東京大学
THE UNIVERSITY OF TOKYO

拡張階層型領域間境界分割に基づく並列前処理手法

中島 研吾
東京大学情報基盤センター

2009年8月6日

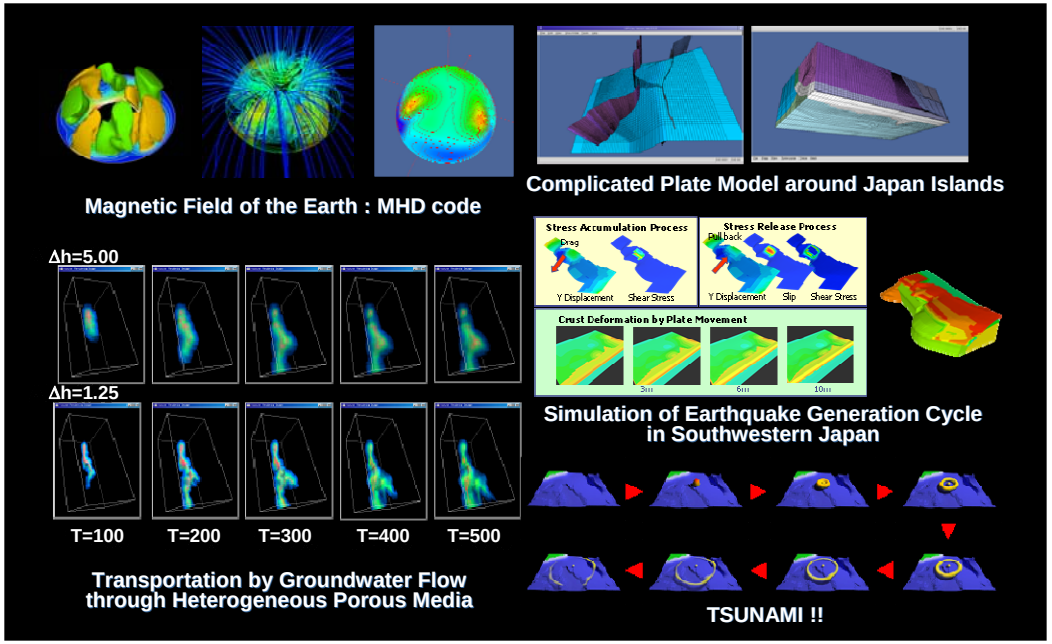
2009年並列／分散／協調処理に関する『仙台』サマー・ワークショップ（SWoPP仙台2009）
日本応用数理学会「行列・固有値問題の解法とその応用」研究部会（MEPA）

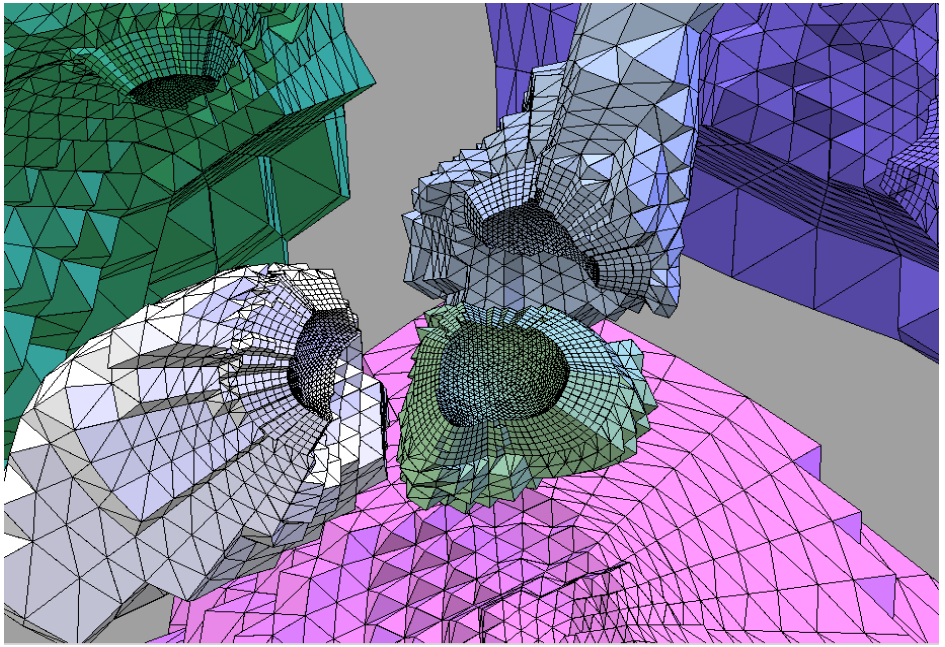
講演の概要

- 背景
  - ターゲットアプリケーション
  - T2Kオープンスパコン
  - Hybrid vs. Flat MPI
- ノード内並列化手法の実装
  - MC，CM-RCMリオーダリング
  - First Touch，再並び替え
- HIDの可能性
- HIDの改良
- まとめ

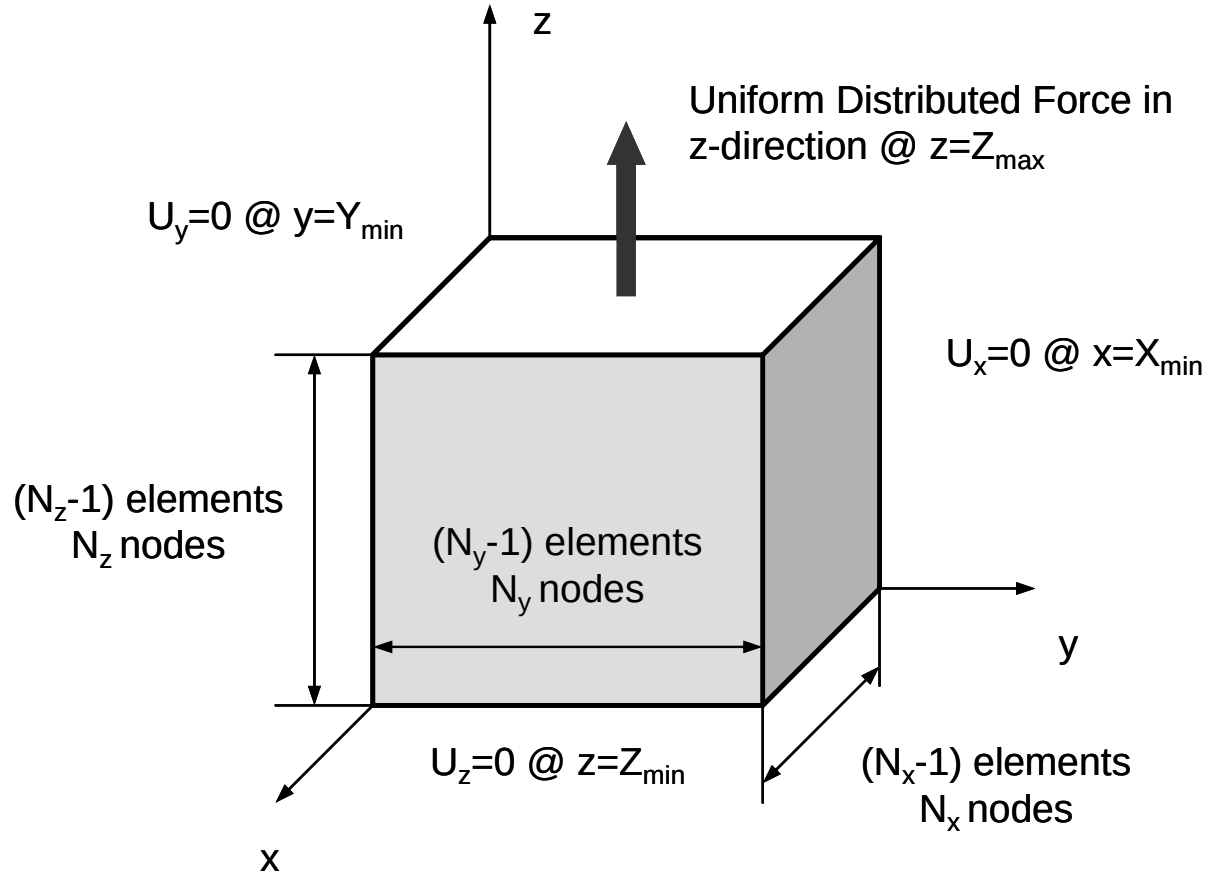
並列有限要素法による大規模シミュレーション

- 非構造格子
- 要素単位のローカルな処理⇒大規模疎行列
- 悪条件問題（ill-conditioned problems）
- 前処理付並列反復法



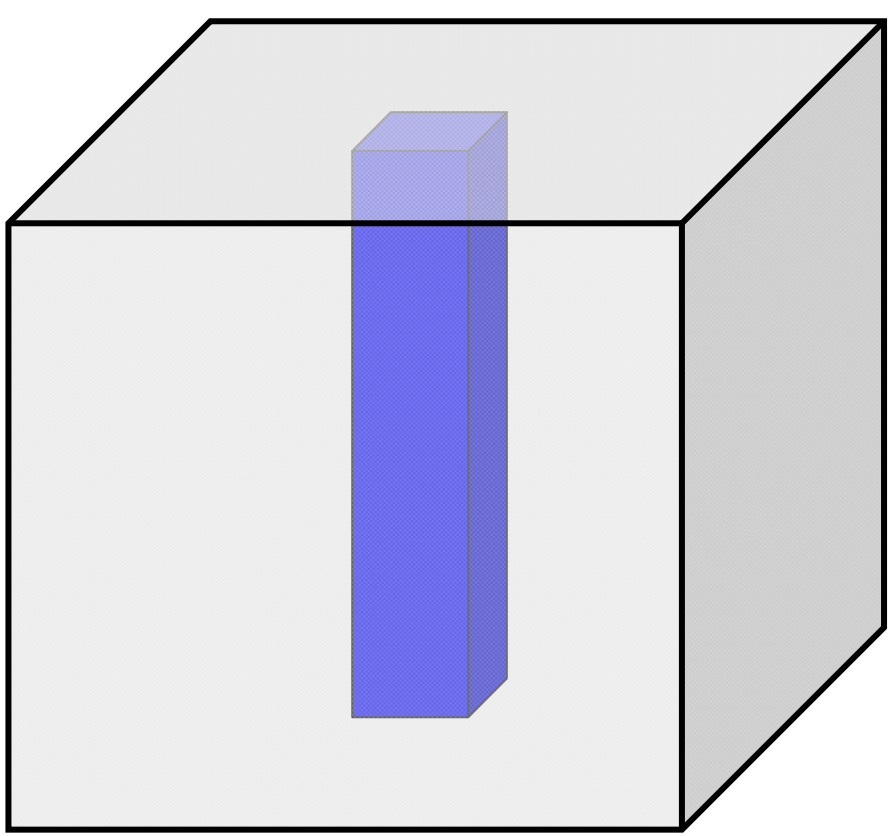


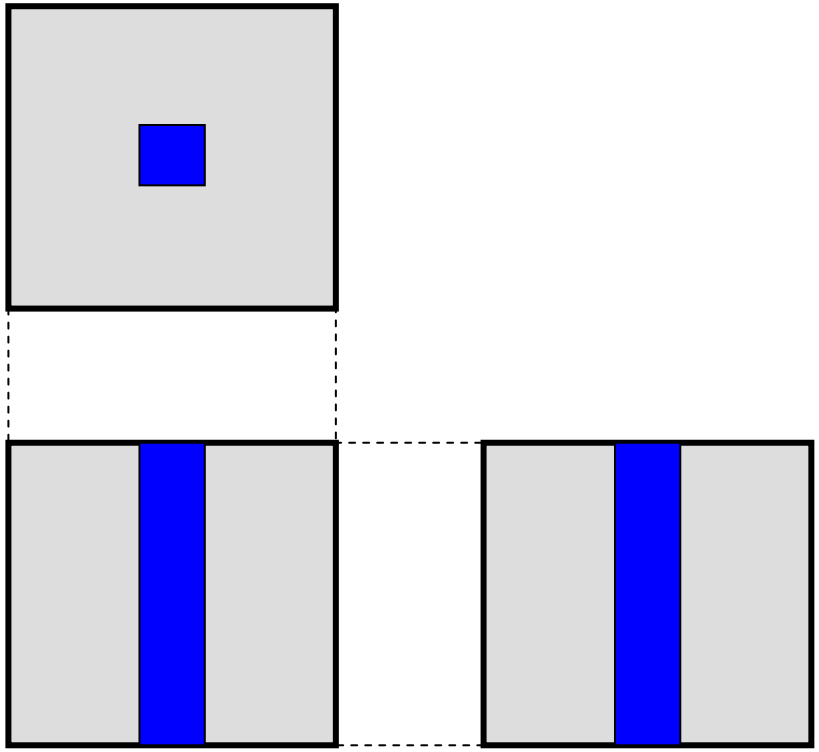
本講演のターゲット: 三次元弾性問題



不均質弾性問題：80^3要素（1.59M DOF）

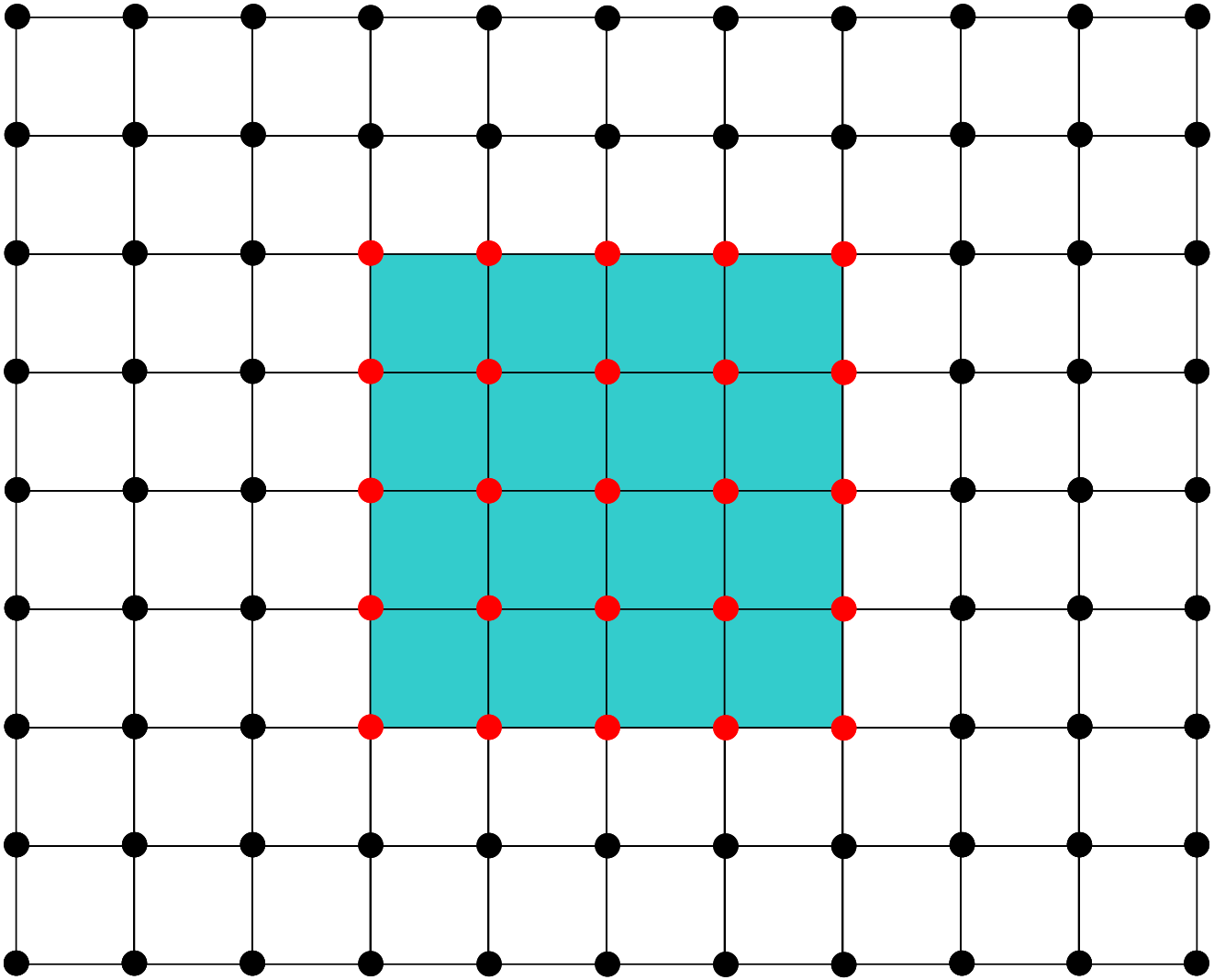
■：4×4×80





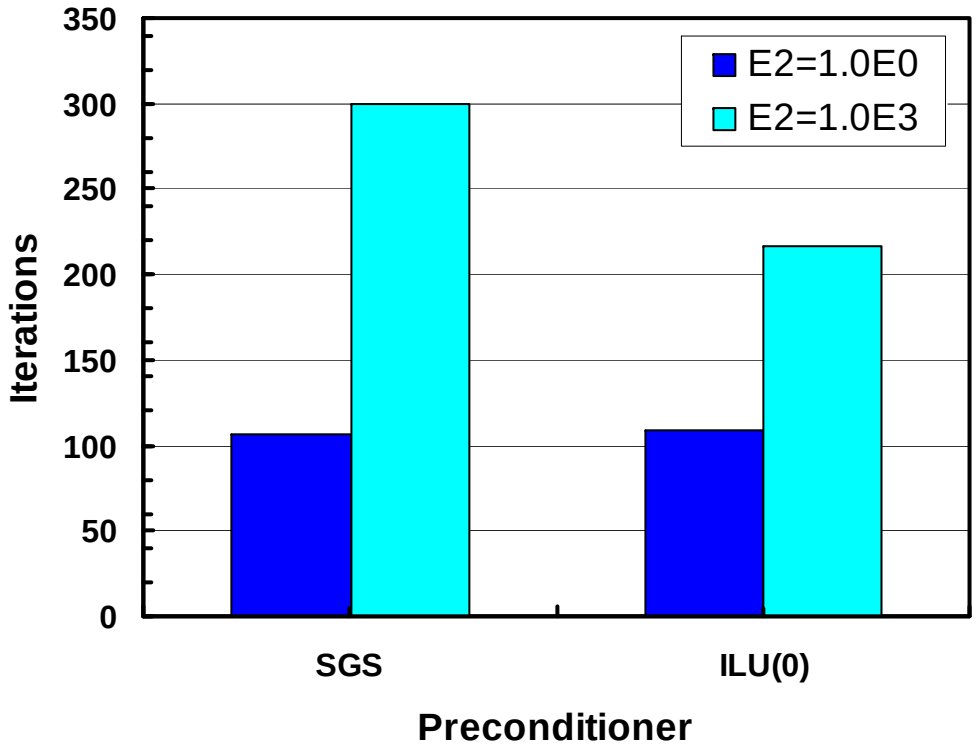
不均質弾性問題：80^3要素（1.59M DOF）

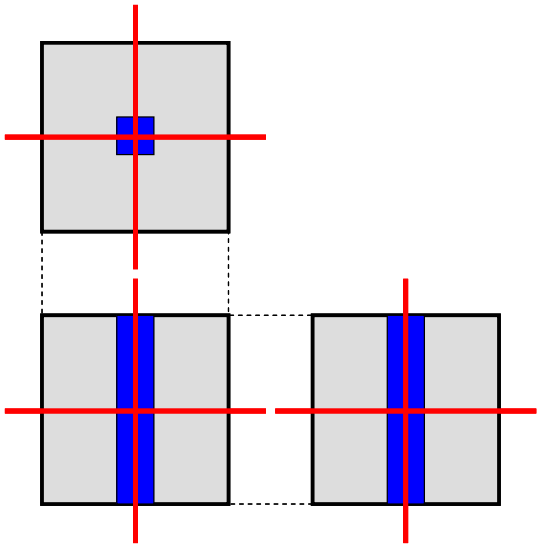
■：4×4×80

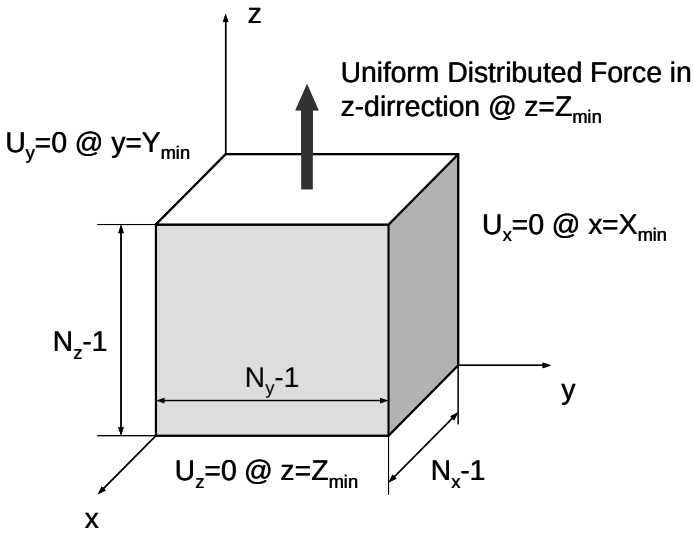


不均質弾性問題：40^3要素 前処理付きGPBiCG，反復回数

- : ν= 0.30
- : E_1=1.00
- : E_2=1.00 or 1,000



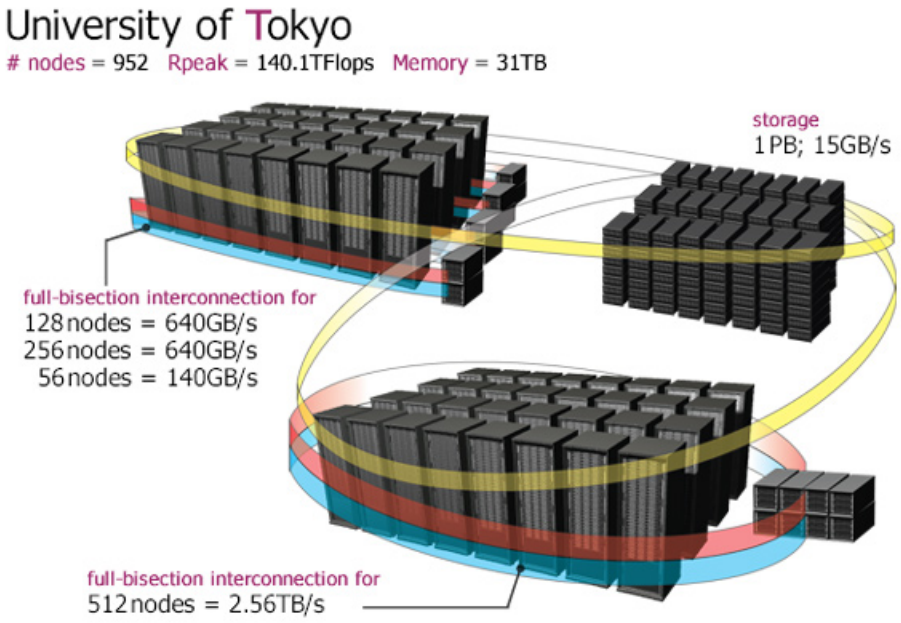




T2K（東大）（1/2）

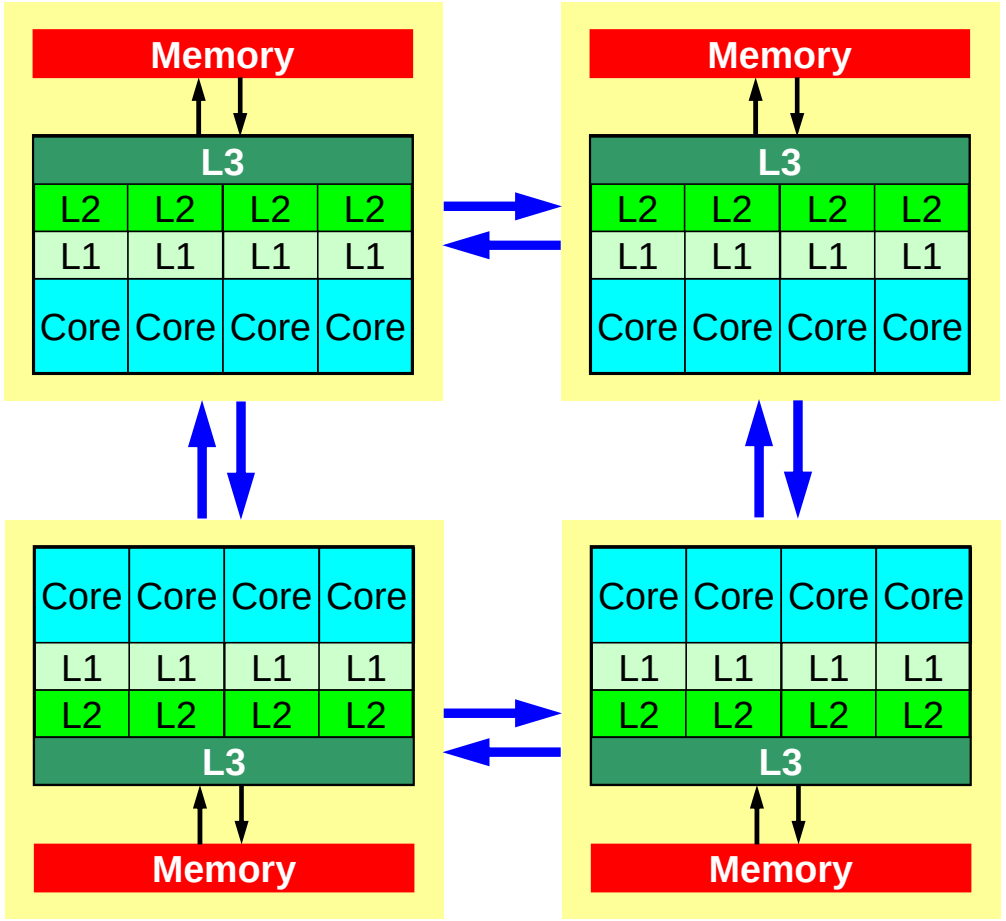
- T2Kオープンスパコン仕様
  - http://www.open-supercomputer.org/
  - 筑波大，東大，京大

- T2Kオープンスパコン（東大）
  - Hitachi HA8000クラスタシステム
  - 2008年6月～
  - 952ノード（15,232コア），
    141 TFLOPS peak
    - Quad-core Opteron (Barcelona)
  - TOP500 27位（NOV 2008）
    - （その時点で日本では1位だった）



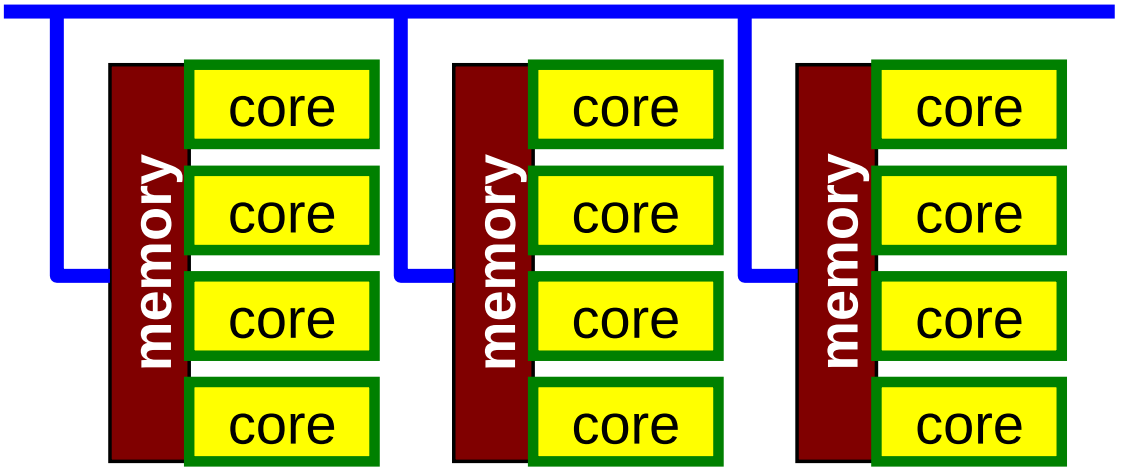
T2K（東大）（2/2）

- AMD Quad-core Opteron (Barcelona) 2.3GHz
  - 4 “sockets” per node
  - 16 cores/node
- マルチコア，マルチソケット
- cc-NUMA（cache coherent Non-Uniform Memory Access）
  - ローカルメモリ上のデータをできるだけ使用する
  - 陽的なコマンドラインスイッチ
  - NUMA control

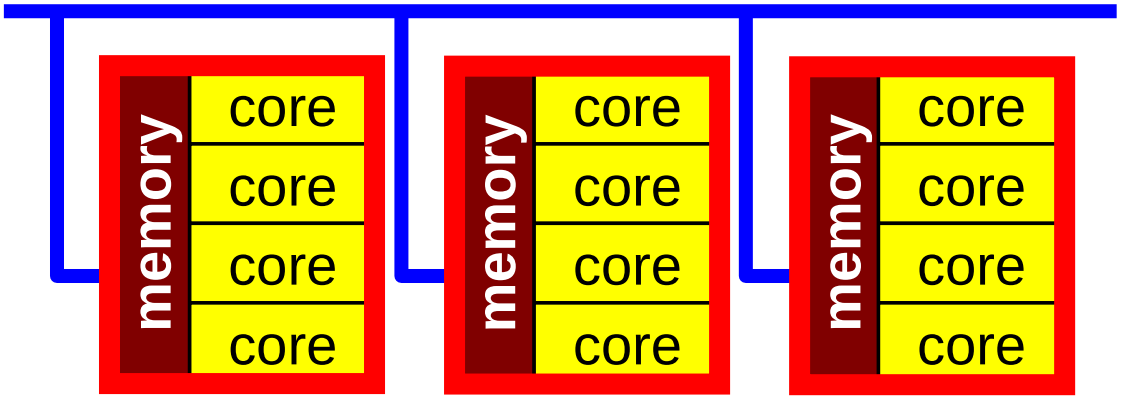


Flat MPI vs. Hybrid

Flat-MPI：Each PE -> Independent



Hybrid：Hierarchal Structure



Flat MPI vs. Hybrid

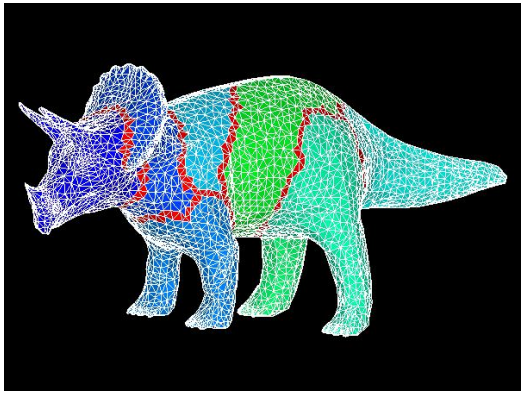
- 性能は様々なパラメータの組み合わせによって決まる

- ハードウェア
  - コア，CPUのアーキテクチュア
  - ピーク性能
  - メモリ性能（バンド幅，レイテンシ）
  - 通信性能（バンド幅，レイテンシ）
  - それらのバランス
- アプリケーション
  - 特性：memory bound，communication bound
  - 問題サイズ

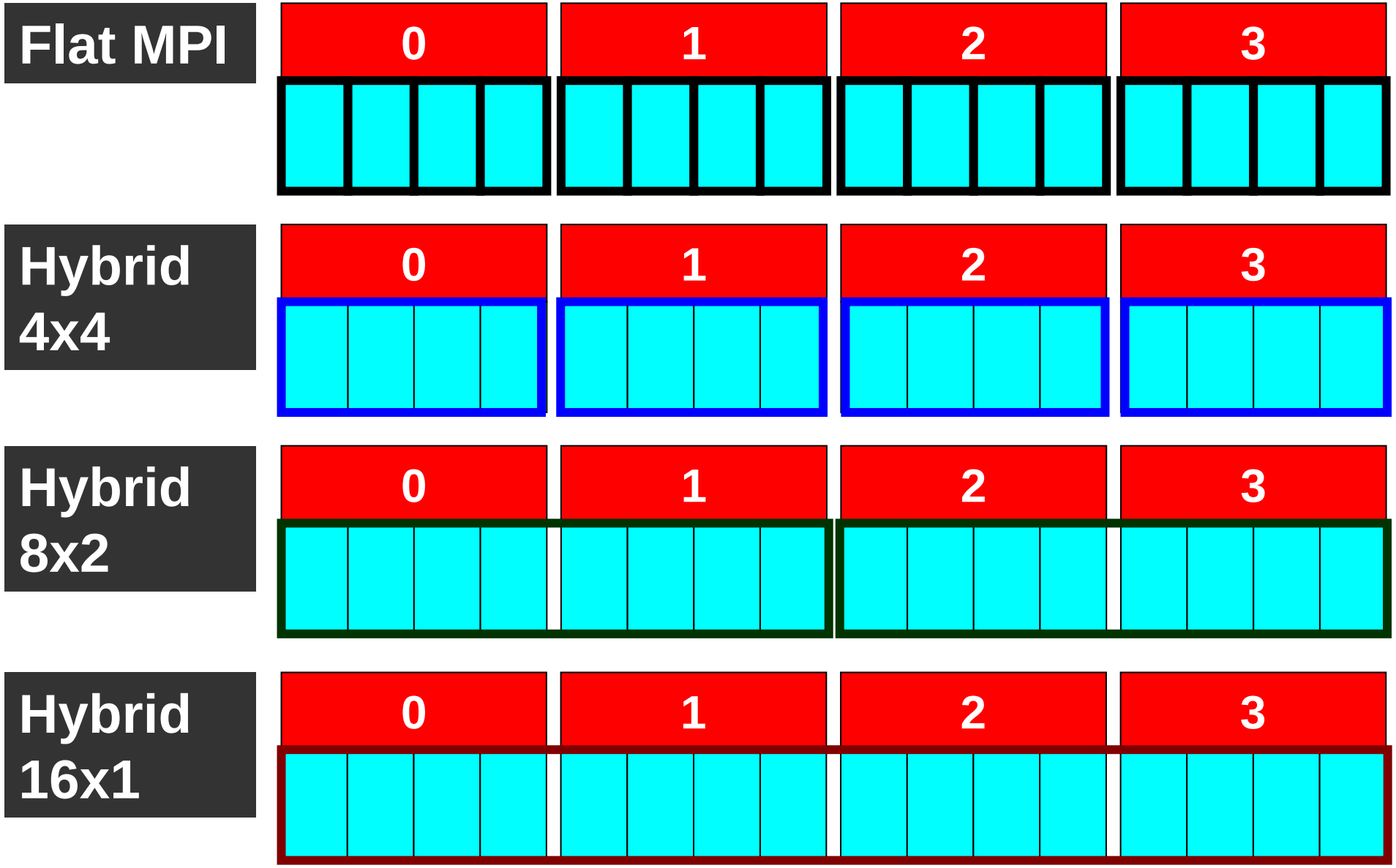
疎行列ソルバー：FEM，FDM

```
for (i=0; i<N; i++) {
  for (k=Index(i-1); k<Index(i); k++{
    Y[i]= Y[i] + A [k]*X[Item[k]];
  }
}
```

- Memory-Bound（メモリに負担）
  - 間接参照
  - Hybrid (OpenMP) は更に memory-bound
- Latency-Bound（並列計算時）
  - 通信は領域境界のみで発生
  - メッセージサイズも小さい
- エクサスケールシステム
  - コア数：O(10^8)
  - 10^8-way MPI：MPI Latencyによるオーバーヘッドは？
  - **Hybridへの期待**
    - **とりあえず，MPIプロセスの数を減らせる（T2Kの場合で1/16）**

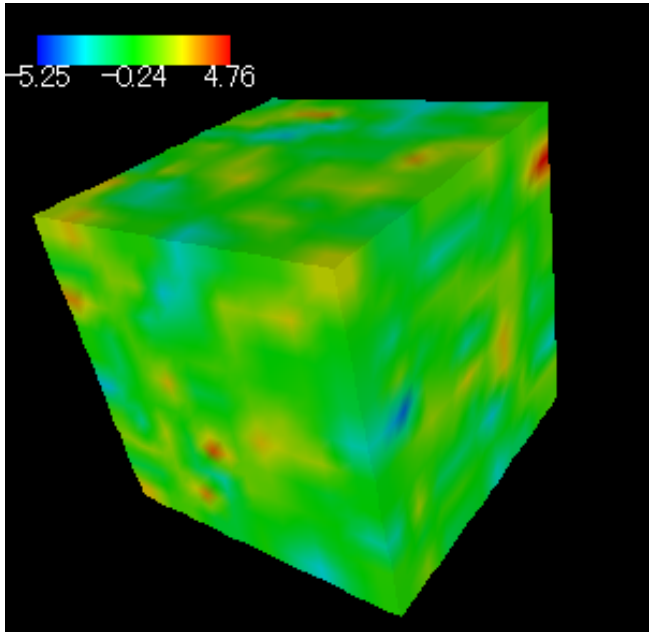


Flat MPI, Hybrid (4x4, 8x2, 16x1)



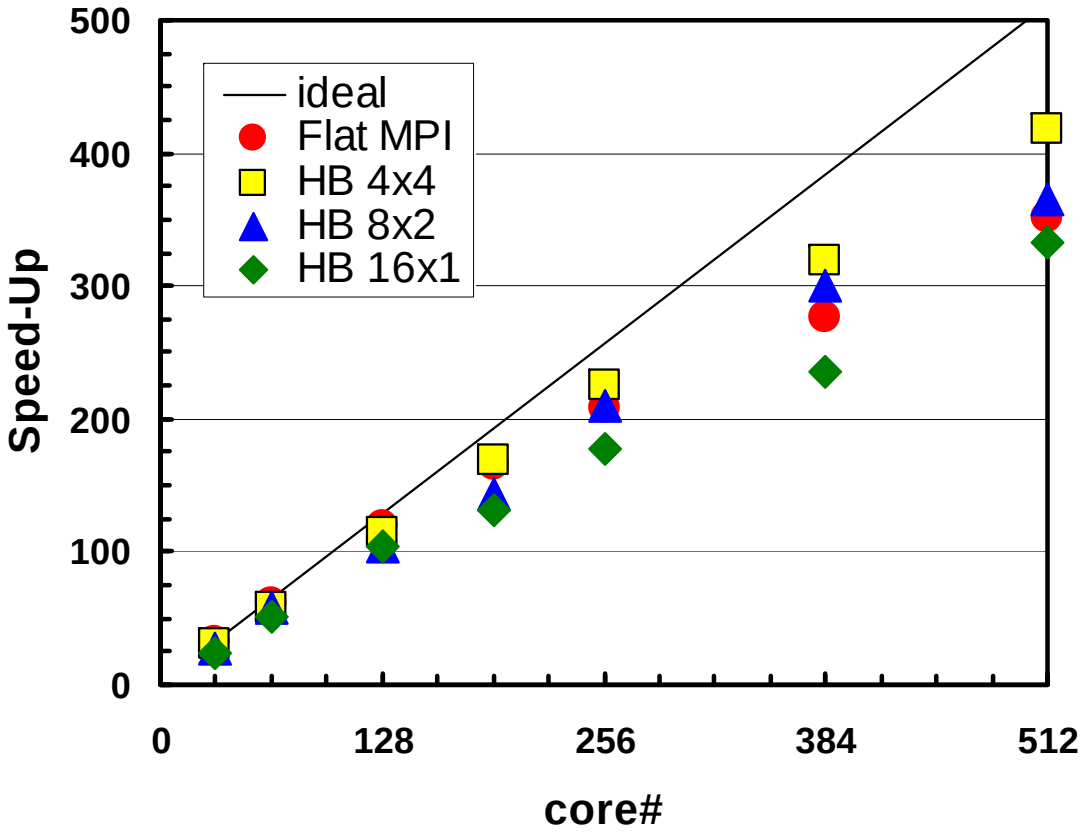
対象とした問題

- 三次元弾性静解析（固体力学），不均質物性
  - $E_{max}=10^{3}$, $E_{min}=10^{-3}$, $\nu=0.25$
    - 地質統計学的手法によって発生 [Deutsch & Journel, 1998]
  - 128^{3} 個の六面体要素, 6,291,456 DOF
    - Strong Scaling
- 反復解法：(SGS+CG)
  - Symmetric Gauss-Seidel
  - HID
- T2K（東大）
  - 512 cores (32 nodes)
- FORTARN90 (Hitachi) + MPI
  - Flat MPI, Hybrid (4x4, 8x2, 16x1)



Speed-Up: 32~512 cores

Flat MPI（32コア）の性能を32とした場合
SGS/GPBiCG不均質問題，Strong Scaling



Hybrid vs. Flat MPI on T2K

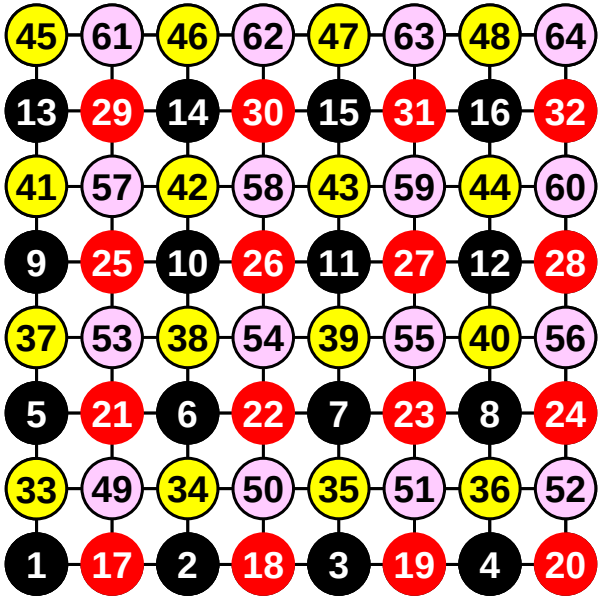
- Hybid 4x4 はFlat MPIと同じ，あるいは少し良い
- （並び替え＋F.T.）によるデータ局所化，メモリアクセス連続性確保によりHybrid 8x2，16x1の性能が大幅に改善
- Hybrid は「コア数が多く」，「コアあたりの問題規模が小さい」場合に特に有効
  - T2Kに適した並列プログラミングモデルである
  - ノード数がもっと増えるとHB 16x1の優位性が高まるかも知れない

- 背景
  - ターゲットアプリケーション
  - T2Kオープンスパコン
  - Hybrid vs. Flat MPI
- ノード内並列化手法の実装
  - MC，CM-RCMリオーダリング
  - First Touch，再並び替え
- HIDの可能性
- HIDの改良
- まとめ

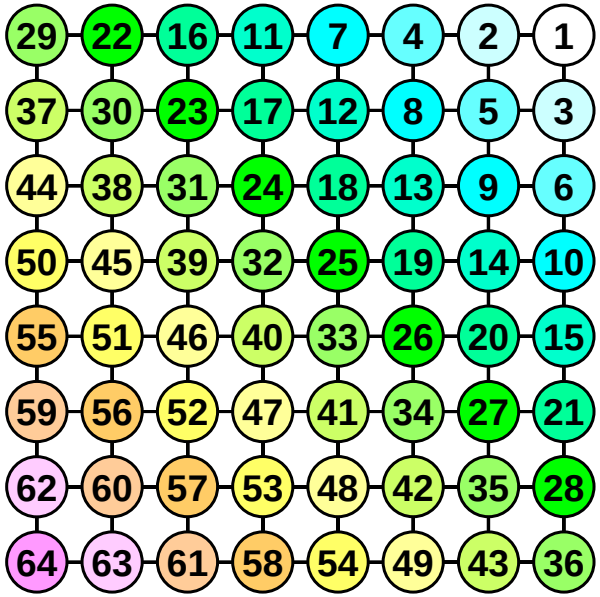
前処理付き反復法のSMP/MulticoreでのOpenMPによる並列化

- DAXPY, SMVP, Dot Products
  - 簡単
- 前処理：ILU系分解，前進後退代入
  - 大域的な依存性（Global dependency）
  - 並び替え（Reordering）による並列性の抽出
    - Multicolor Ordering (MC), Reverse-Cuthill-Mckee (RCM)
    - 同じ色内の要素は独立⇒並列化可能
  - 「地球シミュレータ」向け最適化　[KN 2002,2003]
    - 並列及びベクトル性能
  - CM-RCM：並列性高く安定

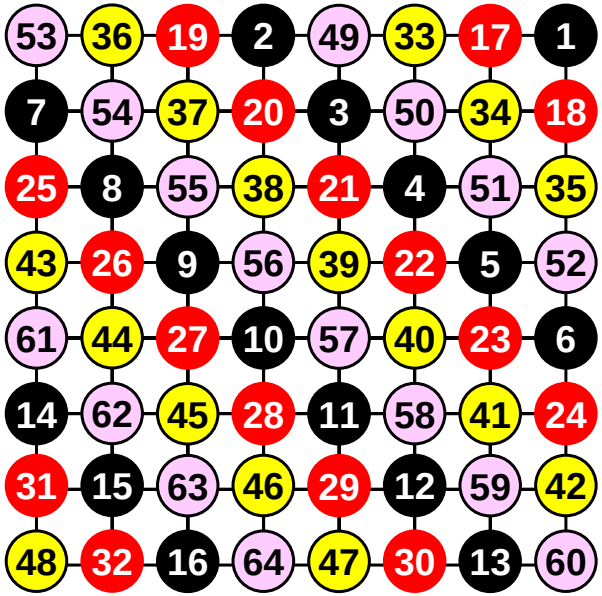
Ordering Methods



MC (Color#=4)
Multicoloring

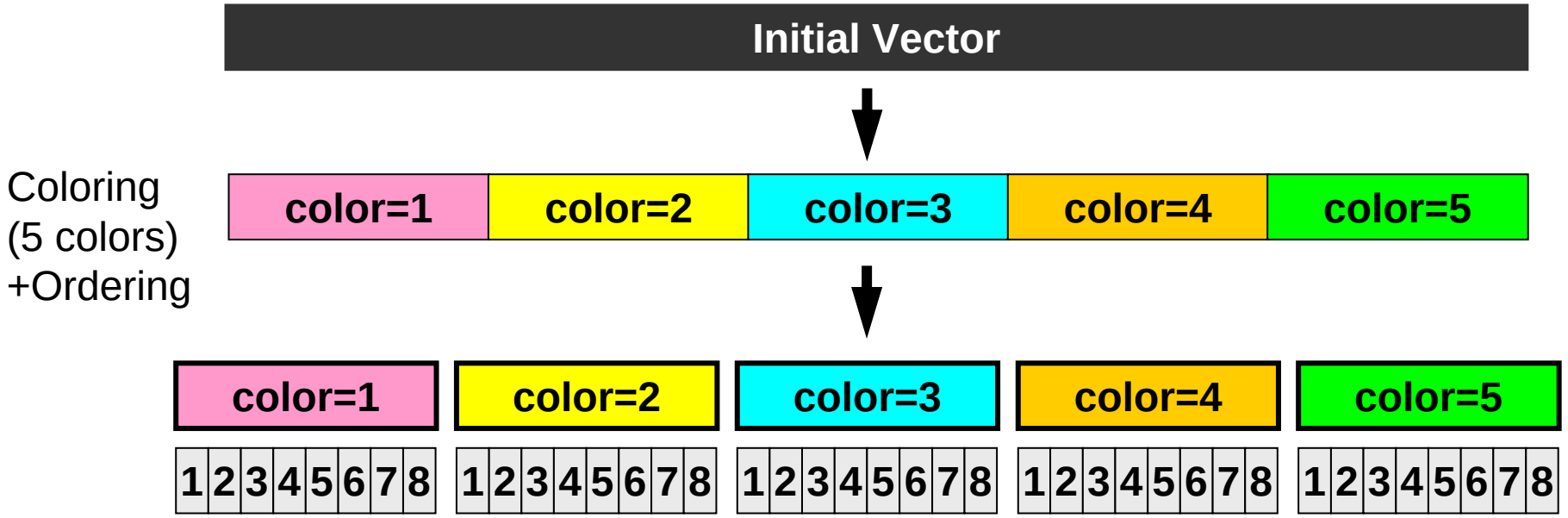


RCM
Reverse Cuthill-Mckee



CM-RCM (Color#=4)
Cyclic MC + RCM

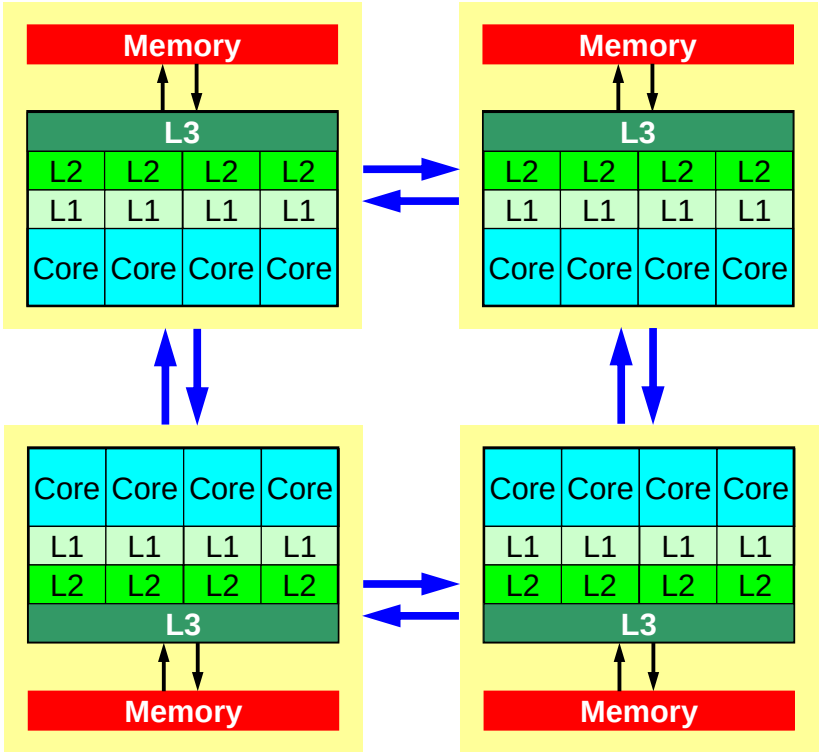
CM-RCMによる並べ替え（reordering）
5 colors, 8 threads



同じ「色」に属する要素は独立，したがって並列計算可能
⇒　スレッド並列化（OpenMP等）が可能

高速化のための手法

- First-Touch
  - ccNUMA Architecture
  - 配列のメモリ・ページ
    - 最初にtouchしたコアのローカルメモリ上に確保
  - 各ソケットのローカルメモリ
- 再並び替え
  - スレッド内での連続メモリアクセス
  - 各ソケットの連続した領域に各スレッド（コア）のメモリ・ページが確保される



First Touch Data Placement

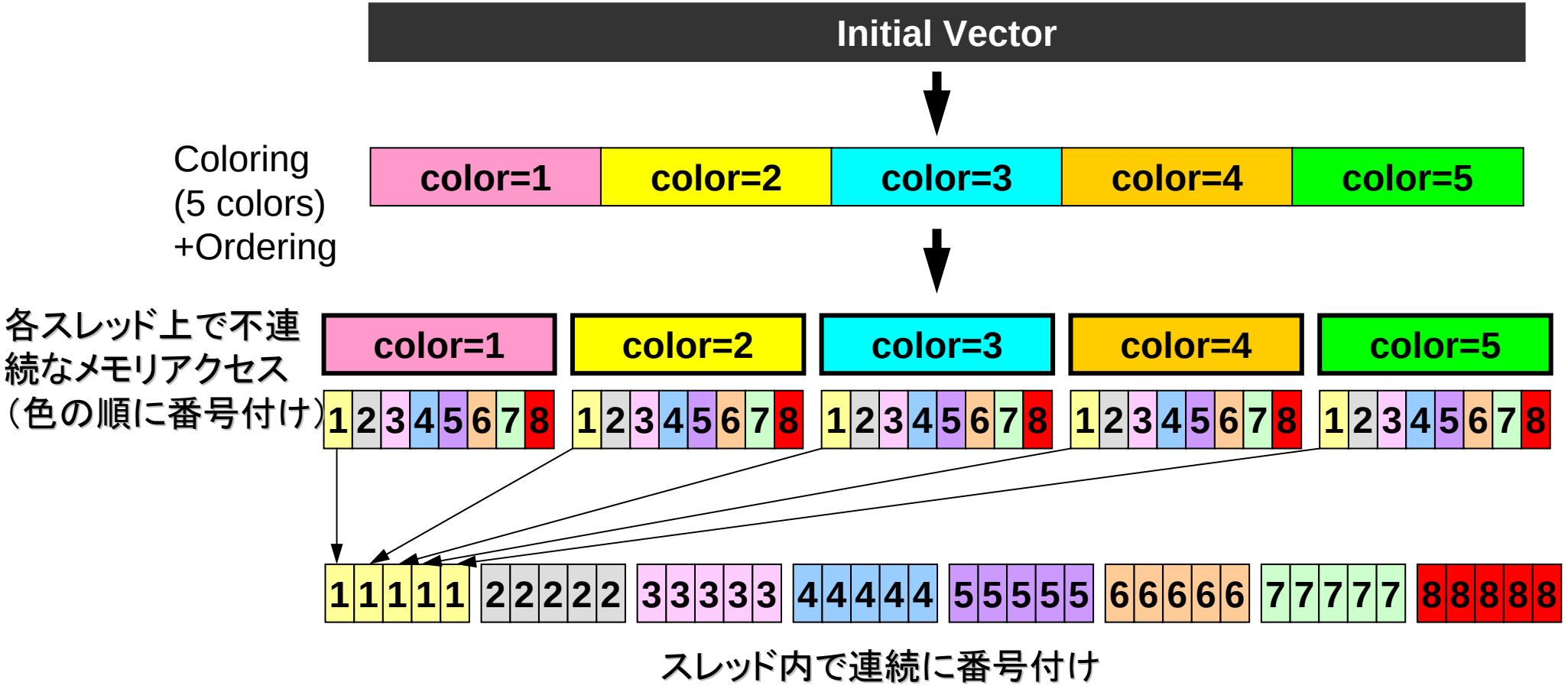
配列のメモリ・ページ:
最初にtouchしたコアのローカルメモリ上に確保
計算と同じ順番で初期化

```
  do lev= 1, LEVELtot
    do ic= 1, COLORtot(lev)
!$omp parallel do private(ip,i,j,isL,ieL,isU,ieU)
      do ip= 1, PEsmpTOT
      do i = STACKmc(ip,ic-1,lev)+1, STACKmc(ip,ic,lev)
        RHS(i)= 0.d0; X(i)= 0.d0; D(i)= 0.d0

        isL= indexL(i-1)+1
        ieL= indexL(i)
        do j= isL, ieL
          itemL(j)= 0; AL(j)= 0.d0
        enddo

        isU= indexU(i-1)+1
        ieU= indexU(i)
        do j= isU, ieU
          itemU(j)= 0; AU(j)= 0.d0
        enddo
      enddo
      enddo
!$omp omp end parallel do
    enddo
  enddo
```

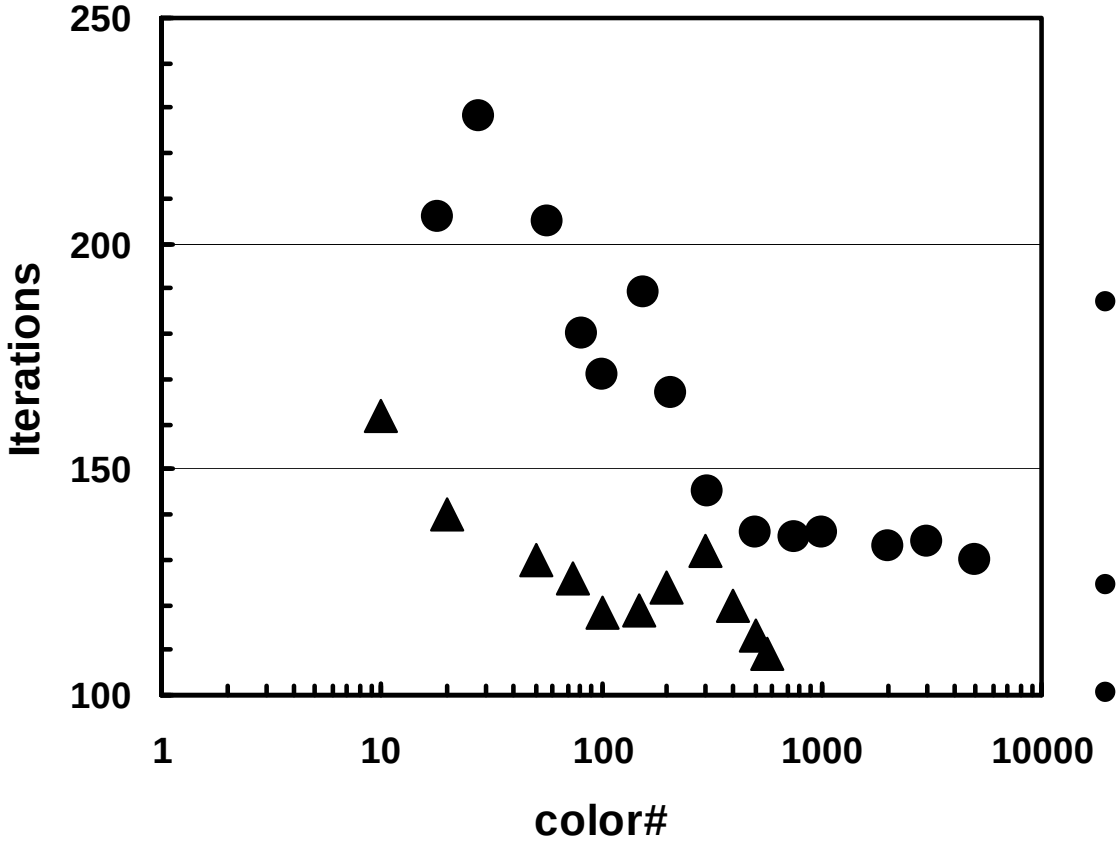
各スレッド上でメモリアクセスが連続となるように更なる並び替え
5 colors, 8 threads



計算効率：T2K 1ノード HB16x1

SGS/GPBiCGソルバー

（均質（E_2=1.0）三次元弾性問題1.59M DOF：80^3要素）

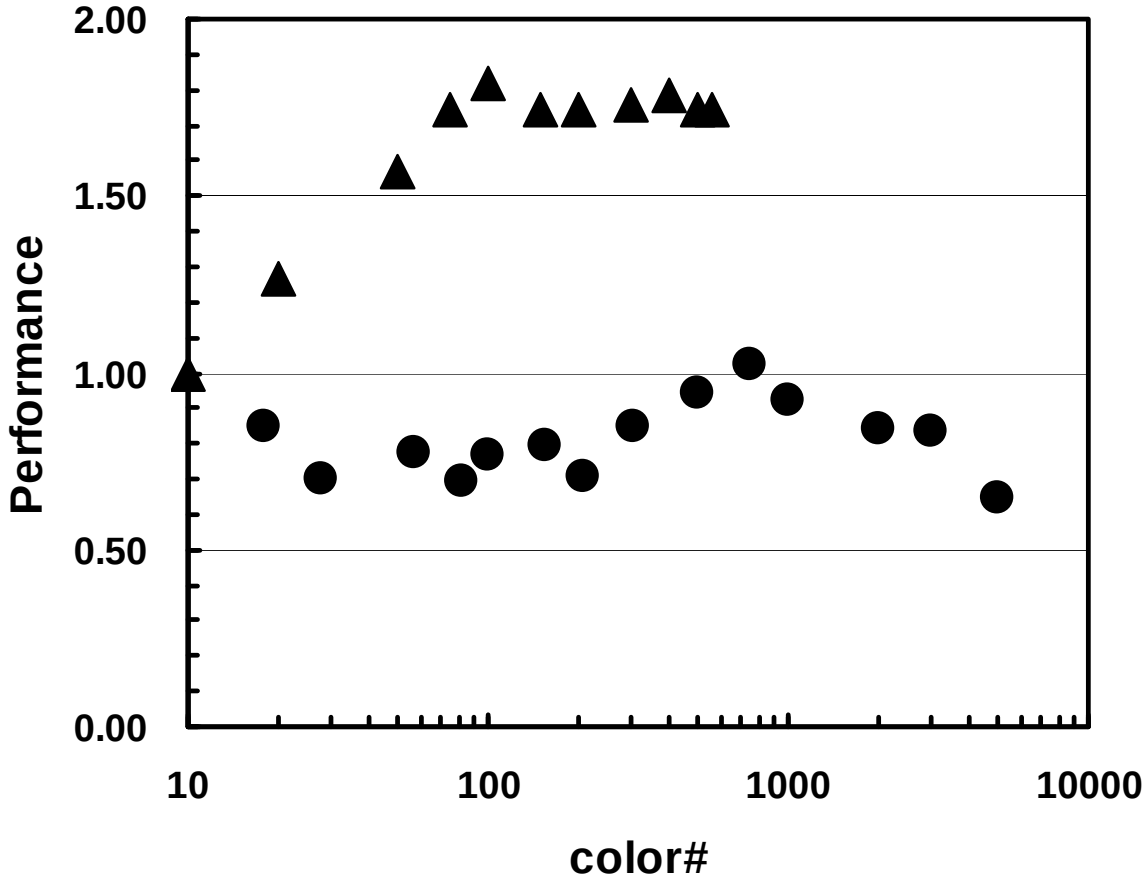


● Mulricoloring（MC）
▲ CM-RCM

- 色数が多いほど反復回数減少（Incompatible Nodes）
- CM-RCMの方が収束良
- CM-RCMで色数最大（516）の場合＝RCM

計算効率：T2K 1ノード HB16x1

SGS/GPBiCGソルバーの計算時間（1.59M DOF）
CM-RCM(10)の場合を基準

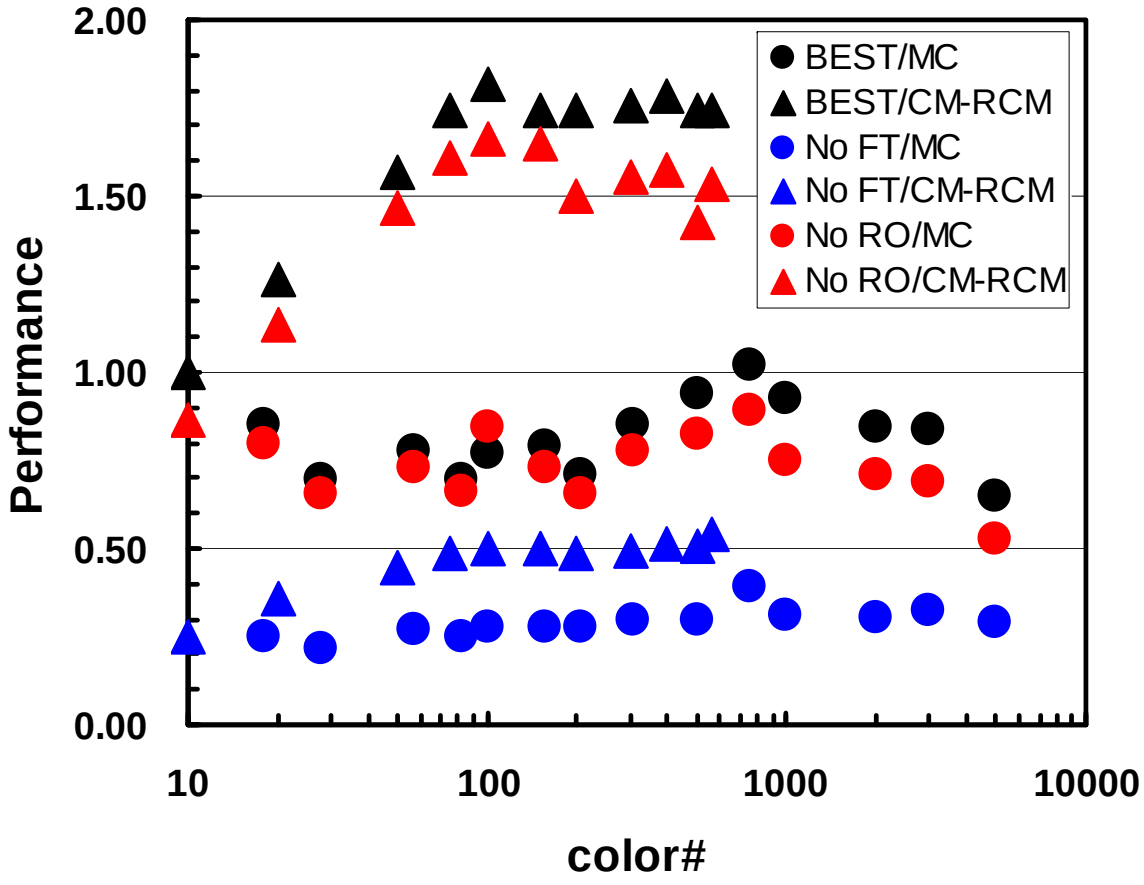


● Mulricoloring（MC）
▲ CM-RCM

- 色数が1000を超えると，反復回数減少しても計算時間が増加する場合がある
  - 同期オーバーヘッド
- Original RCMが比較的効率良い
  - Load Imbalance効果少

計算効率:T2K 1ノード HB16x1

SGS/GPBiCGソルバーの計算時間(1.59M DOF)
BEST/CM-RCM(10)の場合を基準



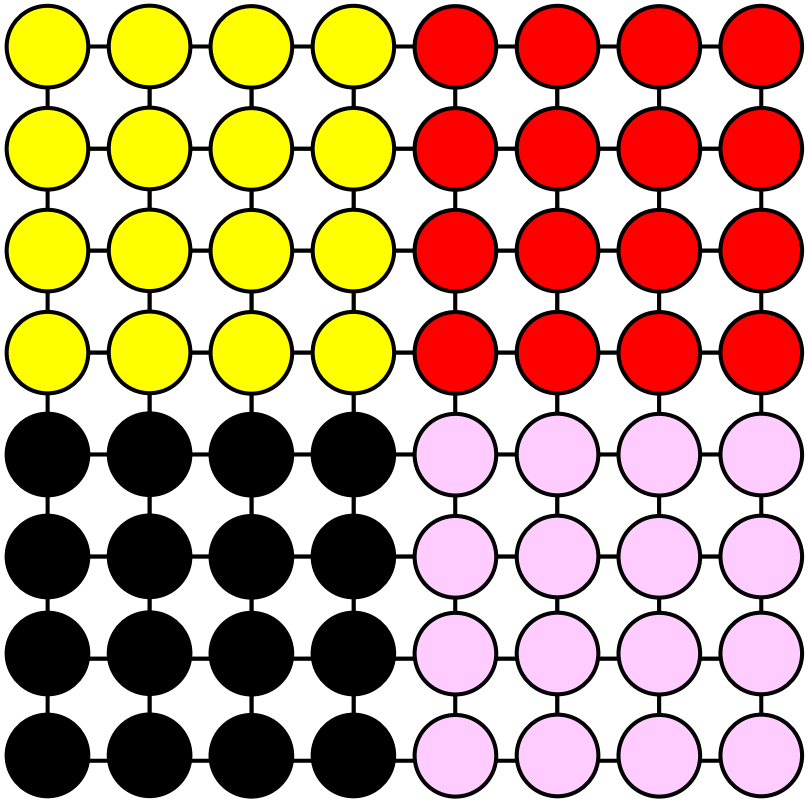
	First Touch	再並び替え
BEST	○	○
No FT		○
No RO	○	

- 前頁の結果:BEST
- これ+NUMA controlも必要
- 再並び替えの効果
  - 色数多いとき顕著

- 背景
  - ターゲットアプリケーション
  - T2Kオープンスパコン
  - Hybrid vs. Flat MPI
- ノード内並列化手法の実装
  - MC，CM-RCMリオーダリング
  - First Touch，再並び替え
- HIDの可能性
- HIDの改良
- まとめ

Hybridにおけるノード内並列化手法

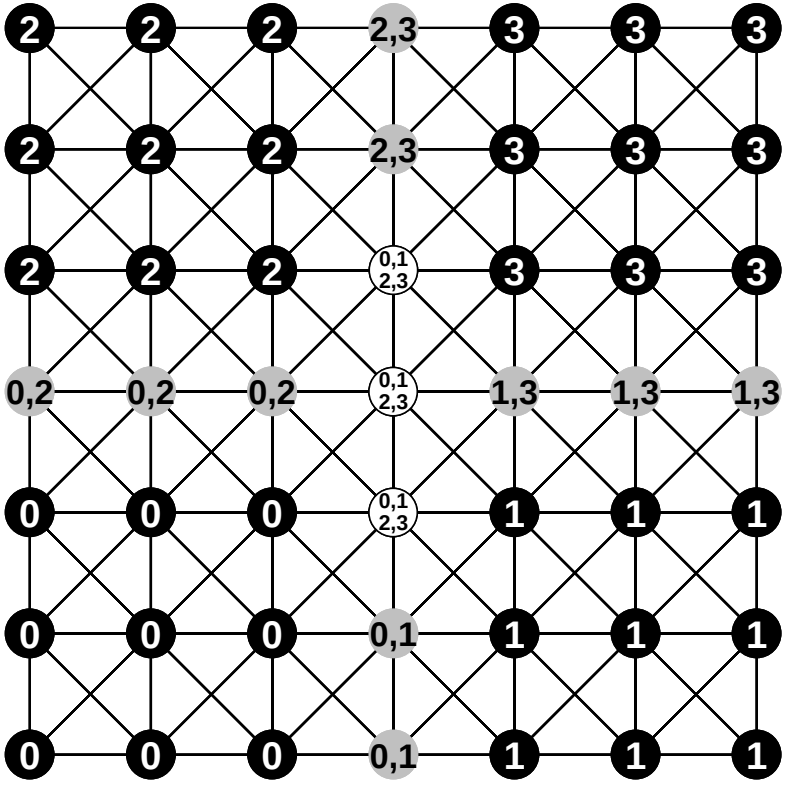
- MC, CM-RCM
  - Fill-inレベルが上がると(依存性の無い)グラフの構造はとても複雑になる。
  - 依存性の変化:ILUT
- 領域分割(Block Jacobi)的
  - オーバーラップ等を考慮するとデータ構造がとても複雑になる
    - MPIでやった方が簡単
  - HIDの適用
    - Hierarchical Interface Decomposition

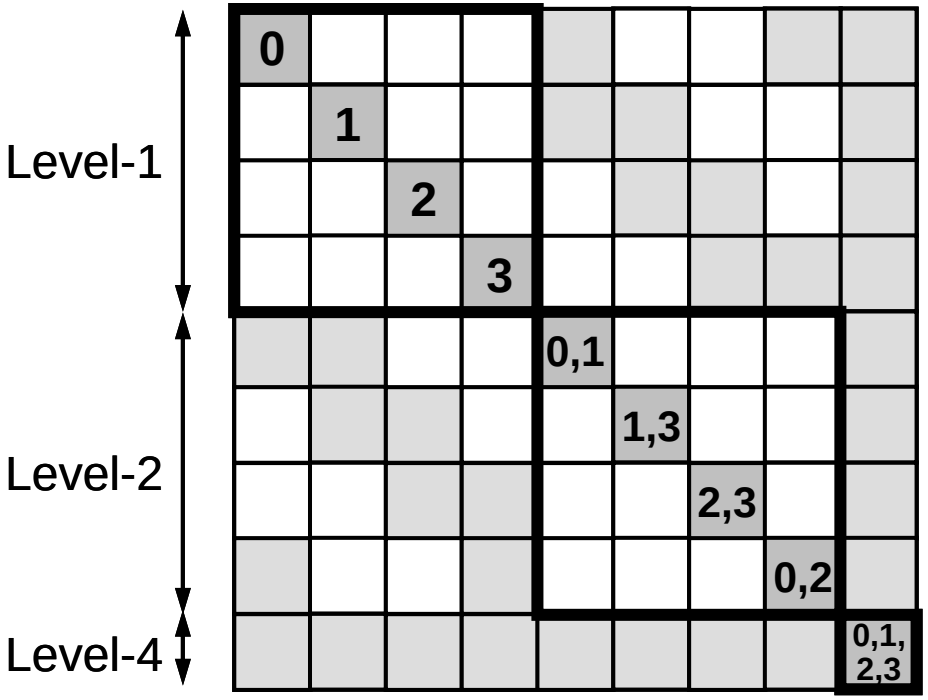


HID：Hierarchical Interface Decomposition［Henon & Saad］

- 階層型領域間境界分割：分散メモリ向け
- 二言くらいで言うと：
  - 階層的な領域分割
    - Nested Dissection
  - 各レベルでは各領域は直接結合しない⇒レベル内並列性

Parallel ILU for each Connector at each LEVEL



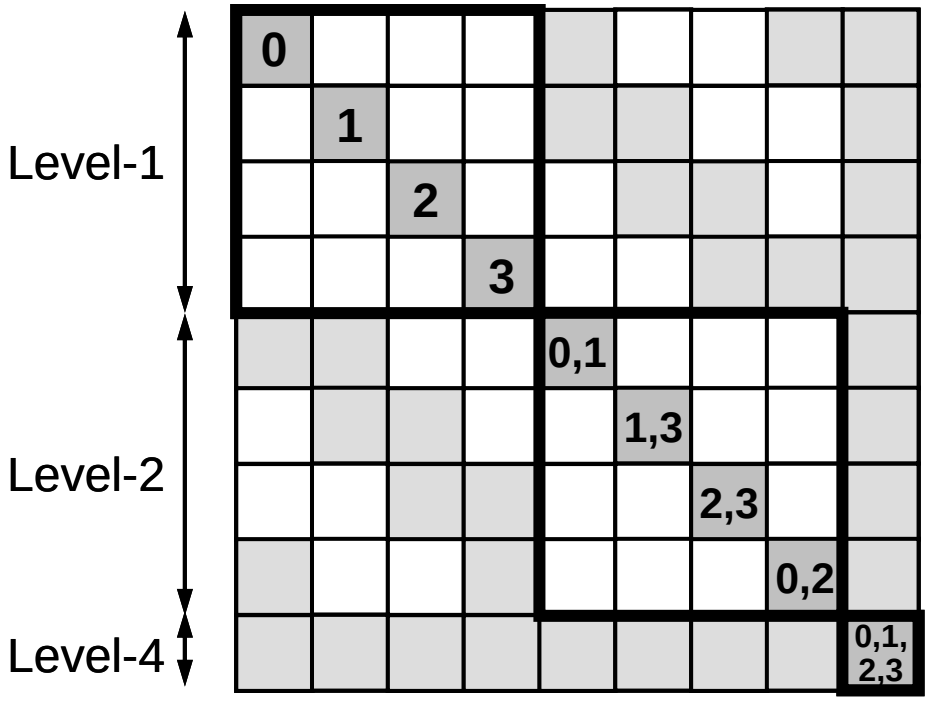


- sub-domainの番号を0,1,2,3とする
- 数字は隣接するsub-domainの番号
- sub-domain：MPIプロセス，OpenMPスレッド

- レベルの若い順に番号を振りなおす
- 各レベル内で不完全LU（0）分解を並列に実施可能

各レベルにおける通信の発生(分散メモリ)

- 「高い」レベルのコネクタに属する節点群における計算
  - 「低い」レベルのコネクタの計算結果を利用
    - 計算済
  - 隣接する「低い」レベルのコネクタに属する節点群が他の領域に属している場合
    - 通信が必要となる



Forward Substitution

```
do lev= 1, LEVELtot
  do i= LEVindex(lev-1)+1, LEVindex(lev)
    SW1= WW(3*i-2,R); SW2= WW(3*i-1,R); SW3= WW(3*i  ,R)
    isL= INL(i-1)+1; ieL= INL(i)
    do j= isL, ieL
      k= IAL(j)
      X1= WW(3*k-2,R); X2= WW(3*k-1,R); X3= WW(3*k  ,R)
      SW1= SW1 - AL(9*j-8)*X1 - AL(9*j-7)*X2 - AL(9*j-6)*X3
      SW2= SW2 - AL(9*j-5)*X1 - AL(9*j-4)*X2 - AL(9*j-3)*X3
      SW3= SW3 - AL(9*j-2)*X1 - AL(9*j-1)*X2 - AL(9*j  )*X3
    enddo
    X1= SW1; X2= SW2; X3= SW3
    X2= X2 - ALU(9*i-5)*X1
    X3= X3 - ALU(9*i-2)*X1 - ALU(9*i-1)*X2
    X3= ALU(9*i  )*  X3
    X2= ALU(9*i-4)*( X2 - ALU(9*i-3)*X3 )
    X1= ALU(9*i-8)*( X1 - ALU(9*i-6)*X3 - ALU(9*i-7)*X2)
    WW(3*i-2,R)= X1; WW(3*i-1,R)= X2; WW(3*i  ,R)= X3
  enddo

  call SOLVER_SEND_RECV_3_LEV(lev,…):

enddo
```

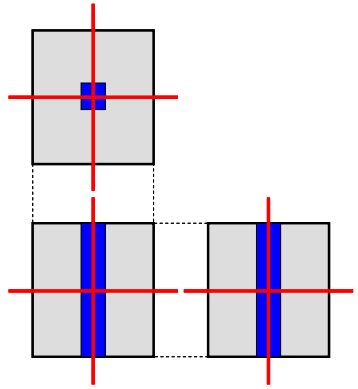
余計な通信が発生

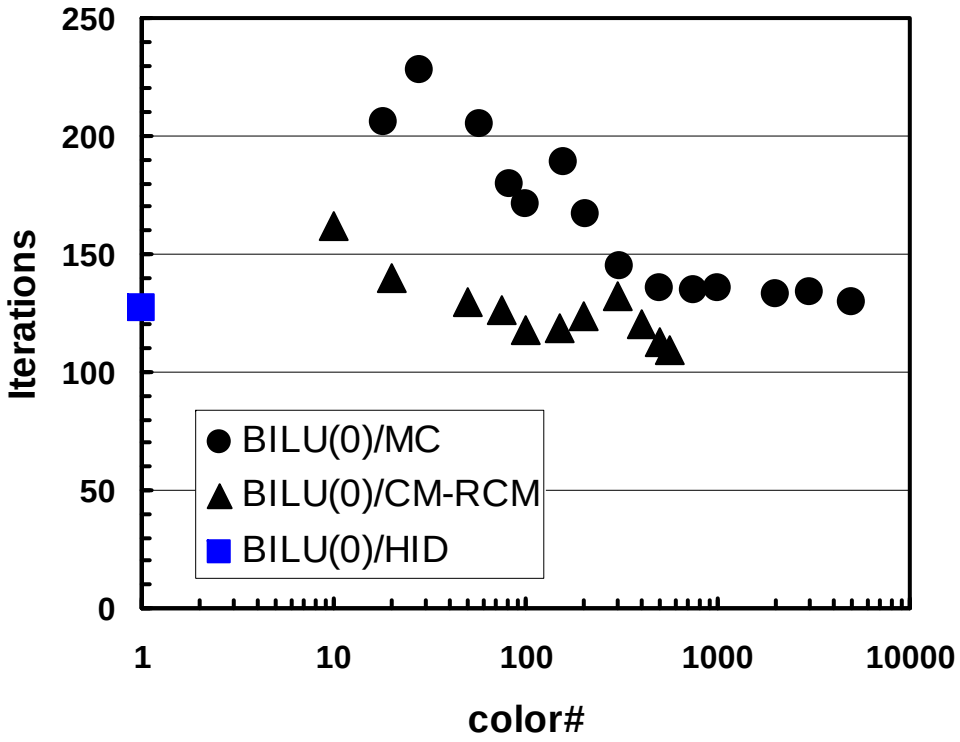
Communications using Hierarchical Comm. Tables.

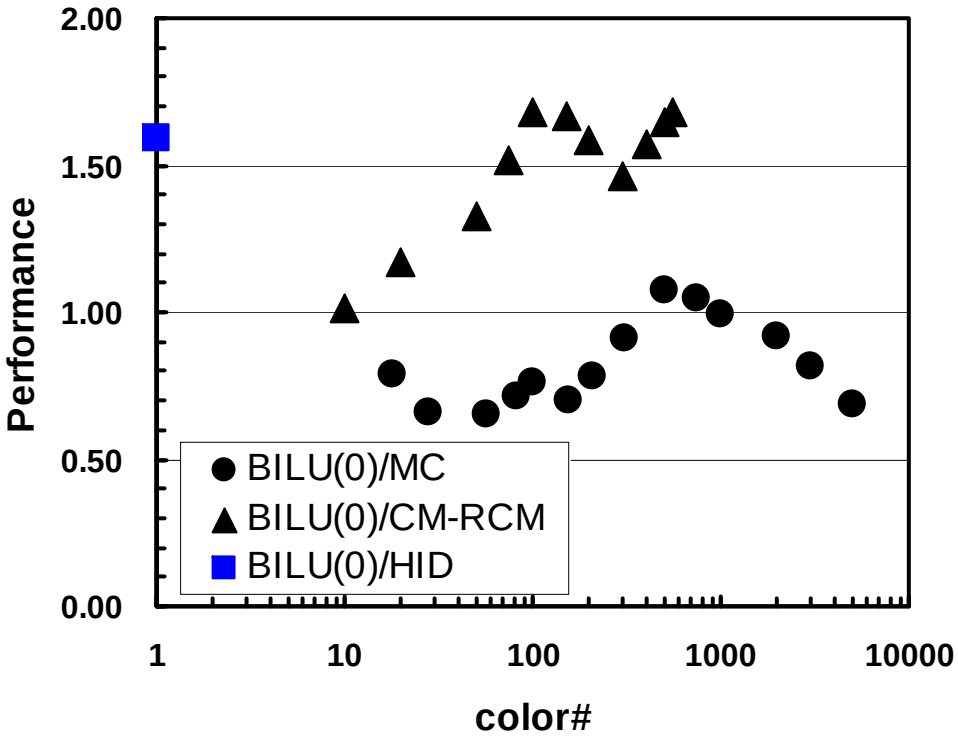
計算効率：T2K　1ノード HB16x1

BILU(0)/GPBiCGソルバー

（**均質（E_2=1.0）**三次元弾性問題1.59M DOF）







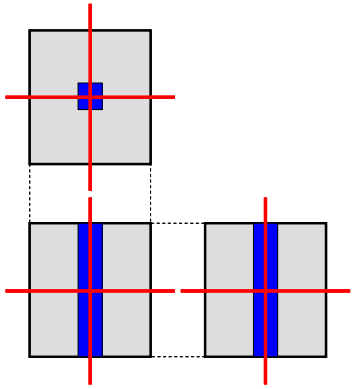
計算効率：T2K　1ノード HB16x1

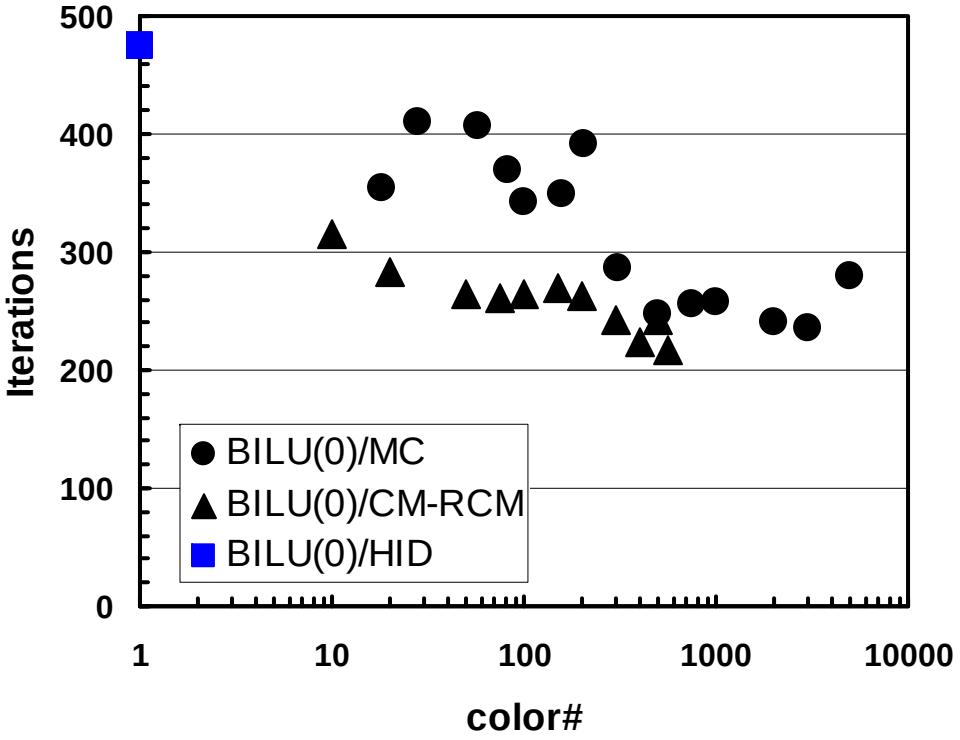
BILU(0)/GPBiCGソルバー

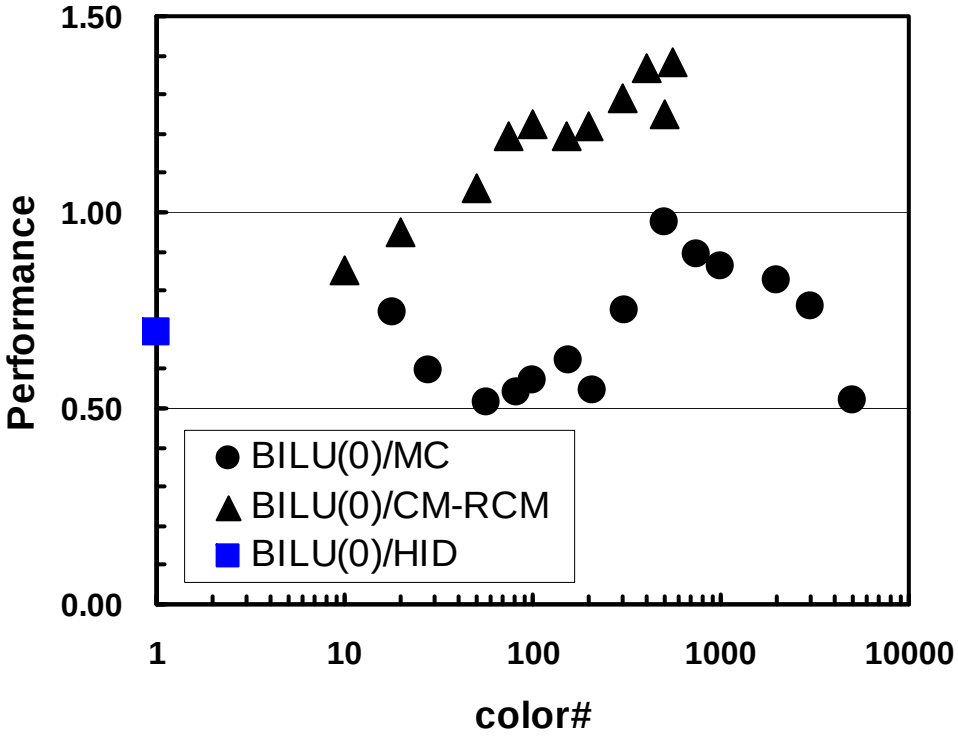
（**不均質（$E_2=10^3$）**三次元弾性問題1.59M DOF）

反復回数は大幅に増加

（16色なのでオーバーヘッドは少ない）



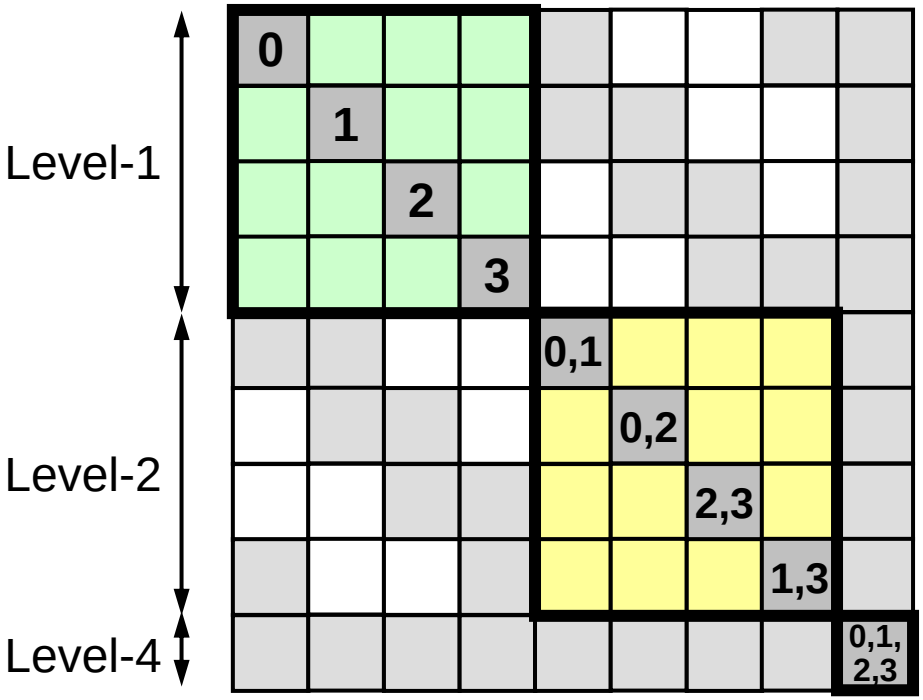




- 背景
  - ターゲットアプリケーション
  - T2Kオープンスパコン
  - Hybrid vs. Flat MPI
- ノード内並列化手法の実装
  - MC，CM-RCMリオーダリング
  - First Touch，再並び替え
- HIDの可能性
- HIDの改良
- まとめ

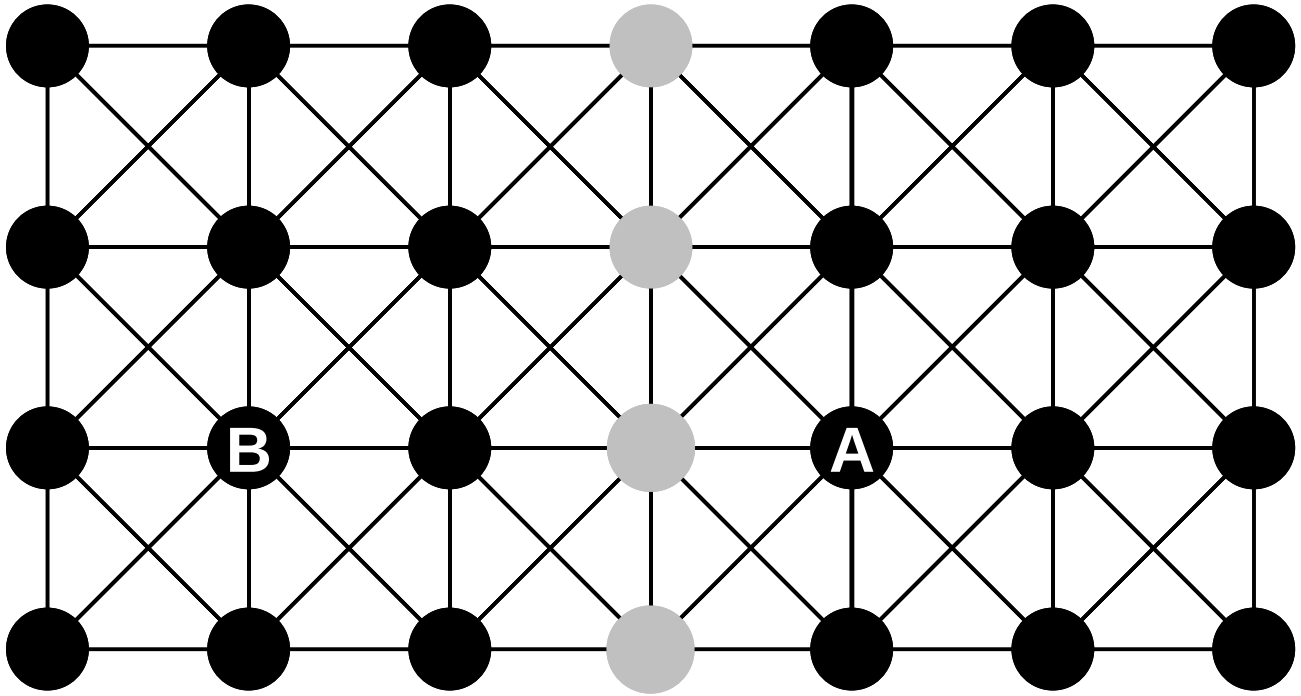
HIDの本来の問題点

- オリジナルのHIDはfill-inのorderが高くなると，領域外の（同じレベルにある節点の）fill-inの影響を考慮できない



Sample Graph

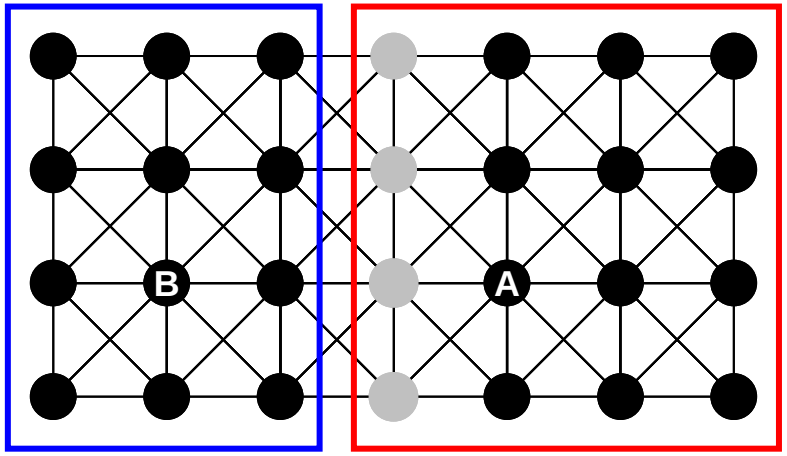
(A) could be referred from (B)
for ILU(2) (depends on numbering)

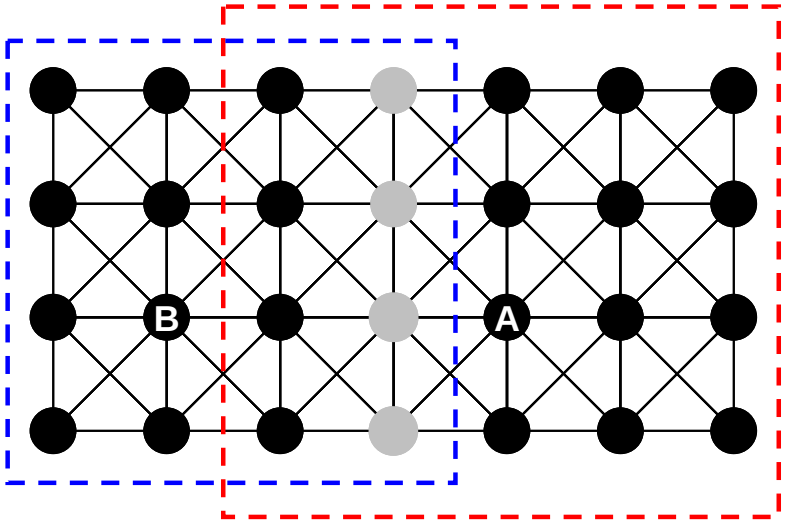


level-1
level-2

Domain Decomposition & Local Data Set

level-1 ●
level-2 ●





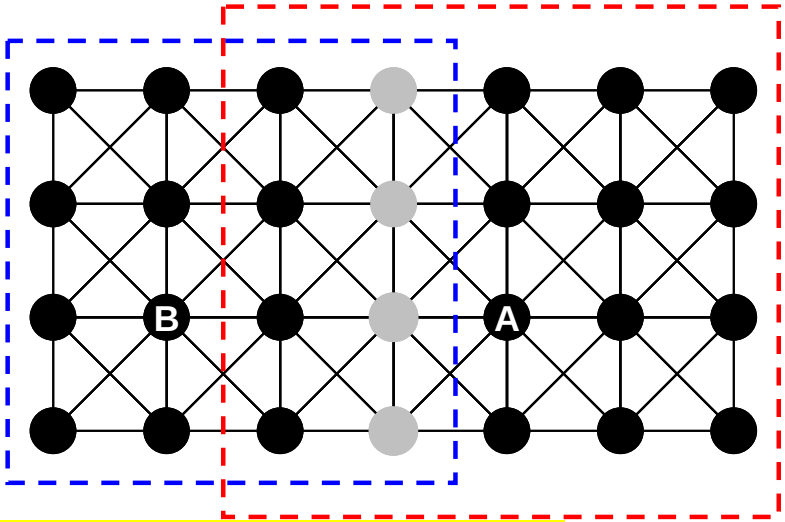
Node-based Domain Decomposition (Internal Nodes)

Distributed Local Data (Internal+External Nodes)

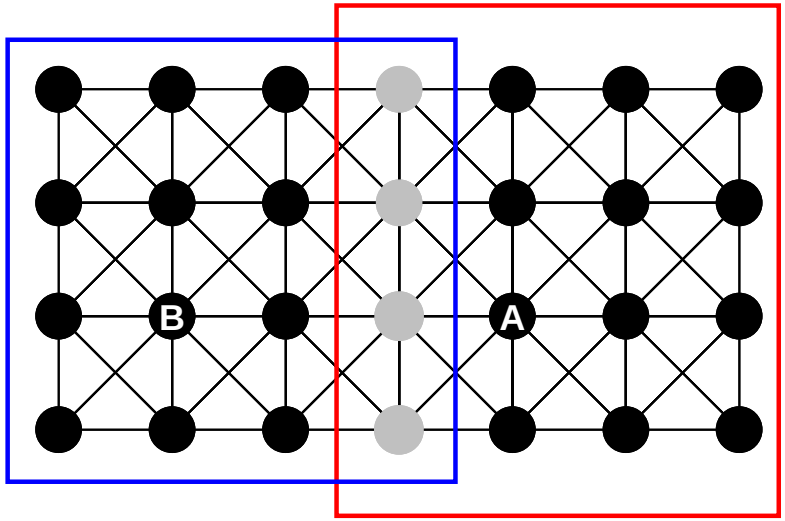
Original Local Data Set

- Original HID
  - NO overlapping/1-layer overlapping
  - cannot consider the effects of fill-ins of higher order for external nodes at *same level*.
    - Effect of “A” is not considered for “B” in BILU(2)

Distributed Local Data



Range for “Global” Operations”

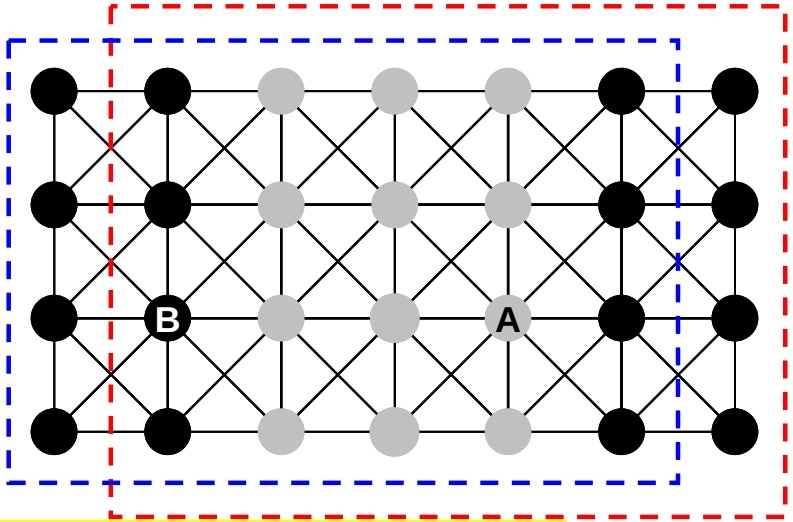


level-1
level-2

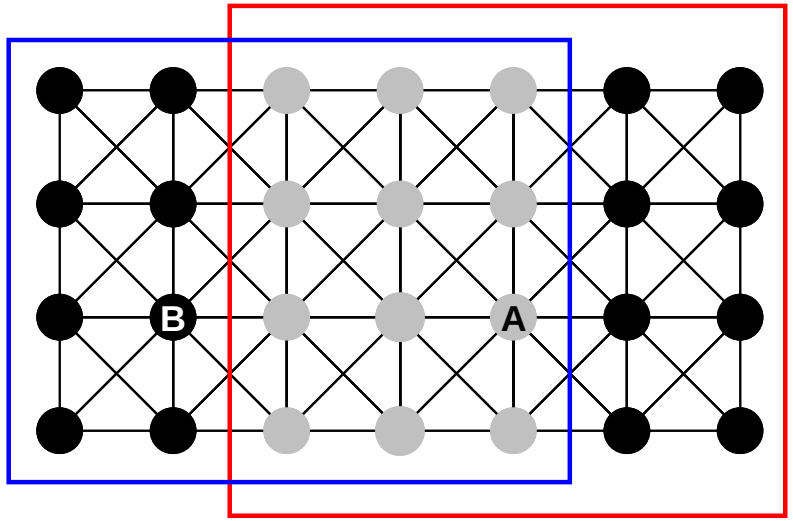
Remedy: Thicker Layers of Separator

- Thicker Separator
  - HID-new
  - can consider the effects of fill-ins of higher order for external nodes at *same level*.
    - Effect of “A” can be considered for “B” in BILU(2)
  - **In global manner**
  - **seems to provide more robust convergence than Remedy 1.**
  - **difficulty for load-balancing**

Distributed Local Data



Range for “Global” Operations”

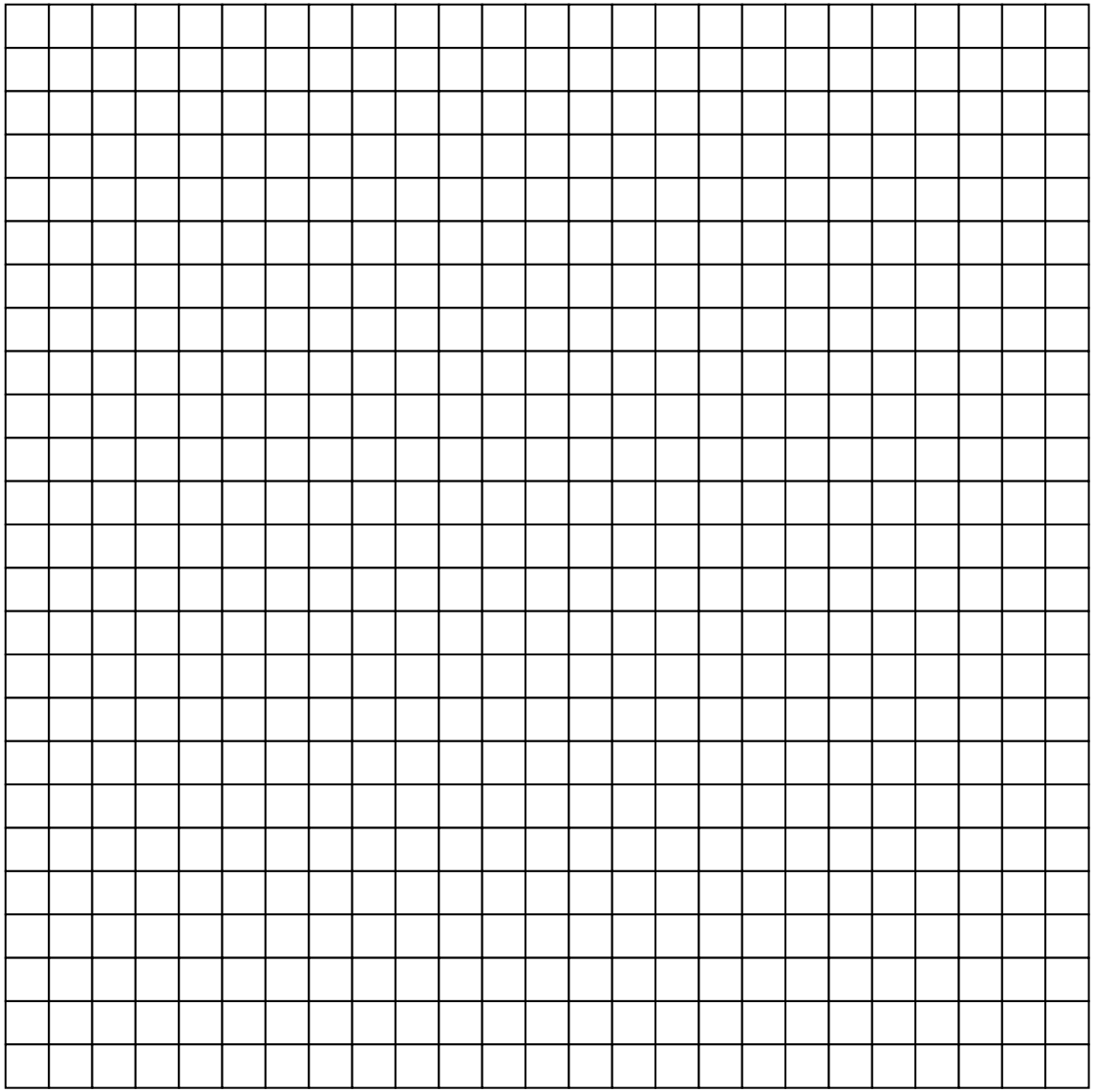


level-1
level-2

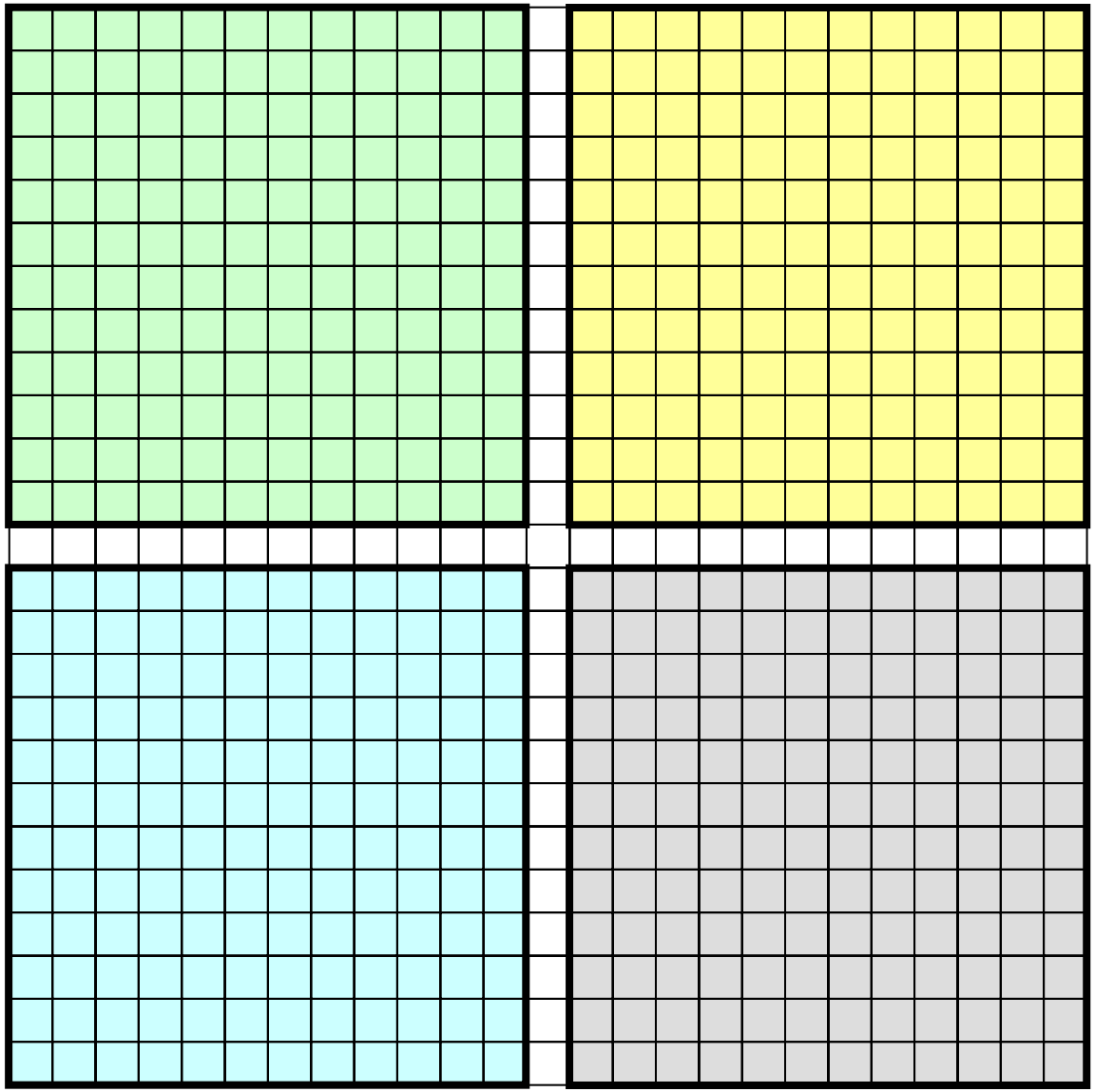
対策: Thicker Separator

- 関連研究
  - Takeshi Iwashita and Masaaki Shimasaki; "Block Red-Black Ordering: A New Ordering Strategy for Parallelization of ICCG Method", International Journal of Parallel Programming, Vol. 31, No. 1, (2003), pp.55-75
    - 差分格子
    - Red-Blackの2レベルを交互に解く
  - 岩下武史，高橋康人，中島浩 「代数ブロック化多色順序付け法による並列化ICCGソルバの性能評価」，2009年並列／分散／協調処理に関する『仙台』サマー・ワークショップ（SWoPP仙台2009）
    - より一般的な問題への拡張

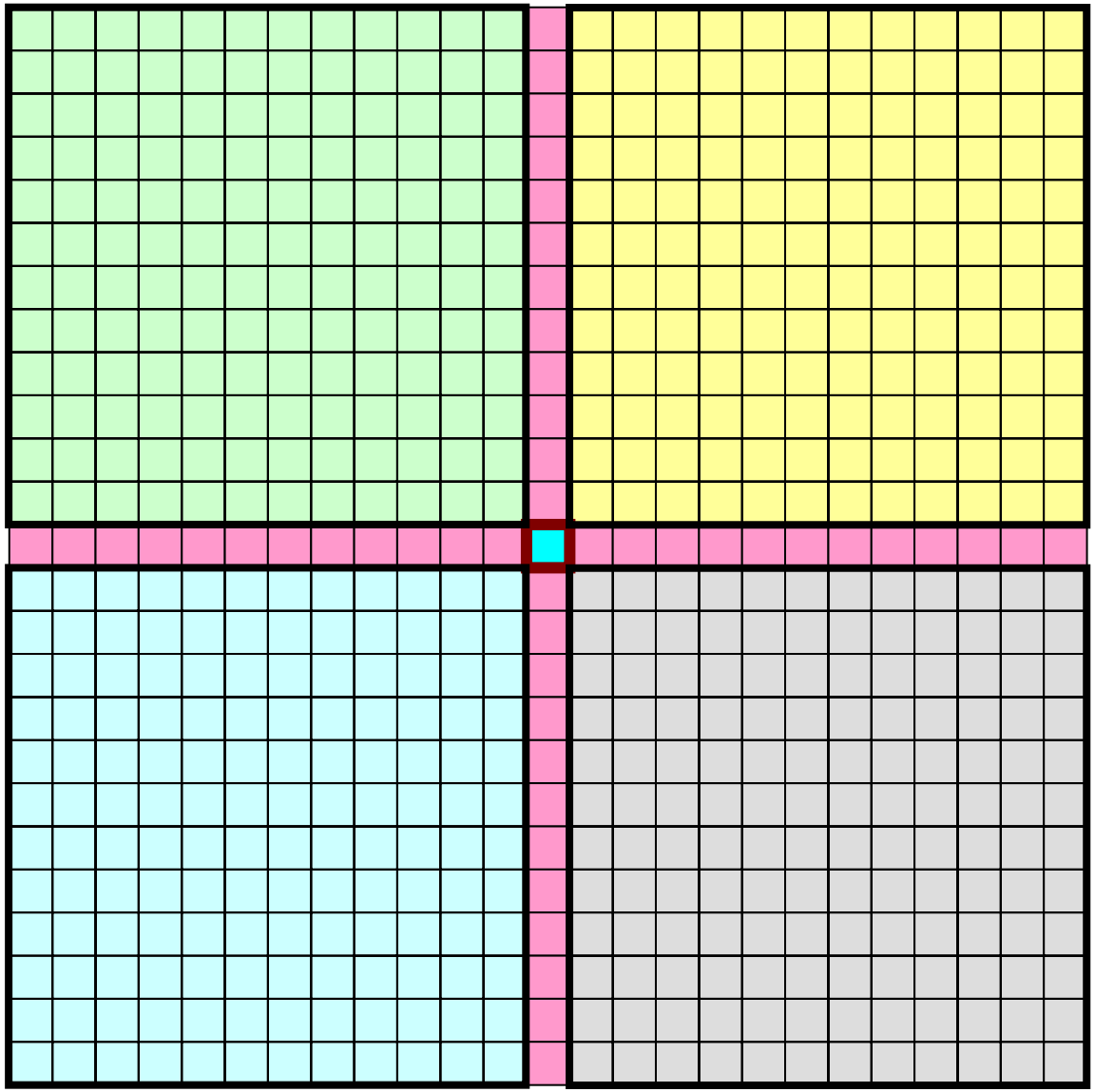
HID



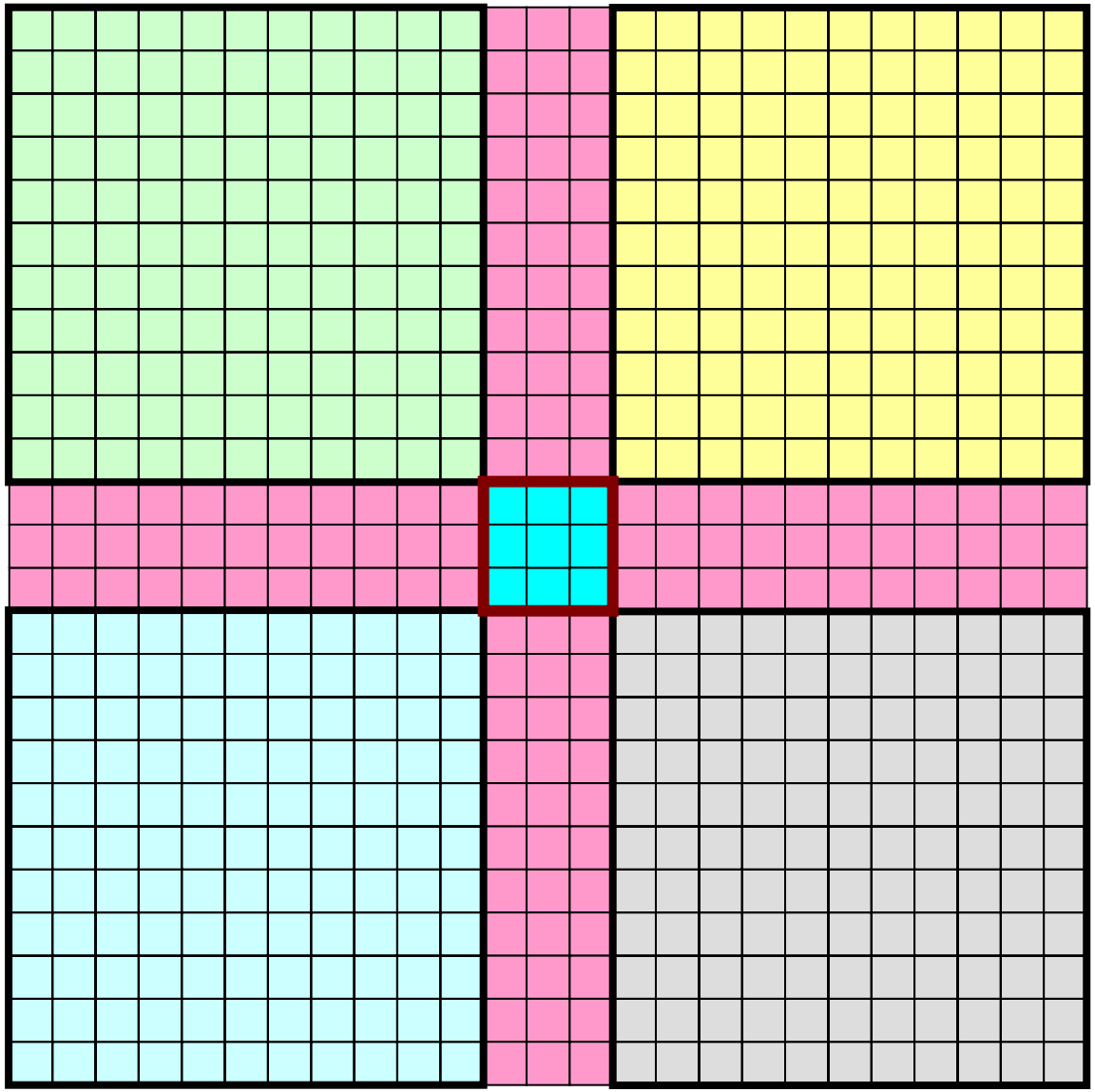
HID



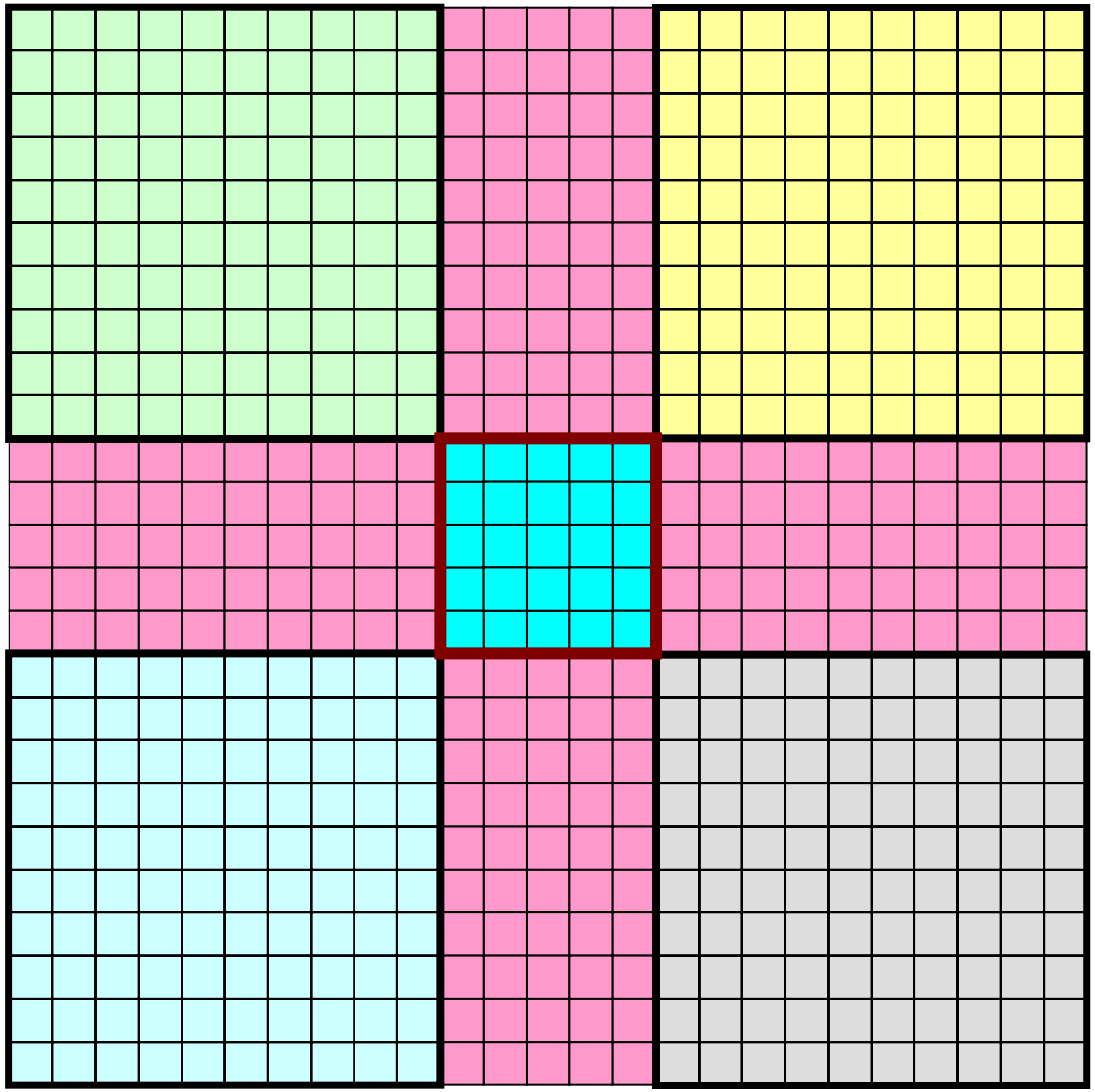
HID-1（separator厚さ=1）



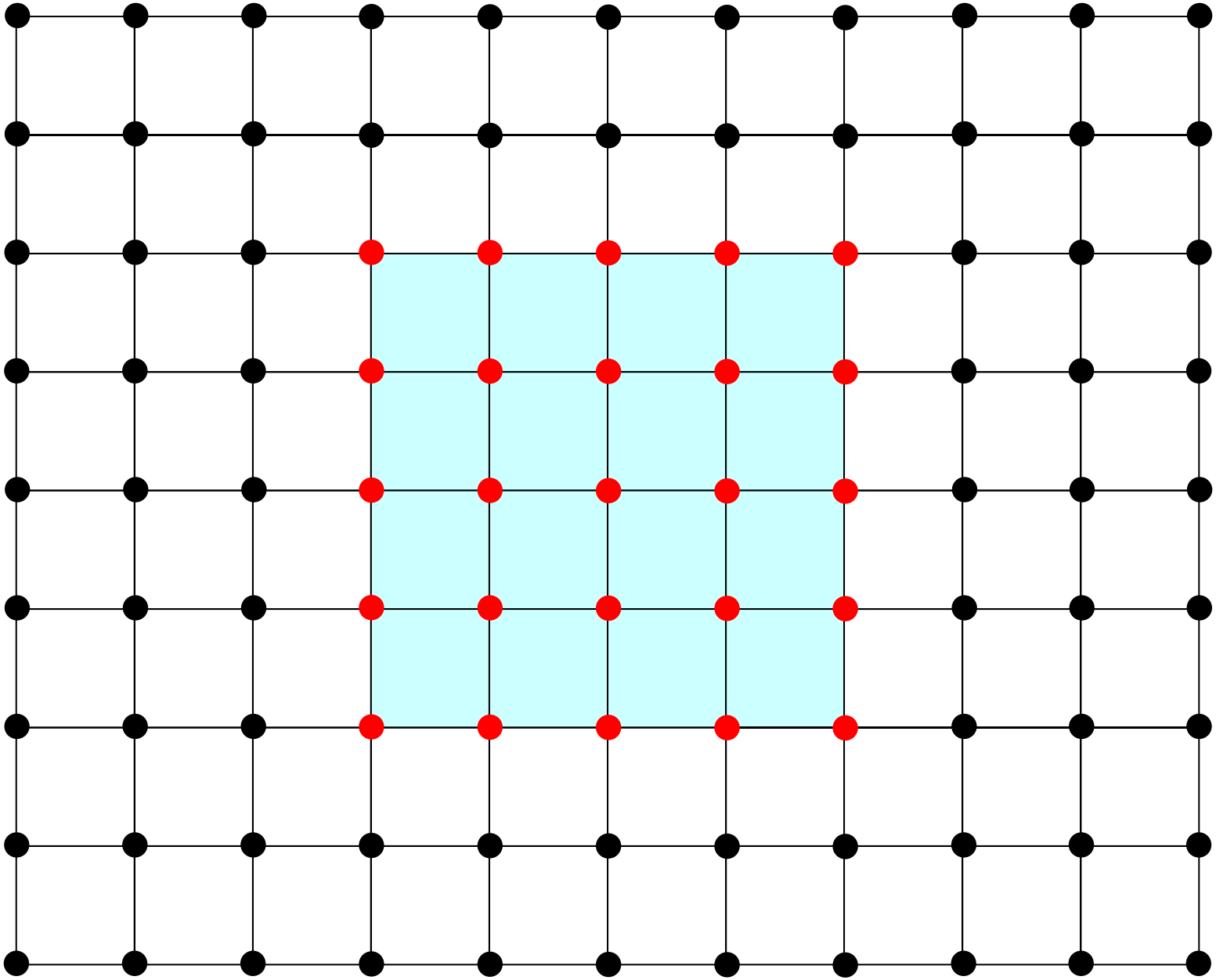
HID-3（separator厚さ=3）



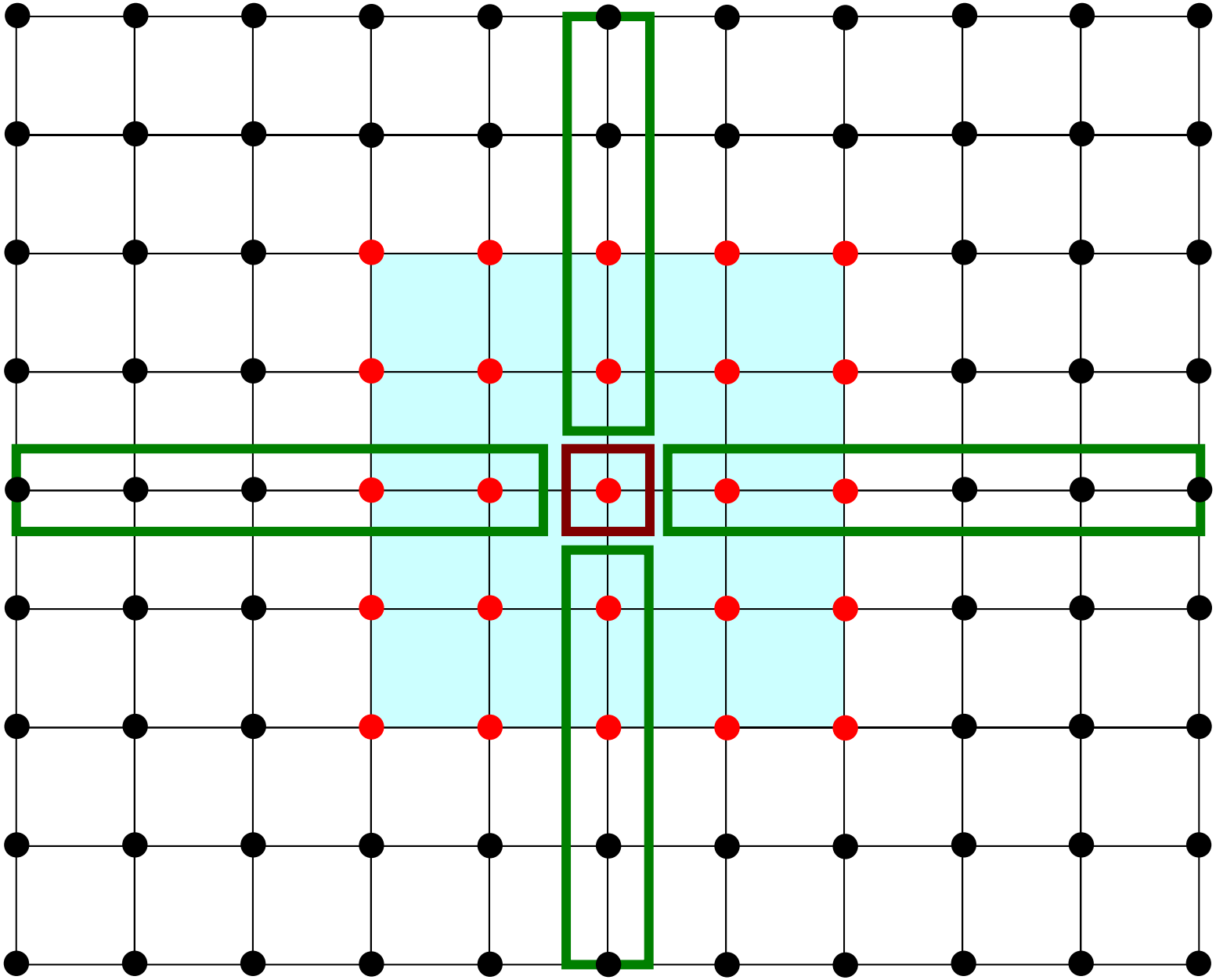
HID-5（separator厚さ=5）



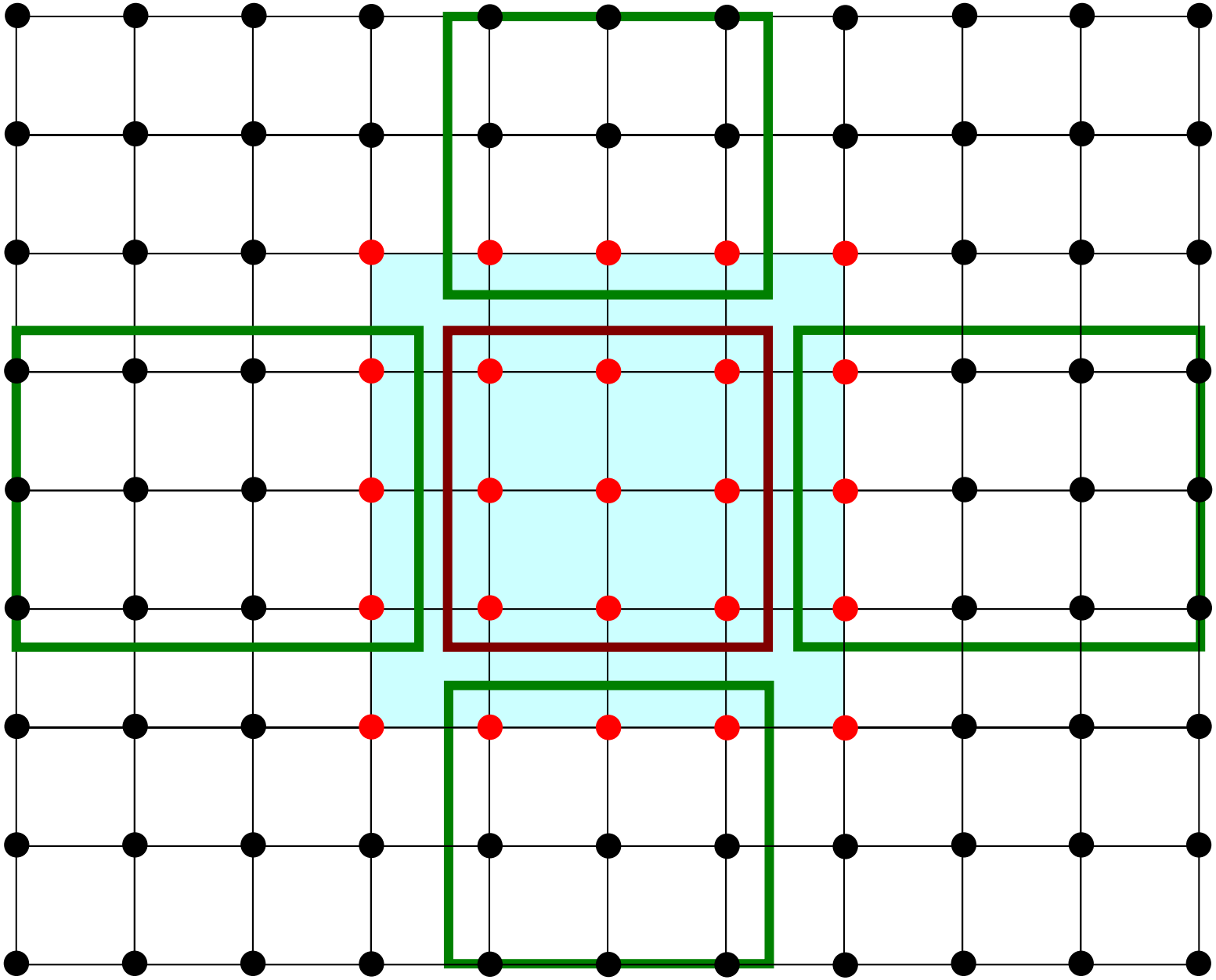
本問題の場合



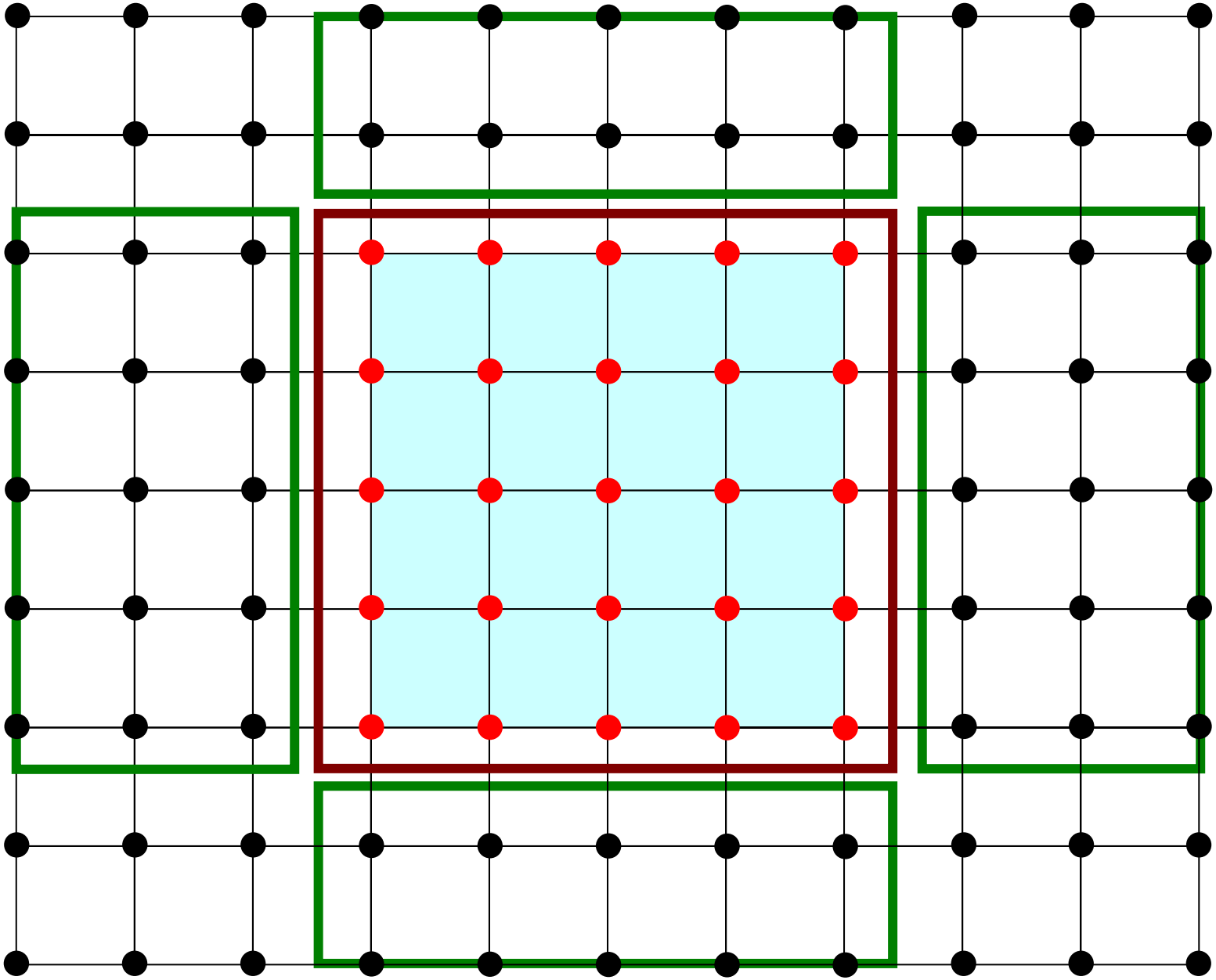
HID-1：結合の強い節点，異なったレベル



HID-3



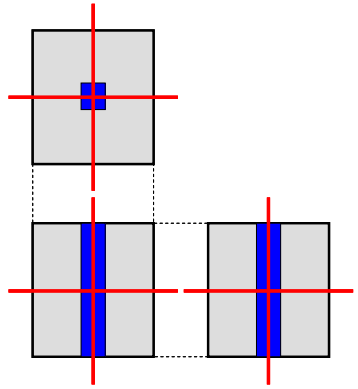
HID-5：物性の異なる領域をカバー可

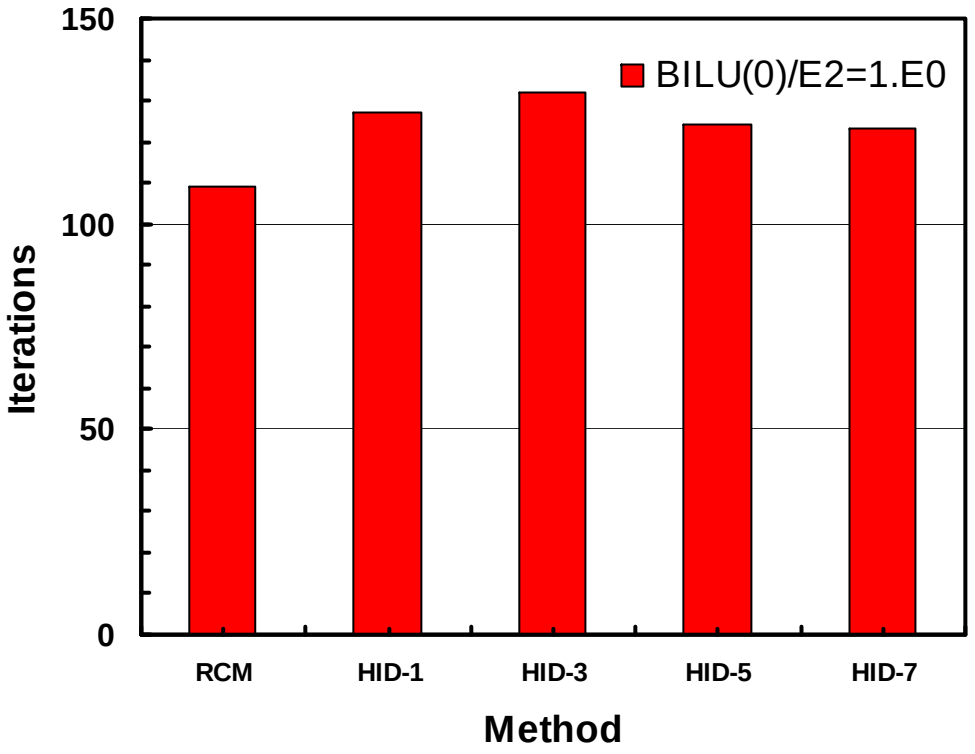


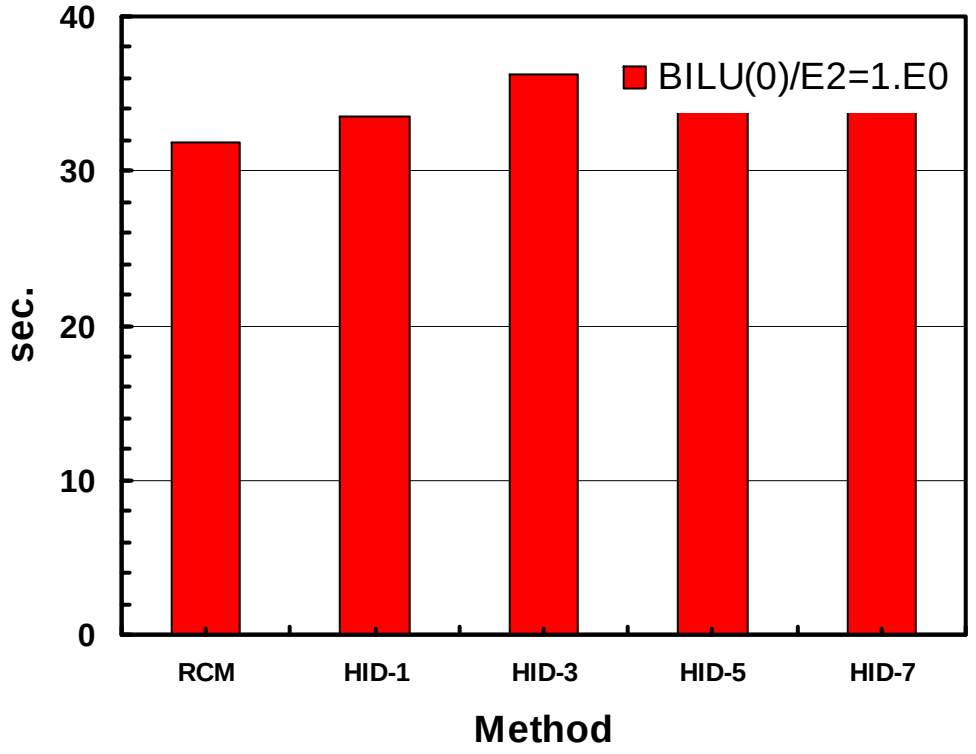
計算効率：T2K　1ノード HB16x1

BILU(0)/GPBiCGソルバー

（**均質（E_2=1.0）**三次元弾性問題1.59M DOF）



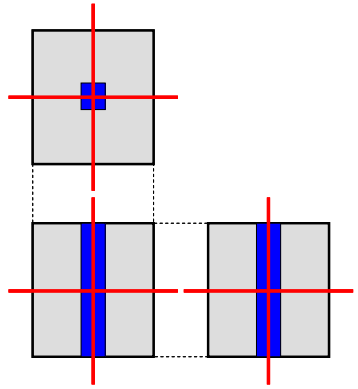


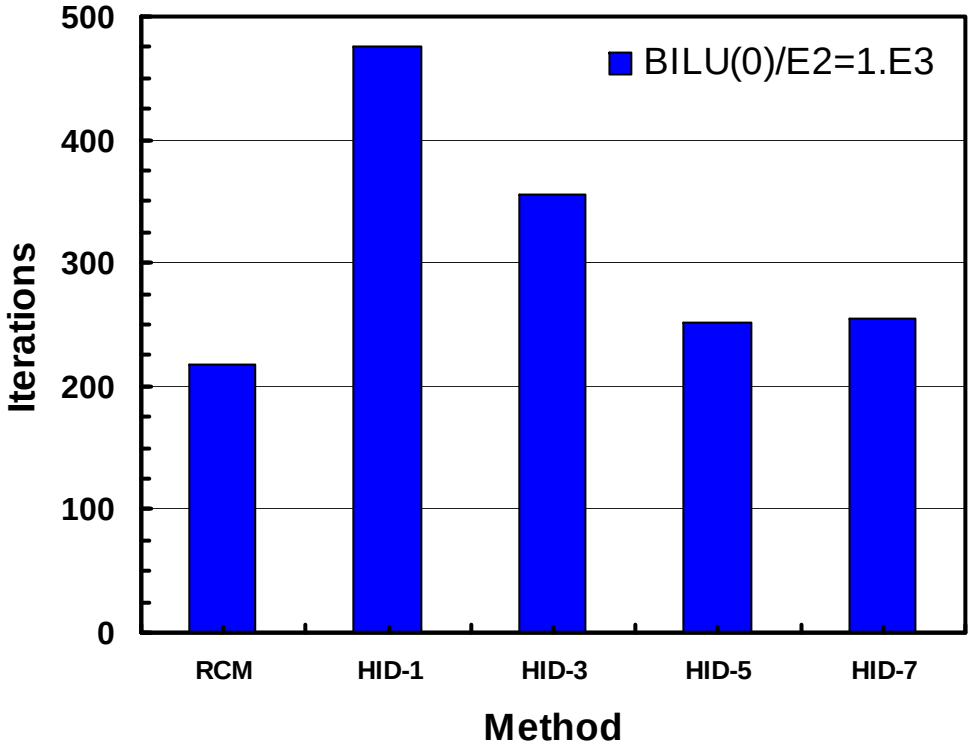


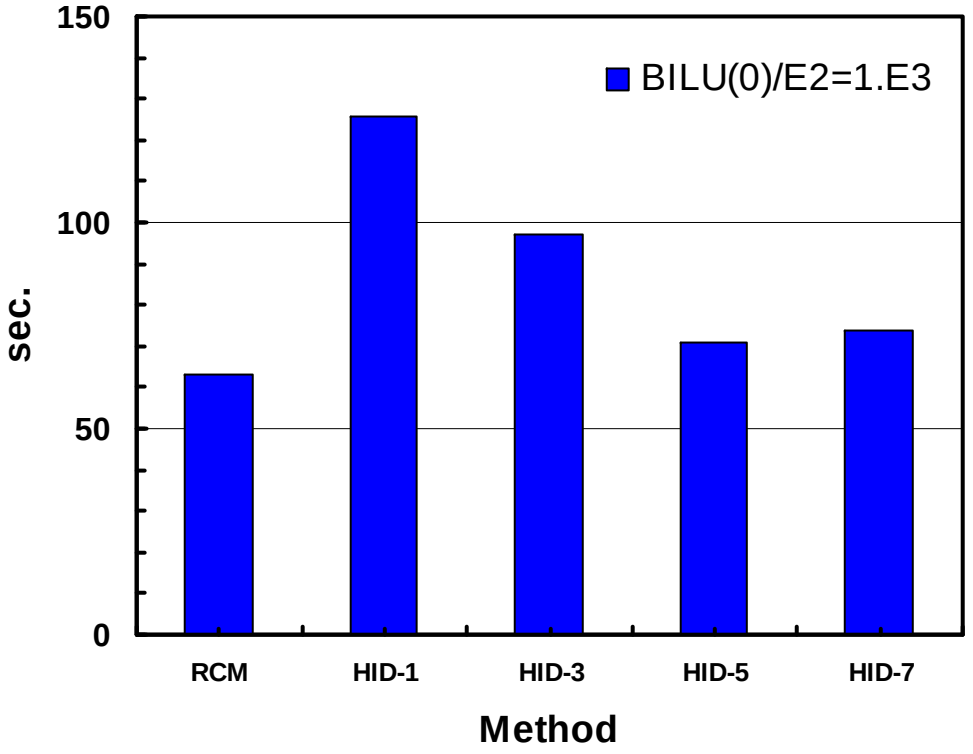
計算効率：T2K　1ノード HB16x1

BILU(0)/GPBiCGソルバー

（**不均質（$E_2=10^3$）**三次元弾性問題1.59M DOF）



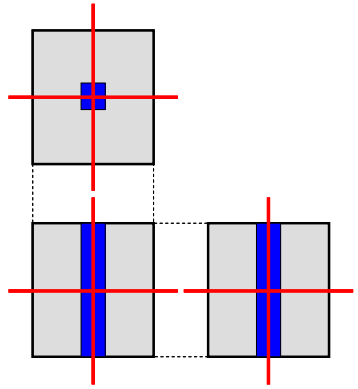


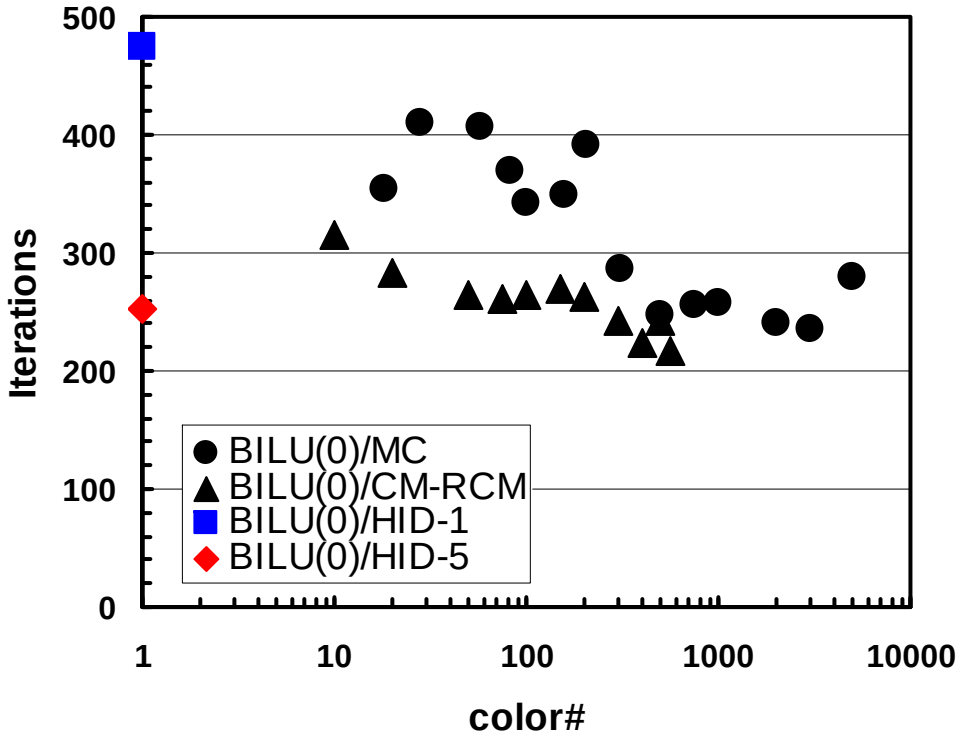


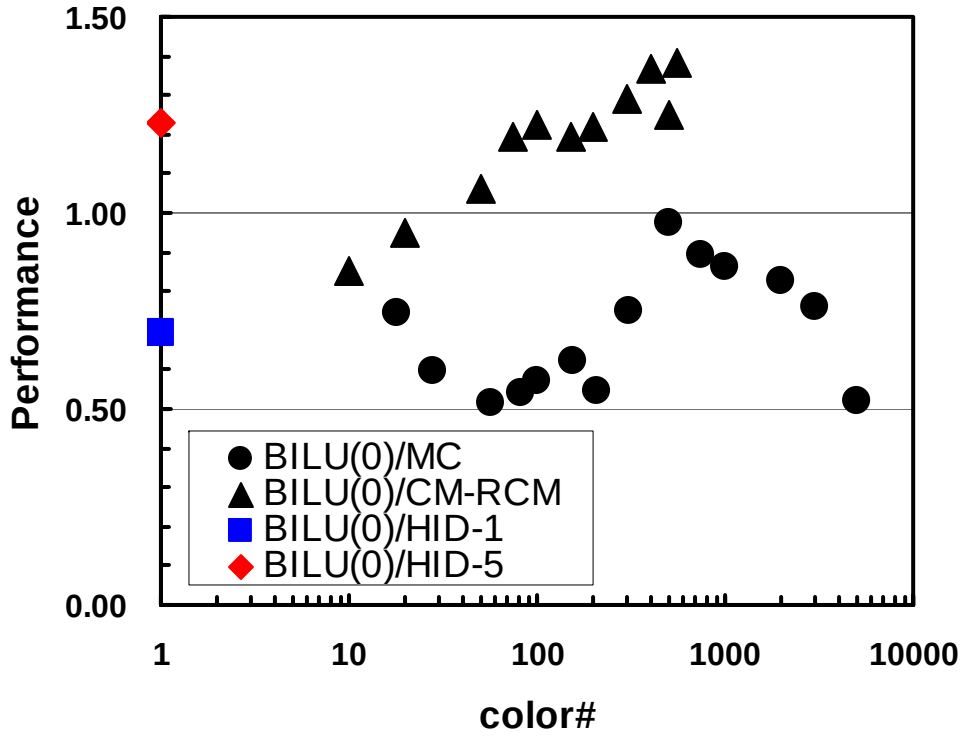
計算効率：T2K　1ノード HB16x1

BILU(0)/GPBiCGソルバー

（均質（$E_2=10^3$）三次元弾性問題1.59M DOF）



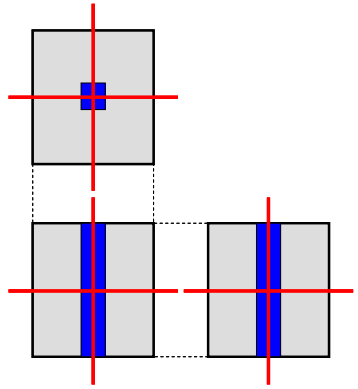


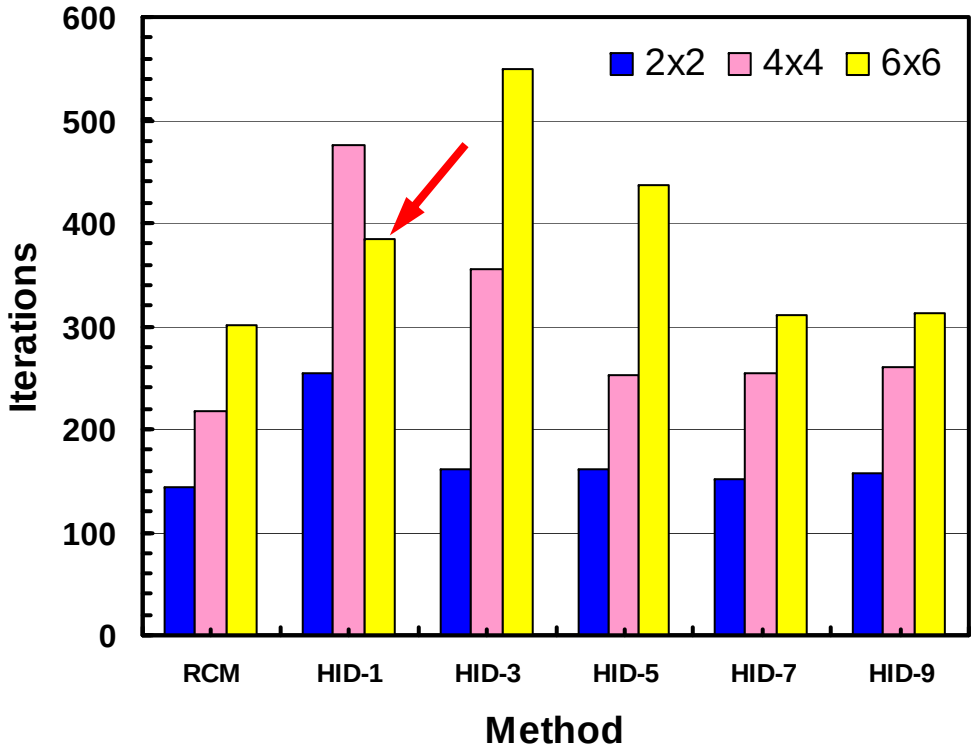


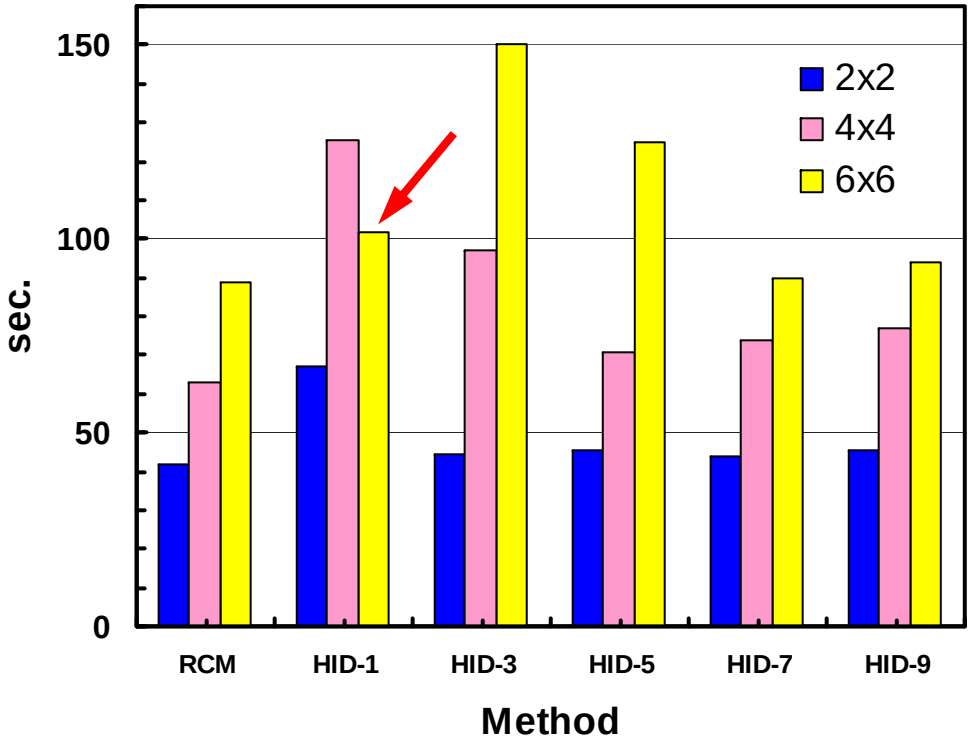
不均質部分の厚さの影響

BILU(0)/GPBiCGソルバー

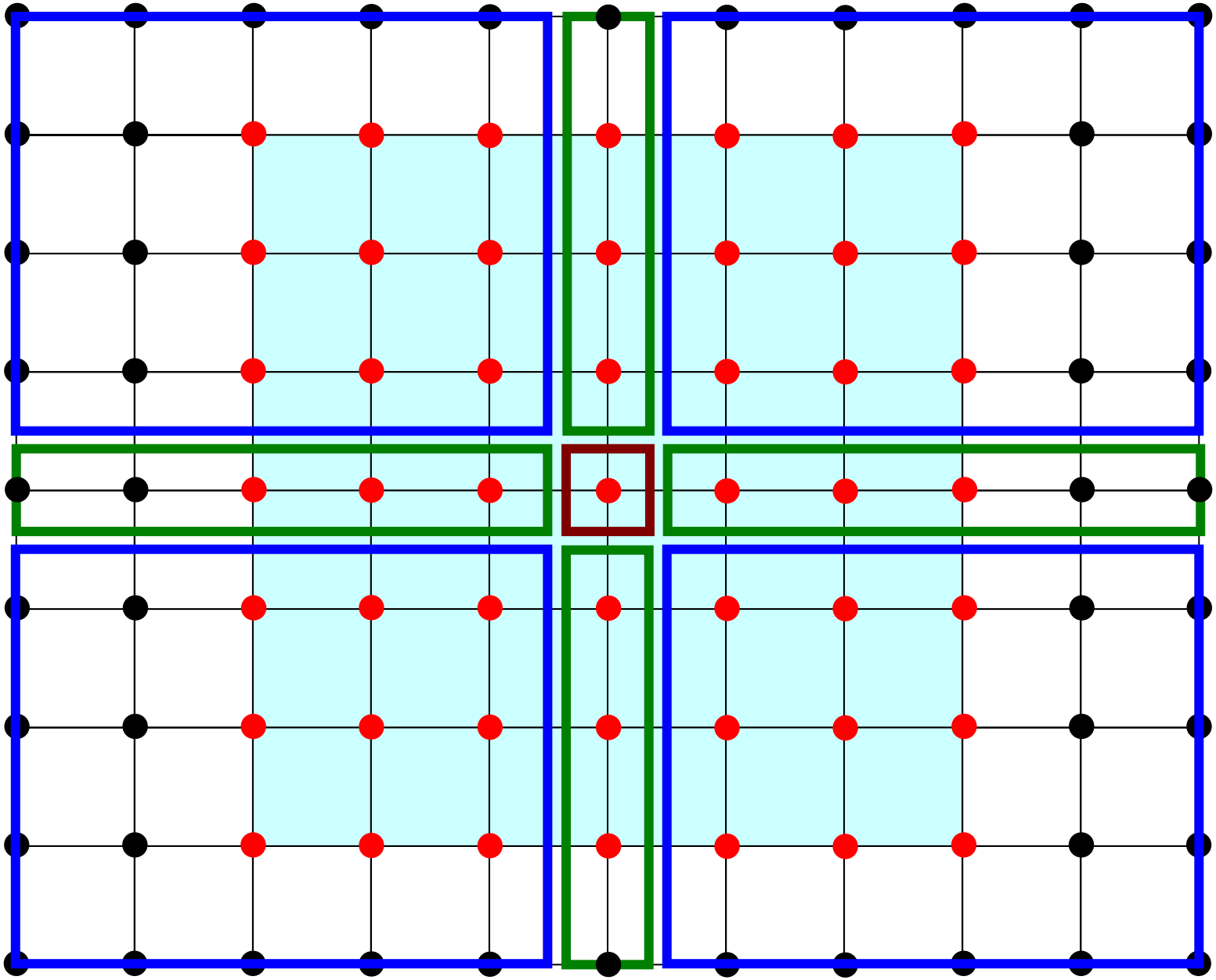
（不均質（$E_2=10^3$）三次元弾性問題1.59M DOF）







HID-1，6×6の場合



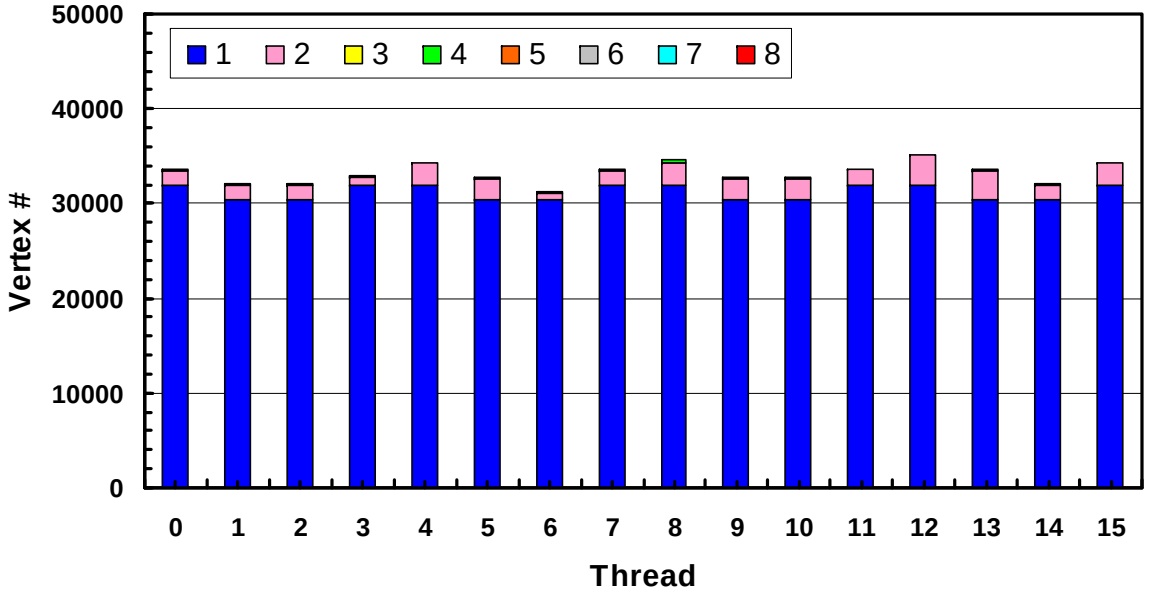
- 背景
  - ターゲットアプリケーション
  - T2Kオープンスパコン
  - Hybrid vs. Flat MPI
- ノード内並列化手法の実装
  - MC，CM-RCMリオーダリング
  - First Touch，再並び替え
- HIDの可能性
- HIDの改良
- まとめ

ノード内並列化手法

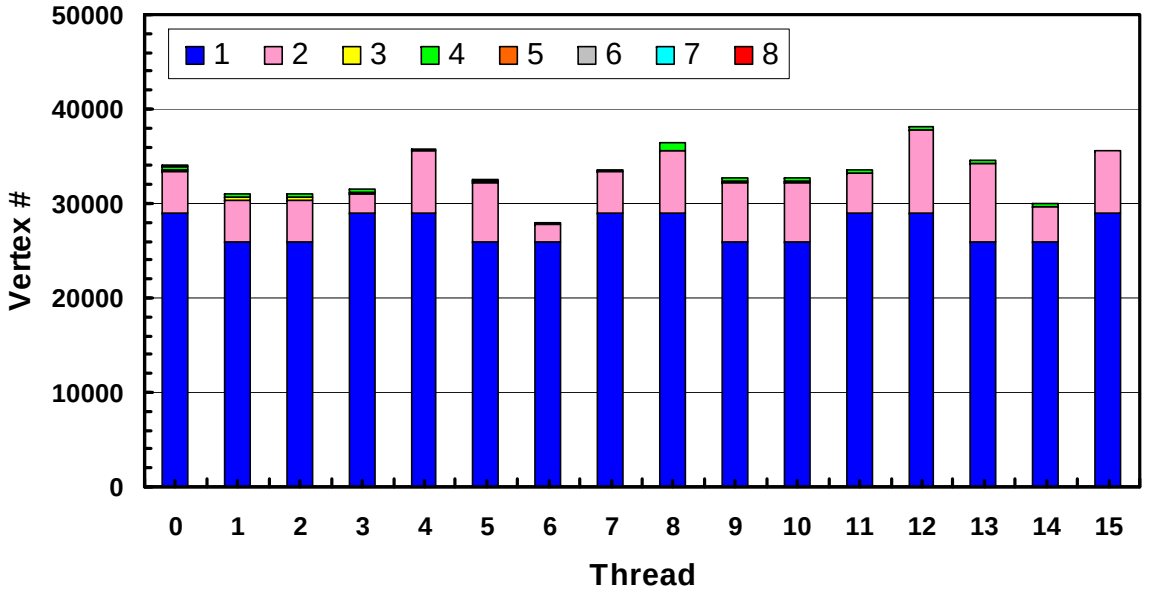
- RCM，CM-RCMの有効性
- HID
  - 悪条件問題の場合には注意が必要
    - コネクター（節点群）の構成
    - 計算順序
      - RCM+CM-RCMに匹敵する場合もある
  - Load Imbalance
- IntelligentなHID向け領域分割機能
  - グラフとしての情報
  - 要素特性の考慮
  - 選択的フィルイン，選択的オーバーラッピング〔KN 2007〕

各スレッド，レベルにおける節点数

HID-1

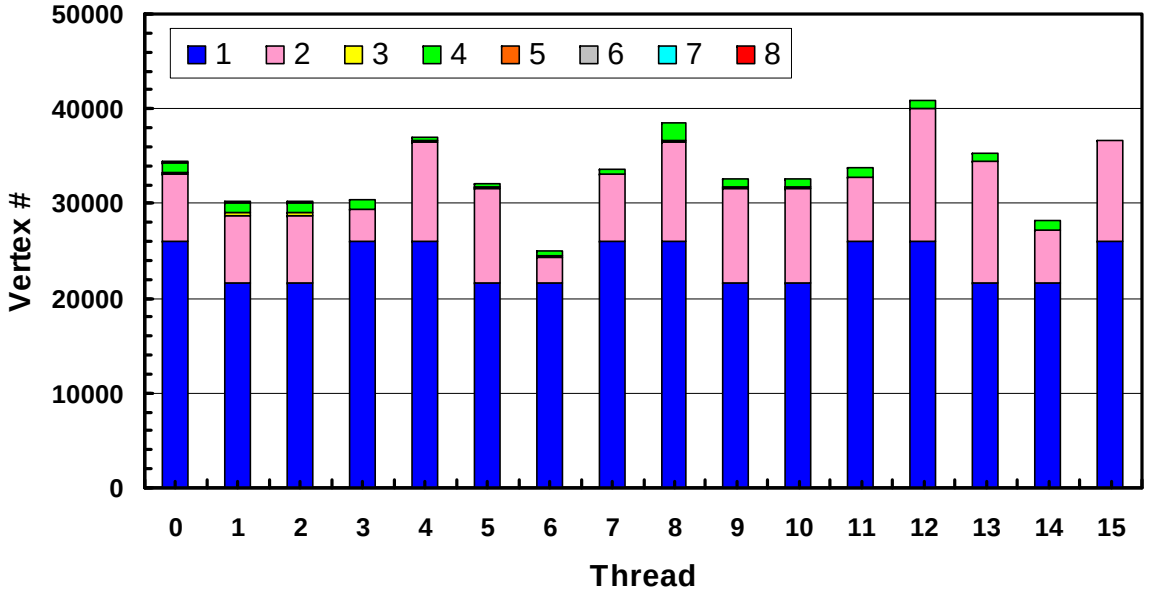


HID-3



各スレッド，レベルにおける節点数

HID-5



HID-7