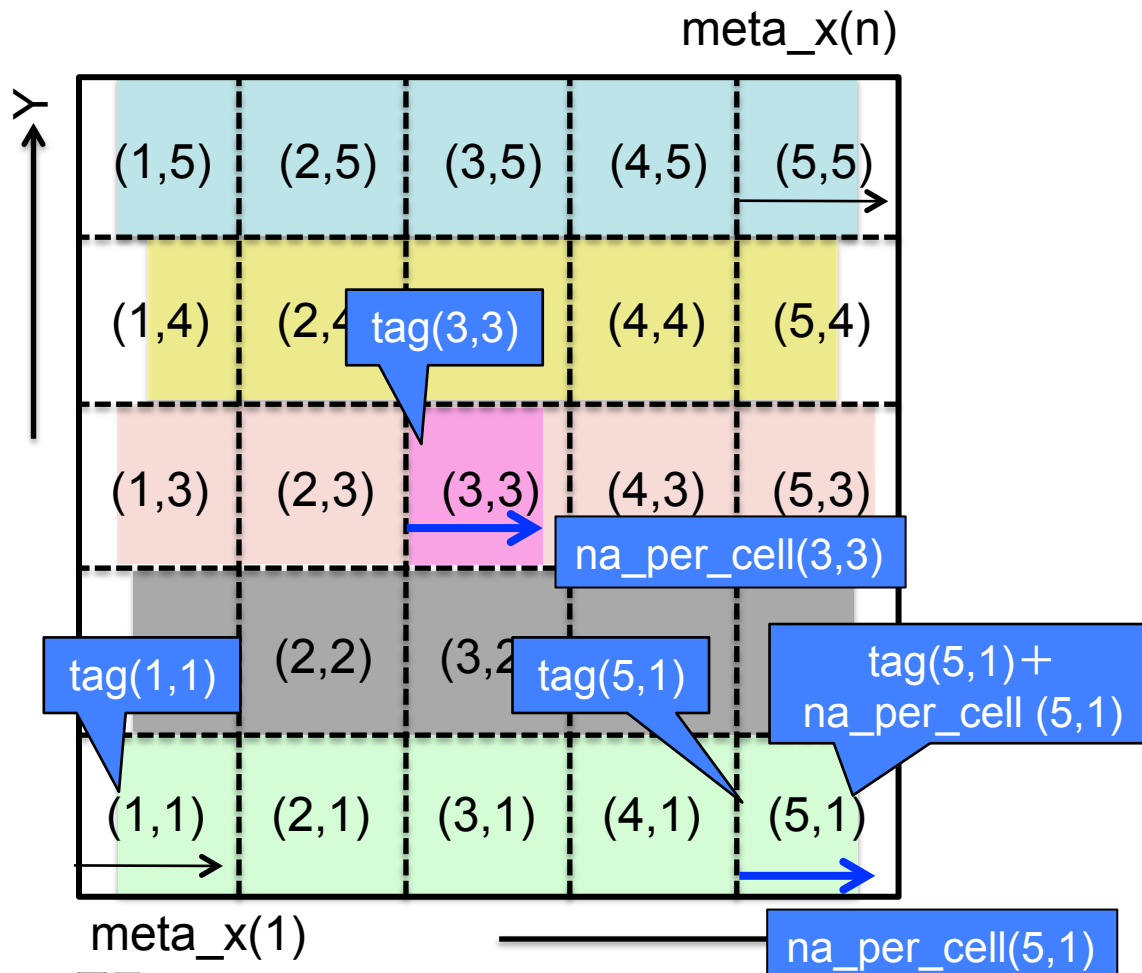
$\text{tag}(1,5) \sim$
 $\text{tag}(5,5) + \text{na_per_cell}(5,5) - 1$

$\text{tag}(1,4) \sim$
 $\text{tag}(5,4) + \text{na_per_cell}(5,4) - 1$

$\text{tag}(1,3) \sim$
 $\text{tag}(5,3) + \text{na_per_cell}(5,3) - 1$

$\text{tag}(1,2) \sim$
 $\text{tag}(5,2) + \text{na_per_cell}(5,2) - 1$

$\text{tag}(1,1) \sim$
 $\text{tag}(5,1) + \text{na_per_cell}(5,1) - 1$



連続アクセス

例) 40原子x5サブセル=200原子 (5KB)

→ L1キャッシュ(~32KB)に乗り, かつSIMD並列に十分な長さ



下位階層の多極子

下位の定義: 「プロセス分割数 < スーパーセル分割数」

$$8 < 16$$

```
allocate( wm0(:,16,16) )
```

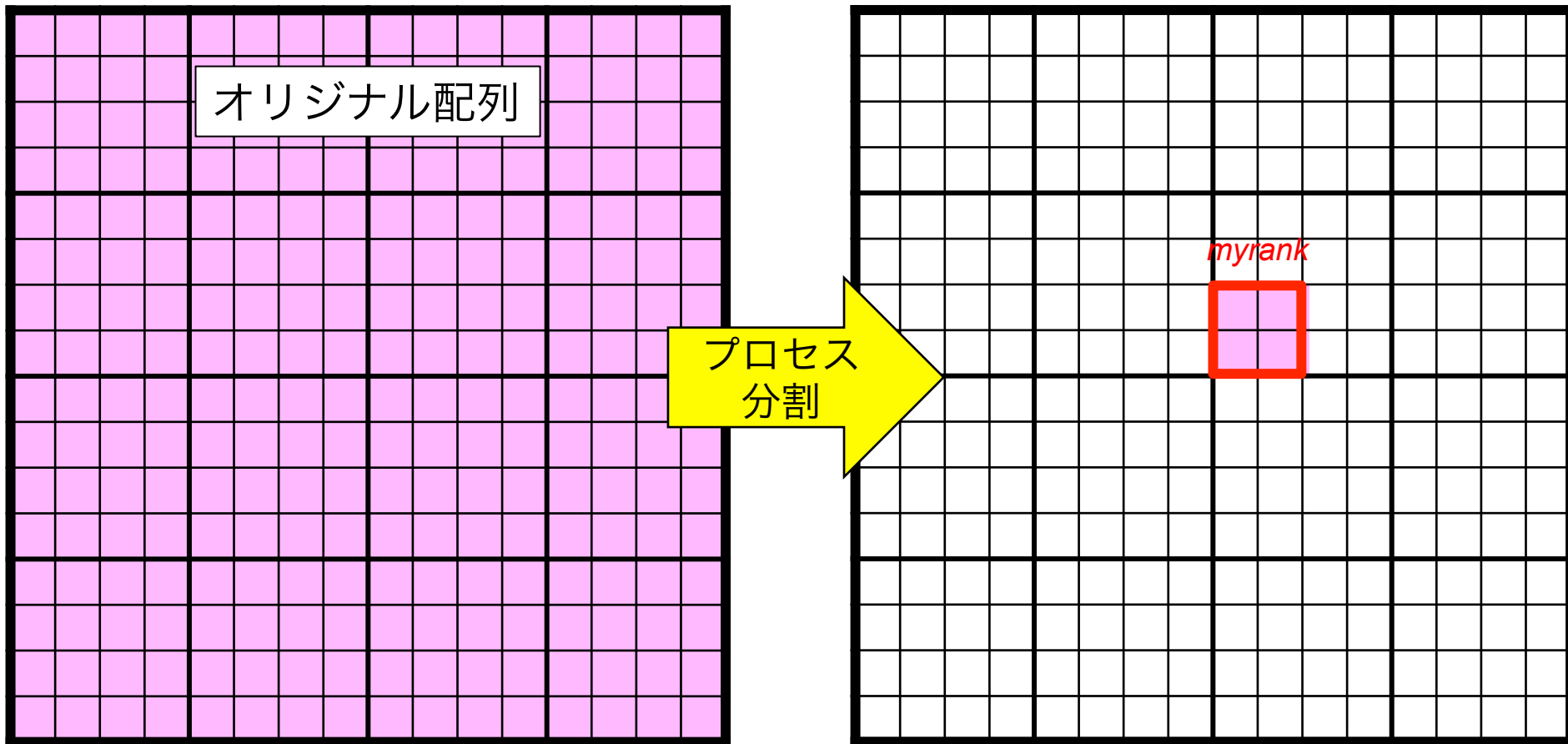
オリジナル配列

プロセス
分割

myrank

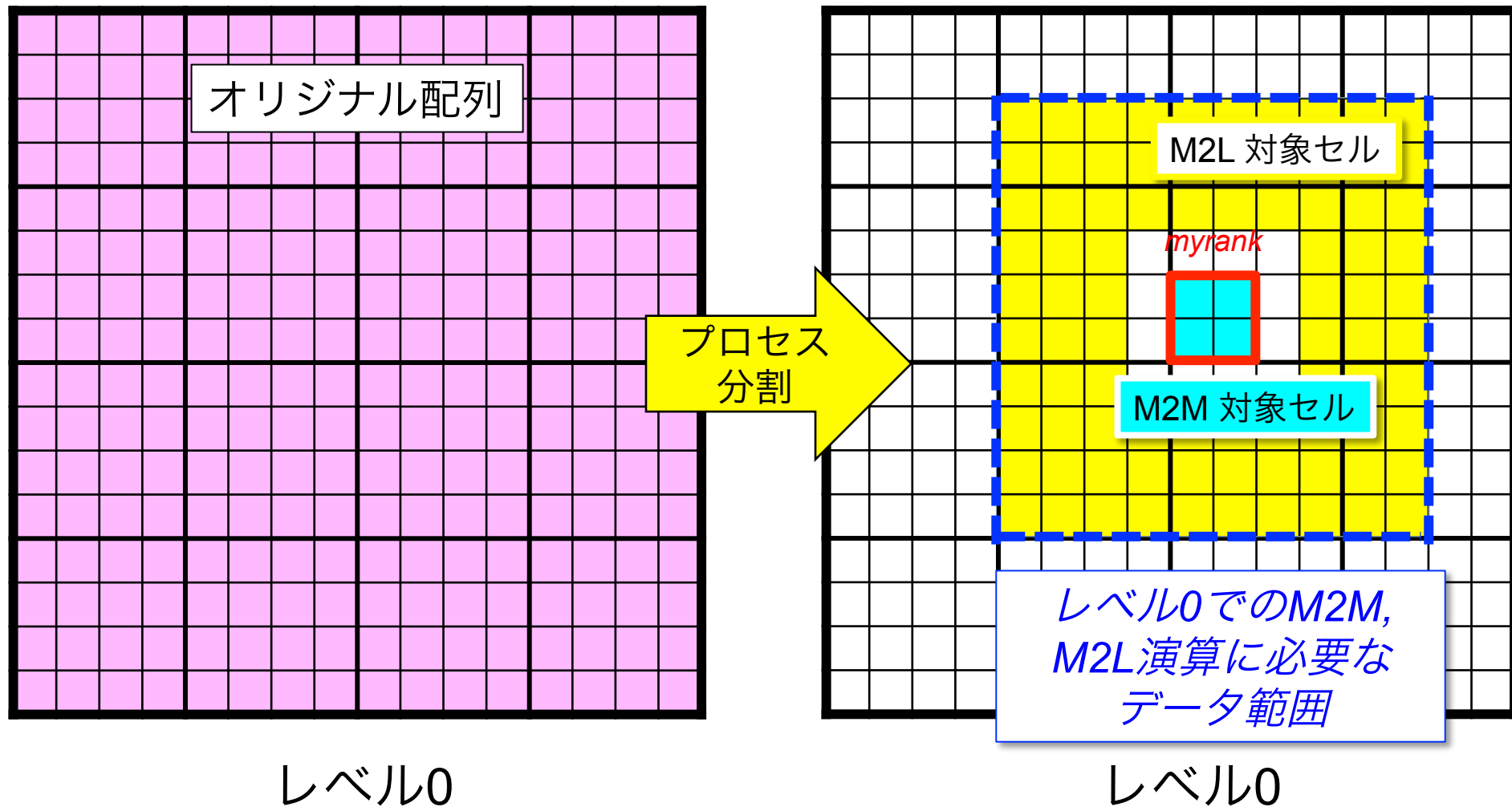
レベル0

レベル0



下位階層の多極子

```
allocate( wm0(:,16,16) )
```



下位階層の多極子

```
allocate( wm0(:,16,16) )
```

オリジナル配列

プロセス
分割

レベル0

配列を袖部付きで局所化

```
allocate( wm_local0(:,10,10) )
```

x方向: $10=4+2+4$

y方向: $10=4+2+4$

M2L対象セル

+4
myrank

-4

+4

M2M対象セル

-4

レベル0でのM2M,
M2L演算に必要な
データ範囲

レベル0



下位階層の多極子

データへのアクセス方法

myrank所持データ:

```
wm_local0(1:(nmax+1)2,5,5)
wm_local0(1:(nmax+1)2,6,5)
wm_local0(1:(nmax+1)2,5,6)
wm_local0(1:(nmax+1)2,6,6)
```

x,y軸方向転送

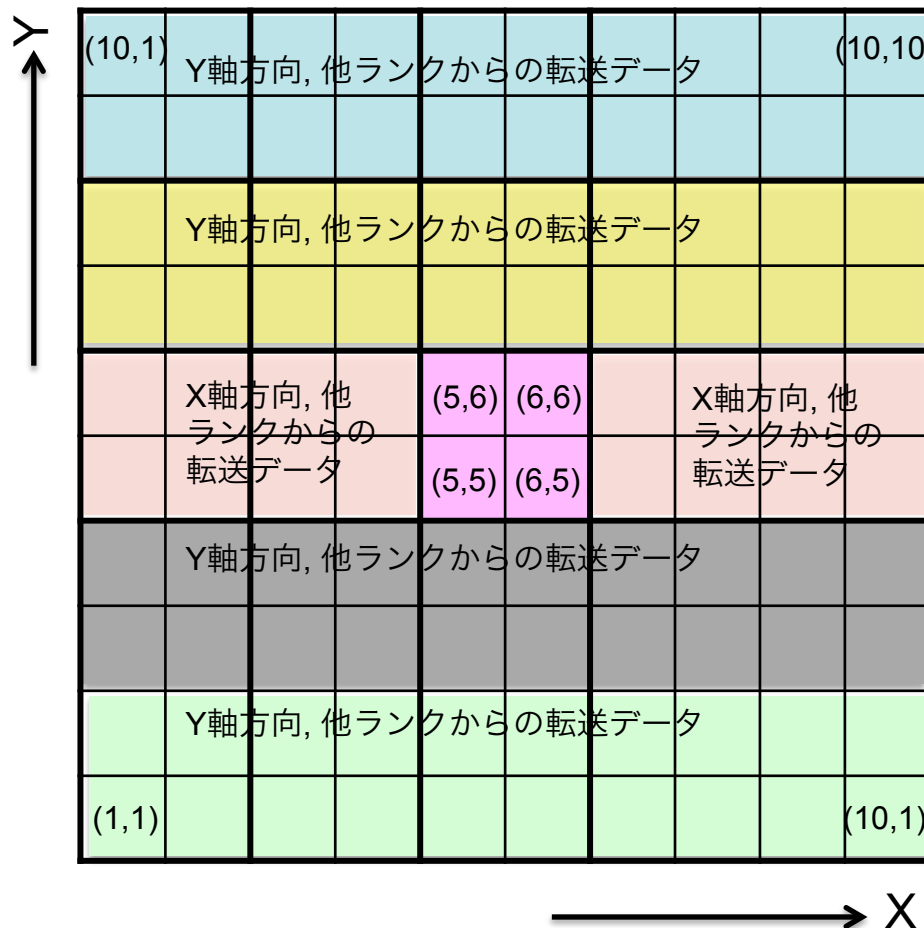


other ranks所持データ:

```
wm_local0(1:(nmax+1)2,1:10,9:10)
wm_local0(1:(nmax+1)2,1:10,7:8)
wm_local0(1:(nmax+1)2,1:10,5:6)
wm_local0(1:(nmax+1)2,1:10,3:4)
wm_local0(1:(nmax+1)2,1:10,1:2)
```

(X,Y): セルの番地

allocate(wm_local0(:,10,10))



全範囲に連続アクセス



上位階層の多極子

```
allocate( wm1(:,8,8) )
```

上位の定義: 「プロセス分割数 \geq スーパーセル分割数」

$$8 = 8$$

オリジナル配列

プロセス
分割

myrank

レベル1

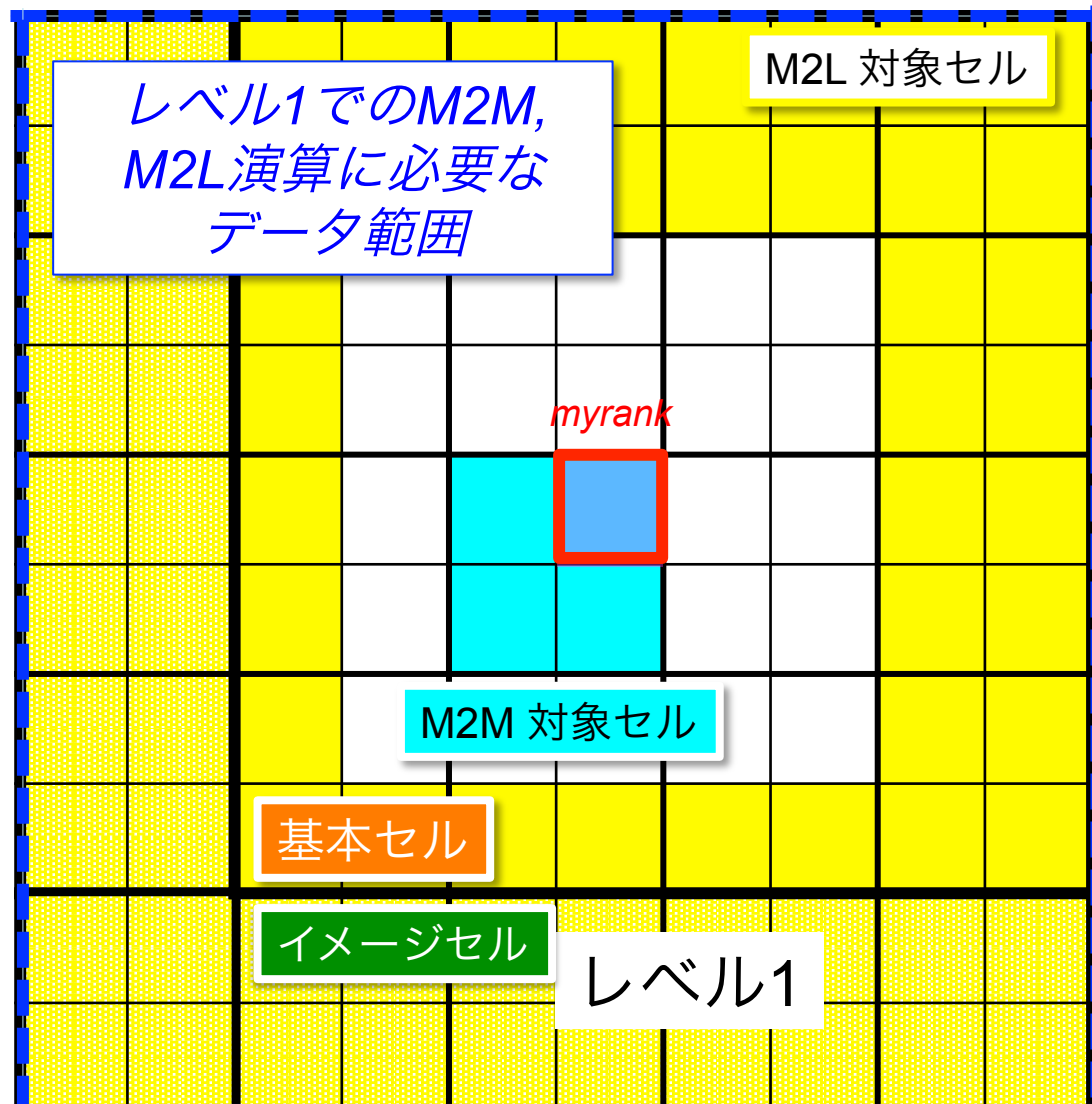
レベル1



上位階層の多極子

オリジナル配列

```
allocate( wm1(:,8,8) )
```





上位階層の多極子

オリジナル配列

```
allocate( wm1(:,8,8) )
```

配列を袖部付きで局所化

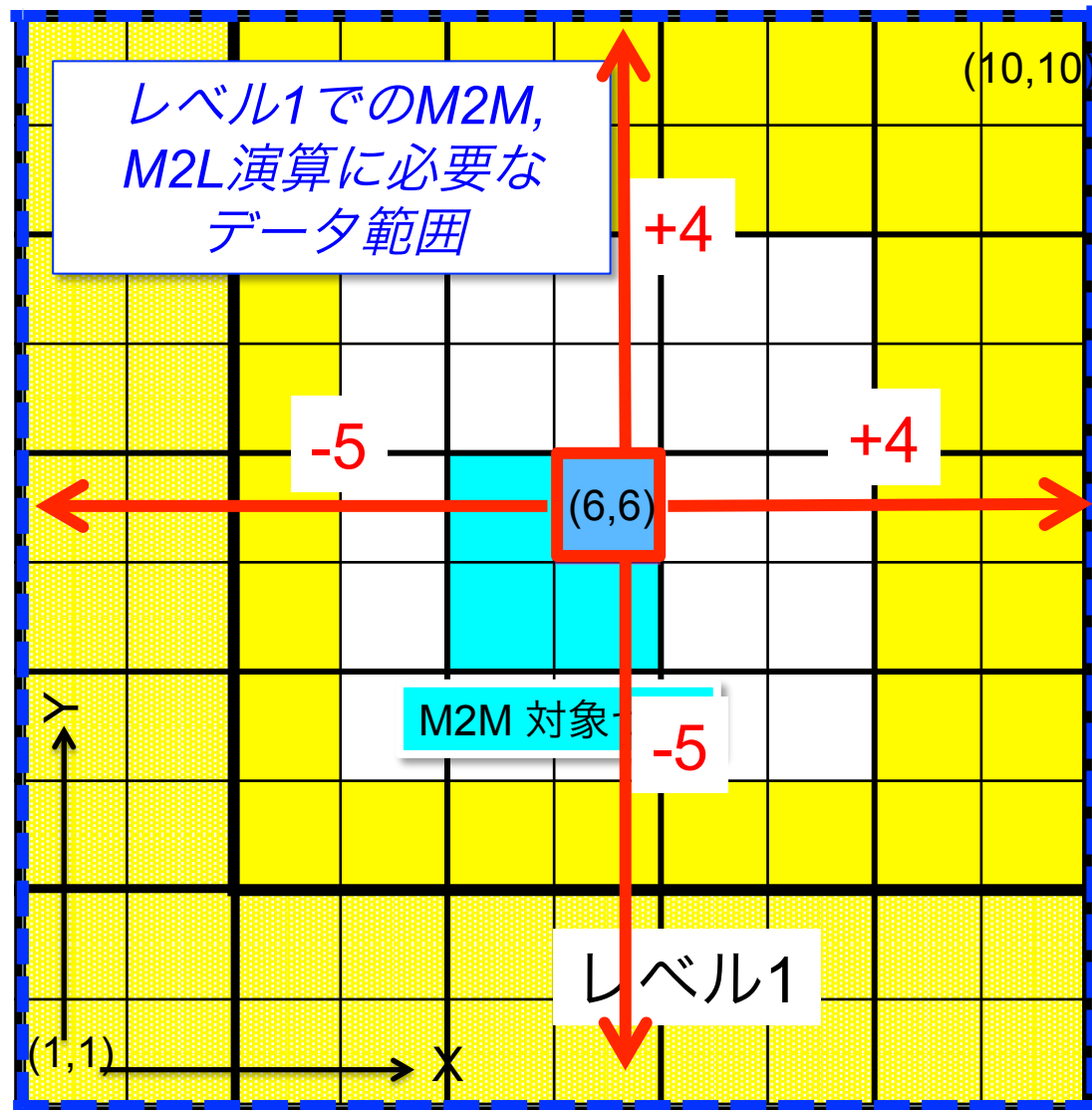
```
allocate( wm_local1(:,10,10) )
```

x方向: $10 = 5 + 1 + 4$

y方向: $10 = 5 + 1 + 4$

myrank所持データ:

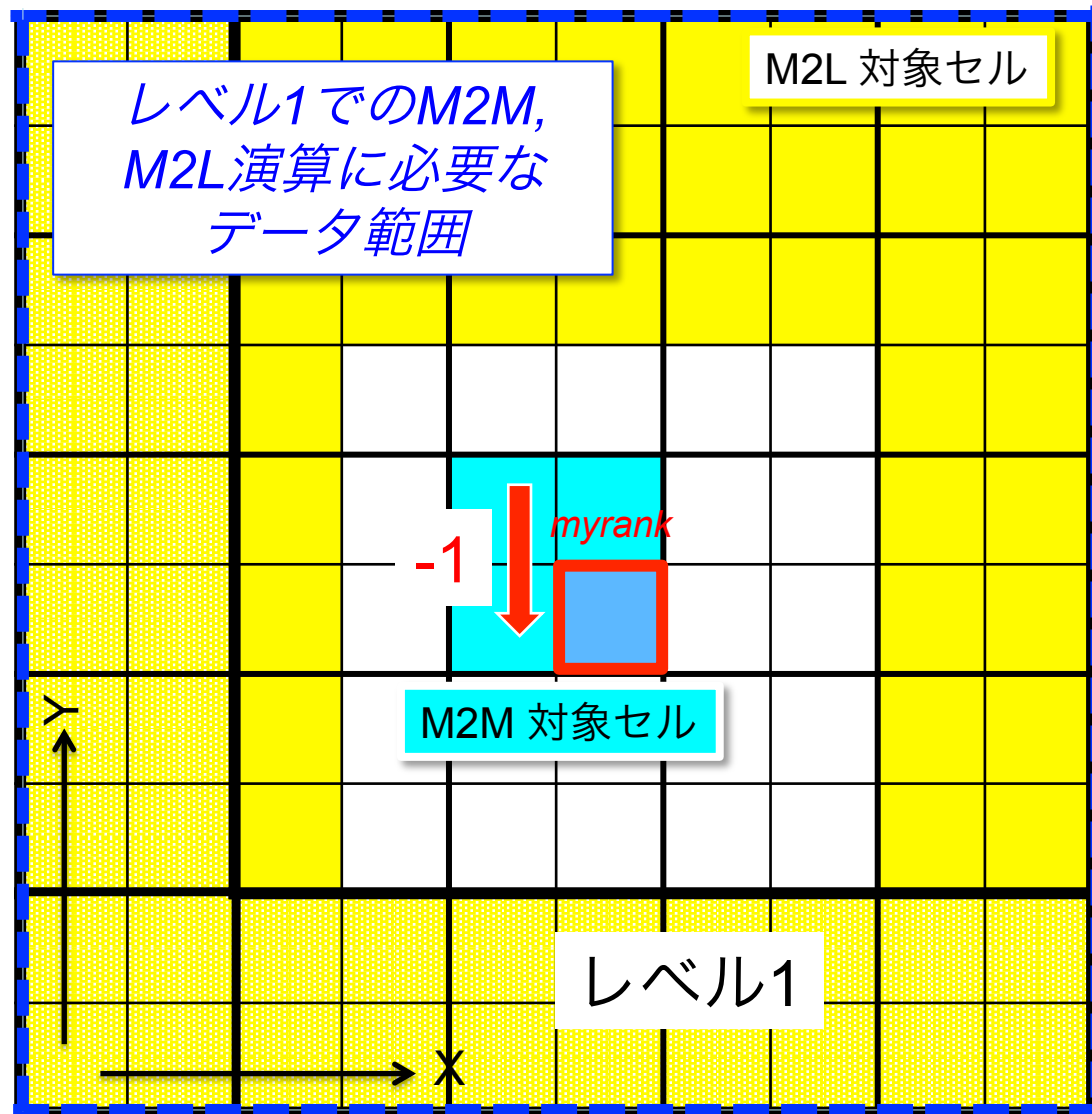
```
wm_local1(:,6,6)
```





上位階層の多極子

y 方向に -1 隣の
プロセスを考えると,





上位階層の多極子

オリジナル配列

```
allocate( wm1(:,8,8) )
```

配列を袖部付きで局所化

```
allocate( wm_local1(:,10,10) )
```

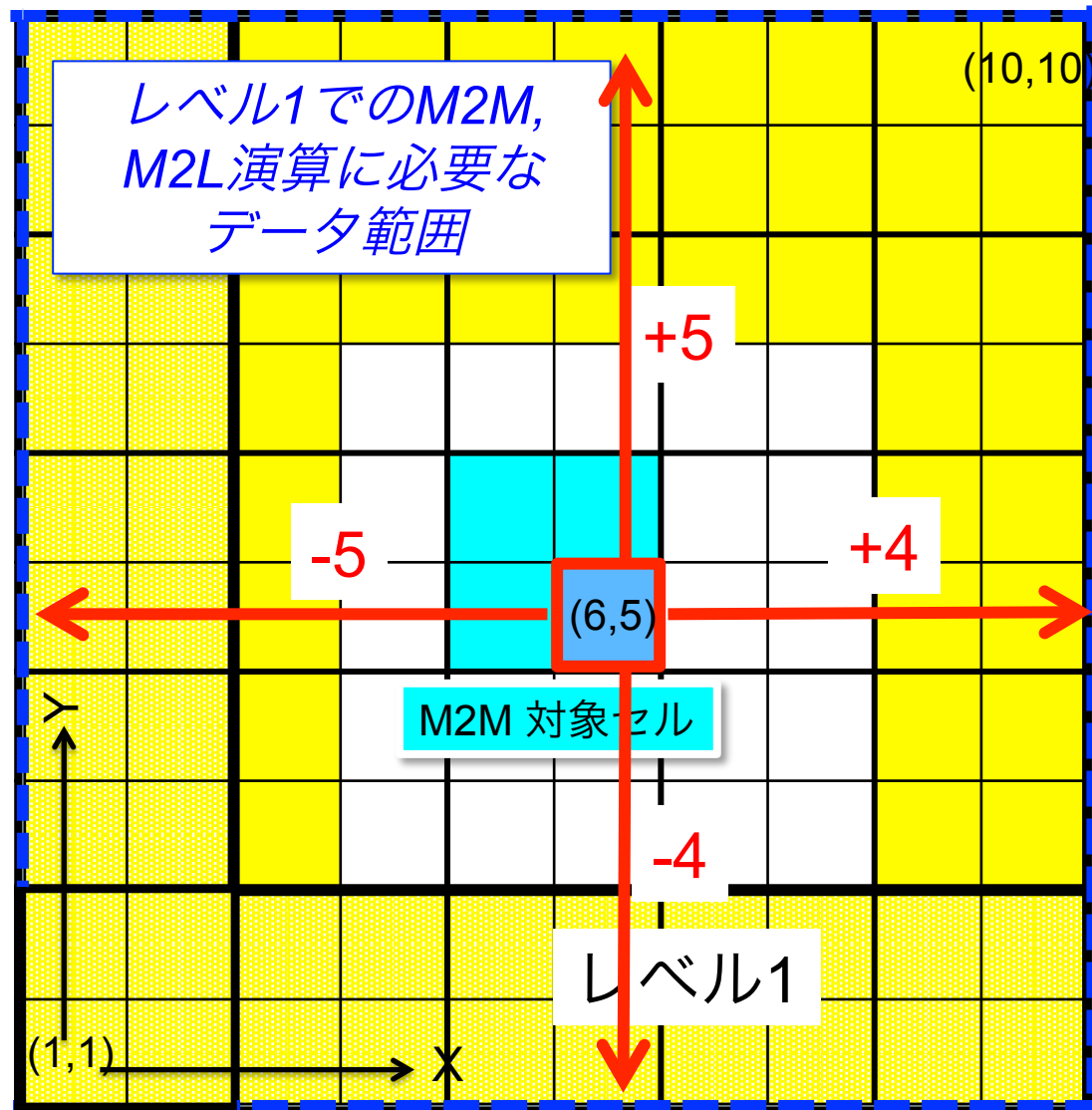
x方向: $10 = 5 + 1 + 4$

y方向: $10 = 4 + 1 + 5$

myrank所持データ:

```
wm_local1(:,6,5)
```

上位階層では, 各軸の +, - 袖部の値 (4 or 5) がプロセス位置に依存する点に注意





まとめ

- 分子内相互作用と分子間 nonbonded 相互作用の計算, および FMM を用いた Coulomb 相互作用の計算を含む分子動力学計算について, その並列化特性を解説した.
- MPI 並列性能および演算性能を向上させるデータ構造について, 座標および多極子 (下位, 上位階層べつ) に解説した.

次回は, このデータ構造を基にした 3 次元トーラスネットワーク上での MPI 並列化技術, および OpenMP, SIMD 並列化技術を具体的に説明する.