東京大学
THE UNIVERSITY OF TOKYO

マルチコア時代の並列前処理手法

Parallel Preconditioning Methods for Iterative Solvers in Multi-Core Era

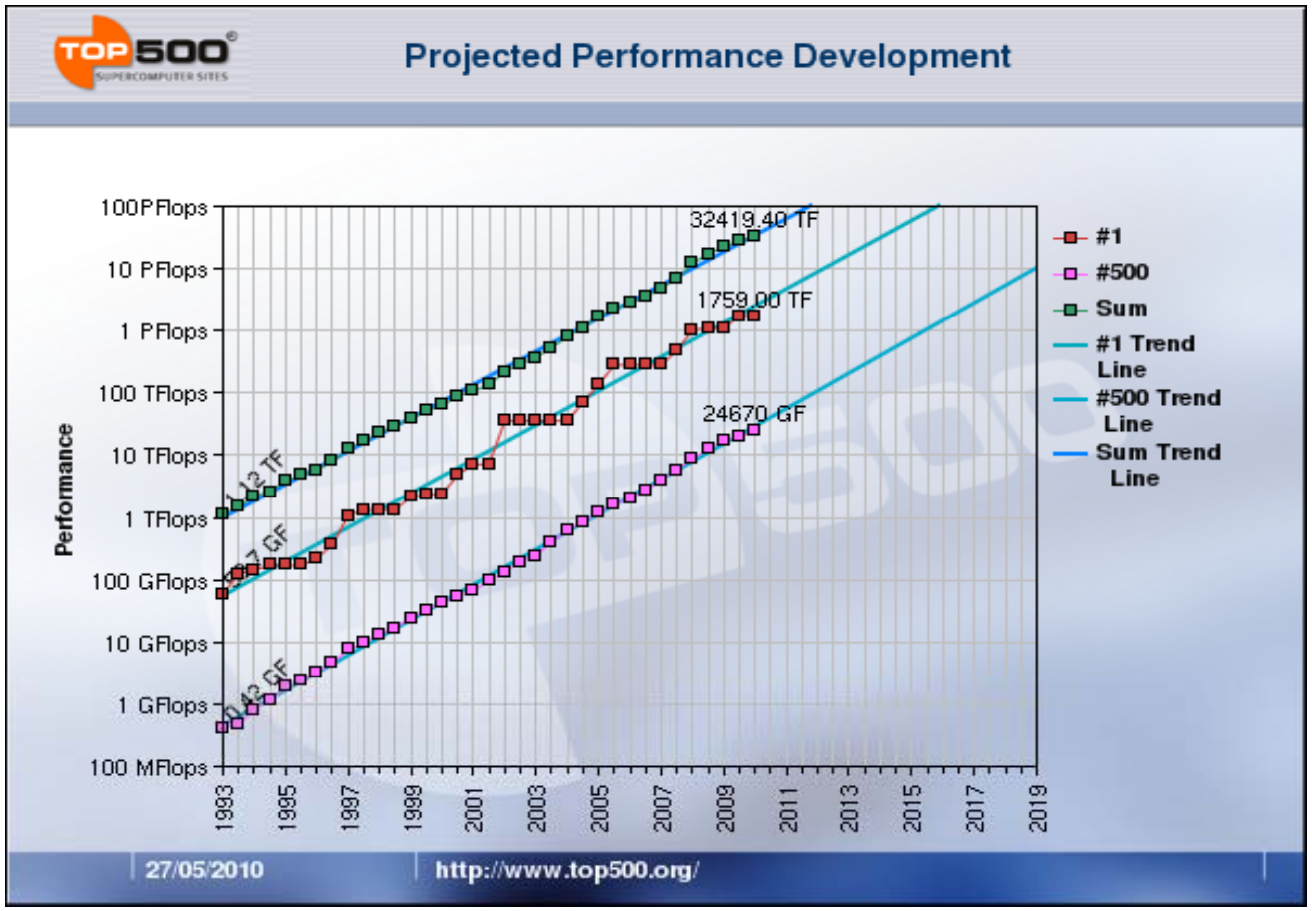
中島 研吾

東京大学情報基盤センター

2010年10月18日

京都大学数理解析研究所（RIMS）研究集会：科学技術計算アルゴリズムの数理的基盤と展開

We are now in Post-Peta-Scale Era



- PFLOPS: Peta (=10^{15}) Floating OPerations per Sec.
- Exa-FLOPS (=10^{18}) will be attained in 2018 or 2019

Exa-Scale Systems

- Peta-scale -> Evolution, Exa-scale -> Revolution
- 様々な技術的問題点（例）
  - $>10^8$ コア数を持つシステムの耐故障性（Fault Tolerance）
  - 電力消費量
    - 現状の最も効率的なシステム：2MW/PFLOPS（年2億円）
    - ExaFLOPS：2GW，年2,000億円→20MWにすることが必要
  - メモリーウォール問題
    - 現状Byte/Flop rate (B/F) 0.40 -> 0.10, 0.02？
- 汎用的システムは困難→分野間協力重要
  - H/W, S/W, Applications
  - 計算機科学，計算科学，数値アルゴリズム

IESP: International Exascale Software Project

- http://www.exascale.org/
- International Project
  - A single country cannot do that …
  - 4 Workshops since 2009
    - 5th is during October 18th-19th in Maui, HI, USA
- Current Status
  - Discussions on Road-map

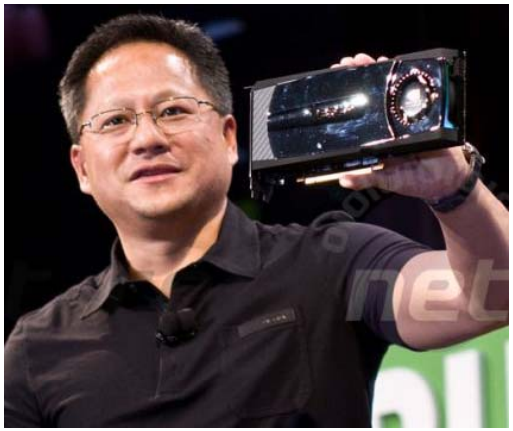
Key-Issues towards Appl./Algorithms on Exa-Scale Systems

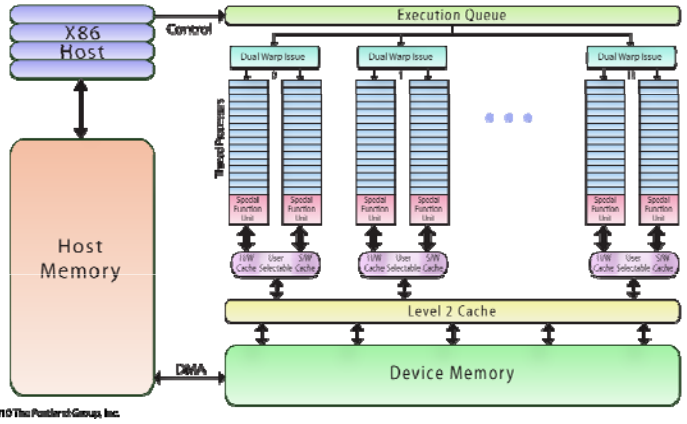
Jack Dongarra (ORNL/U. Tennessee) at SIAM/PP10
（日本応用数理学会誌Vol.20-3に関連記事）

- Hybrid/Heterogeneous Architecture
  - Multicore + GPU
  - Multicore + Manycore (more intelligent)
- Mixed Precision Computation
- Auto-Tuning/Self-Adapting
- Fault Tolerant
- Communication Reducing Algorithms

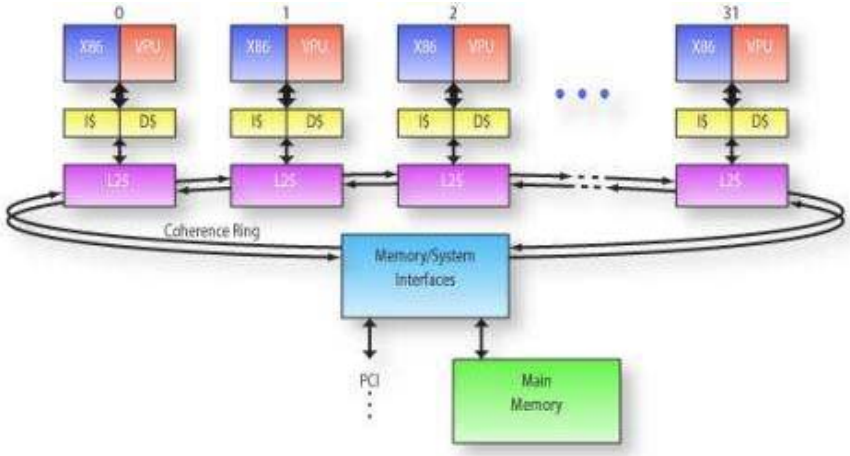
Heterogeneous Architecture by (CPU+GPU) or (CPU+Manycore) will be general in less than 5 years

NVIDIA Fermi





Intel Knights Ferry



CPU+Accelerator (GPU, Manycore)

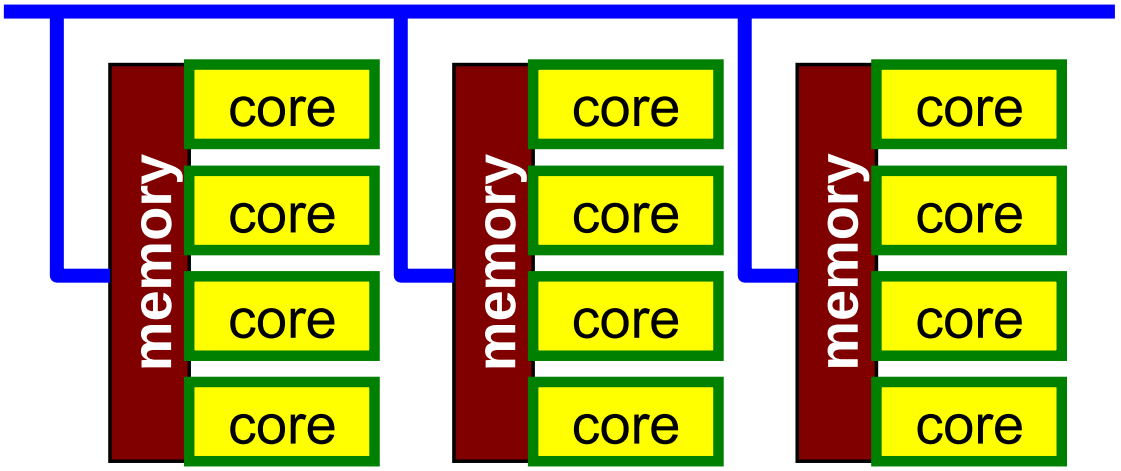
- 高いメモリーバンド幅
- 現状のGPUには様々な問題点
  - 通信：CPU-GPU/GPU-GPU
  - プログラミングの困難さ：CUDA，OpenCLは状況を変えつつあるが
  - 限定されたアプリケーションのみで高効率：陽的FDM，BEM
- メニーコア（Manycores）
  - Intel Many Integrated Core Architecture（MIC）
    - GPUより賢い：軽いOS，コンパイラが使える
  - “Intel Knights Ferry” with 32 cores is available soon for use on development of programming environment (very limited users)
  - “Knights Corner” with >50 cores (22nm) in 2012 or 2013 ?
- 近い将来
  - GPUとManycore（MIC的な意味での）は大差なくなる

Hybrid並列プログラミングモデルは必須

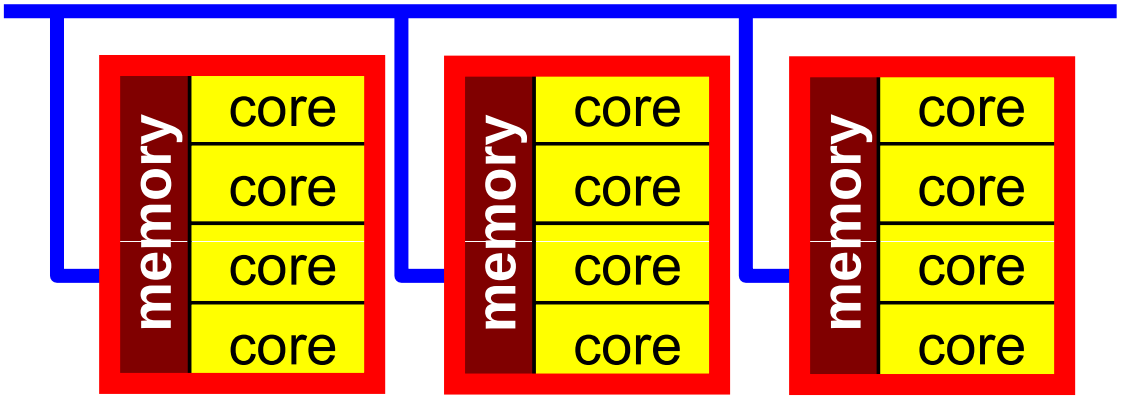
- Message Passing
  - MPI
- Multi Threading
  - OpenMP

Flat MPI vs. Hybrid

Flat-MPI：Each PE -> Independent



Hybrid：Hierarchal Structure



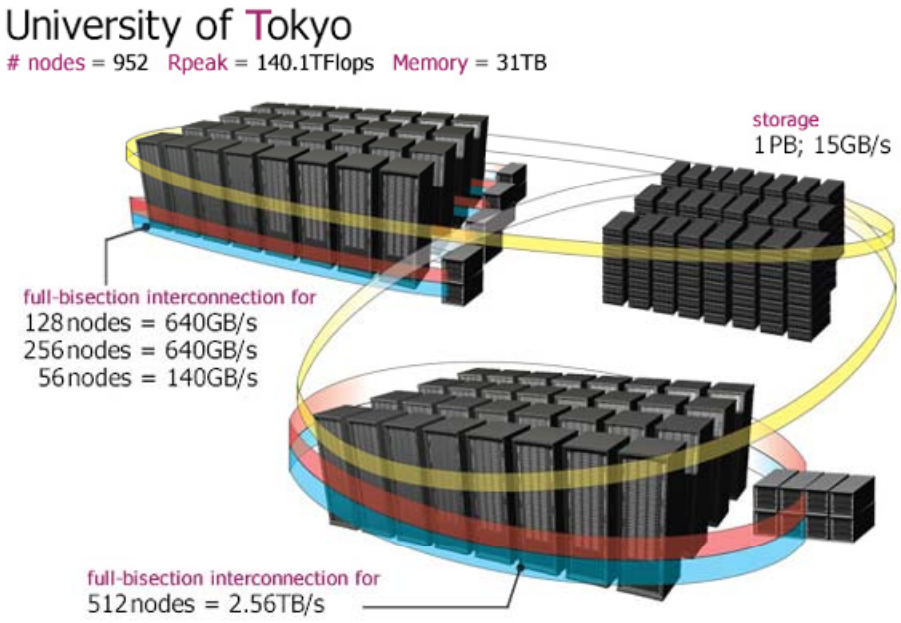
背　景

- T2Kオープンスパコン（東大）
- 並列多重格子法（Multigrid）前処理付きCG法
  - MGCG
- Flat MPI vs. Hybrid（OpenMP+MPI）
  - Hybrid
    - MPIのプロセス数を減らせる→通信オーバーヘッド減少
    - メモリ的には厳しくなる：特に疎行列ソルバー

T2K（東大）（1/2）

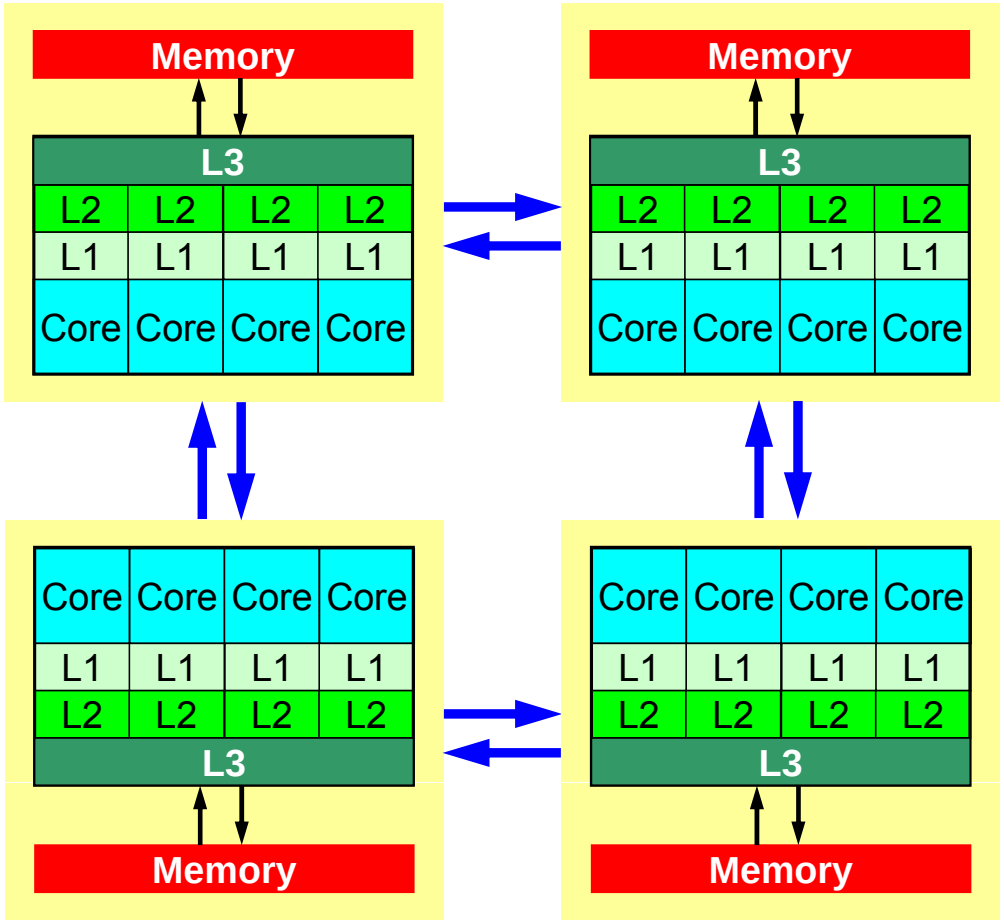
- T2Kオープンスパコン仕様
  - http://www.open-supercomputer.org/
  - 筑波大，東大，京大

- 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 53位（Jun 2010）

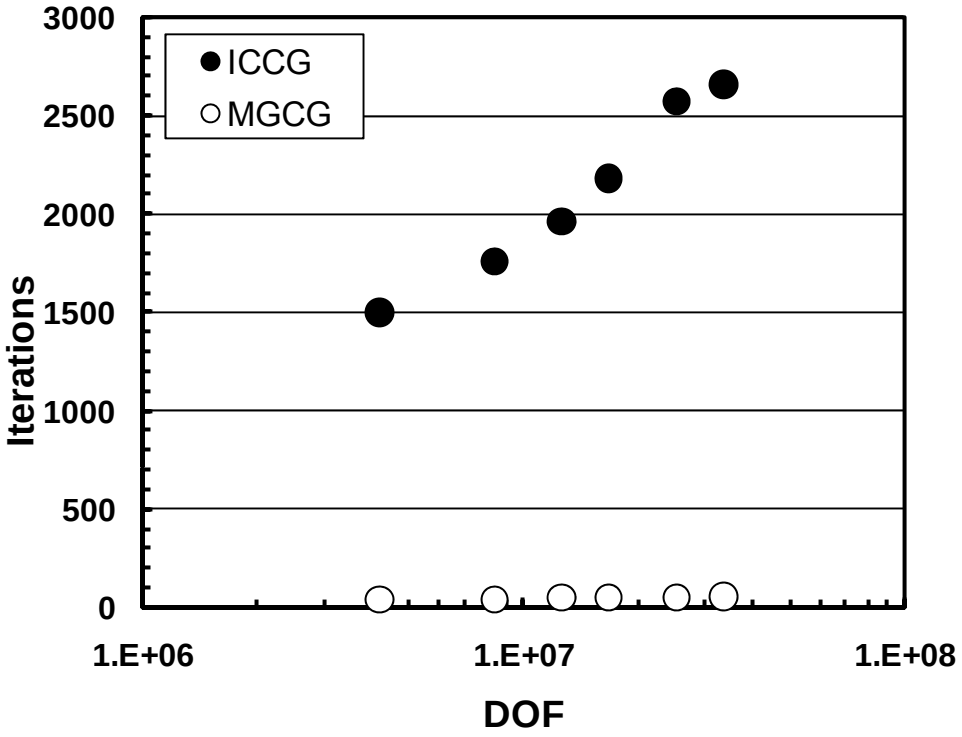


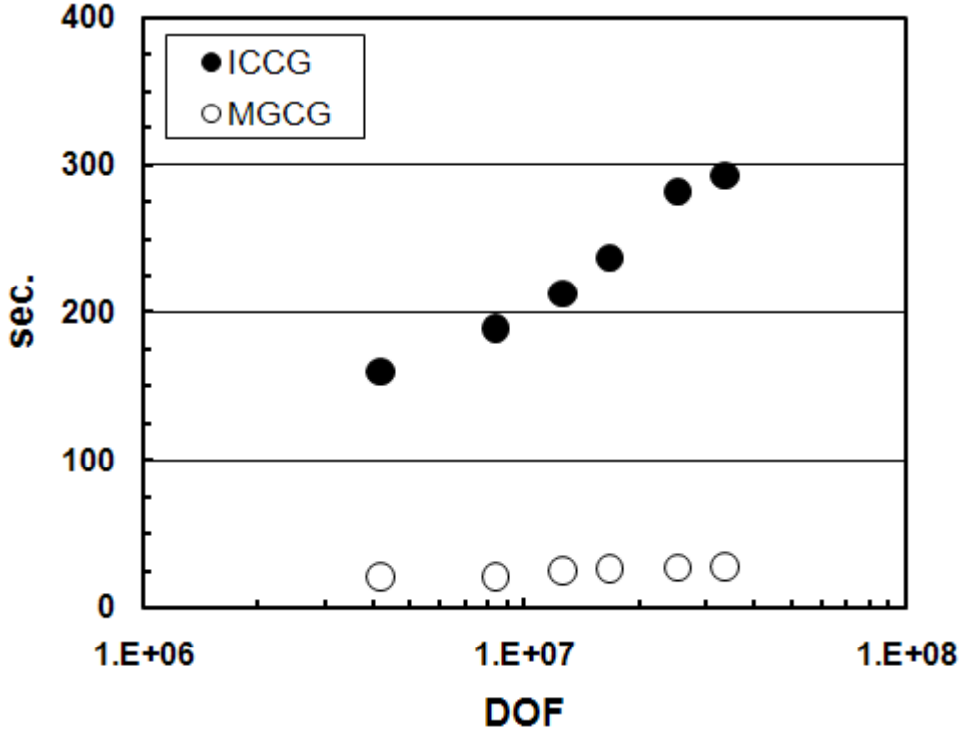
T2K（東大）（2/2）

- 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



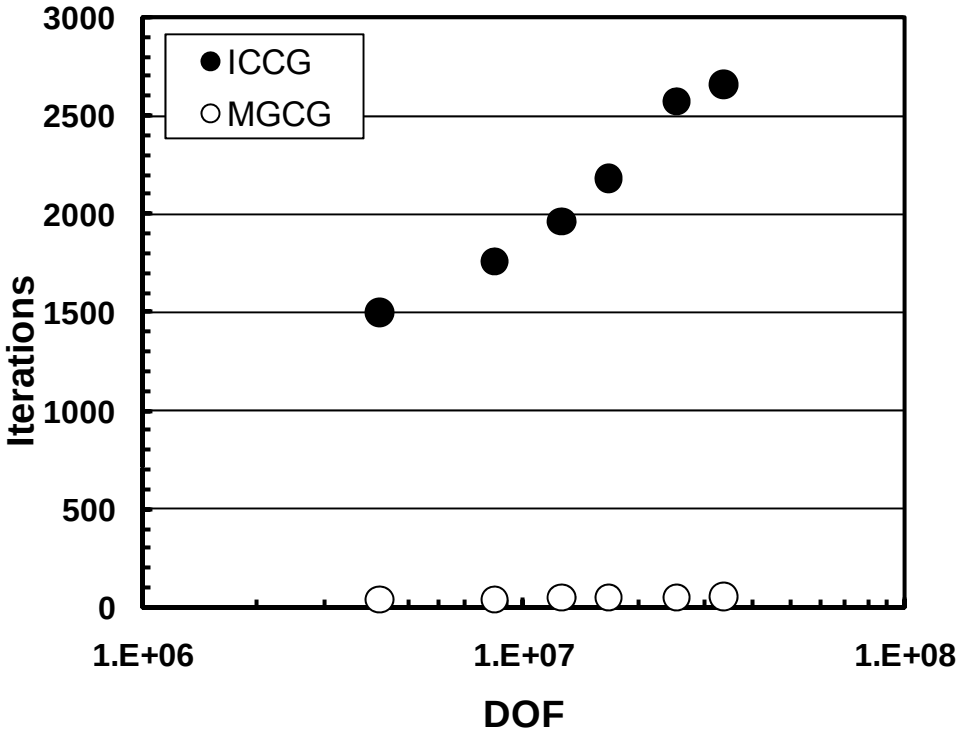
Multigrid is scalable
Weak Scaling: Problem Size/Core Fixed
三次元ポアソン方程式（一様）

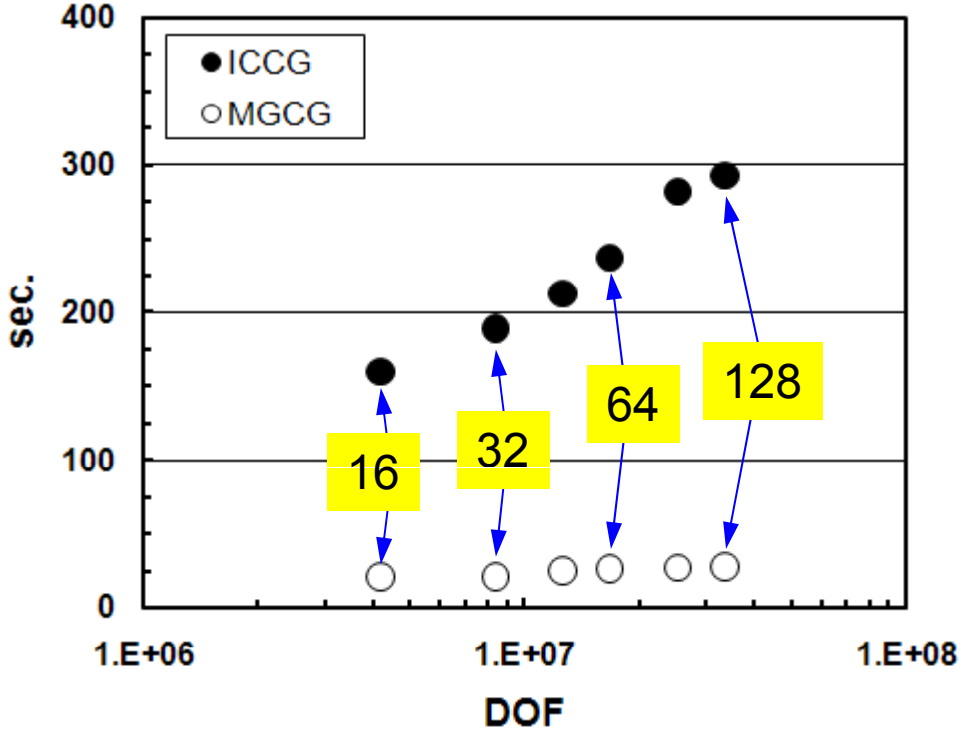




Multigrid is scalable
Weak Scaling: Problem Size/Core Fixed

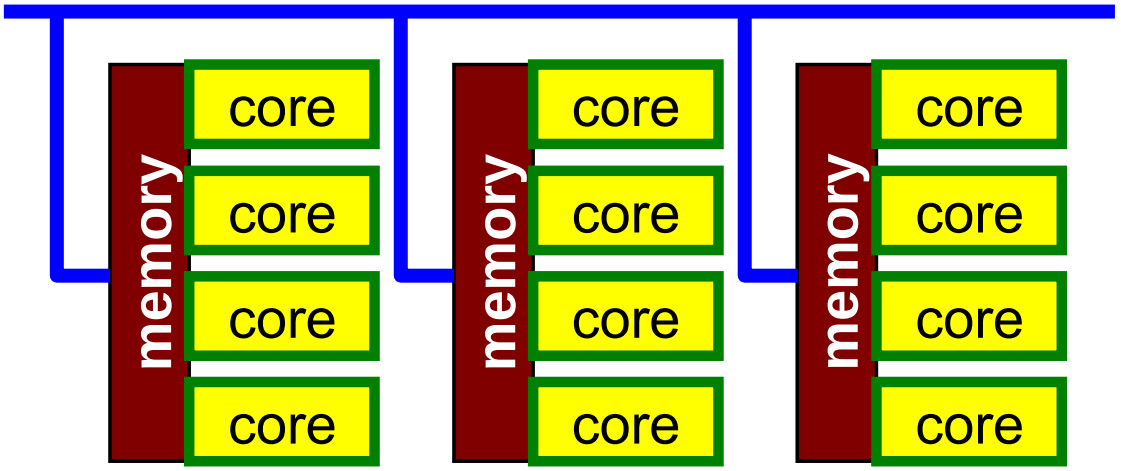
MGCG法の計算時間は「Weak Scaling」では一定＝Scalable



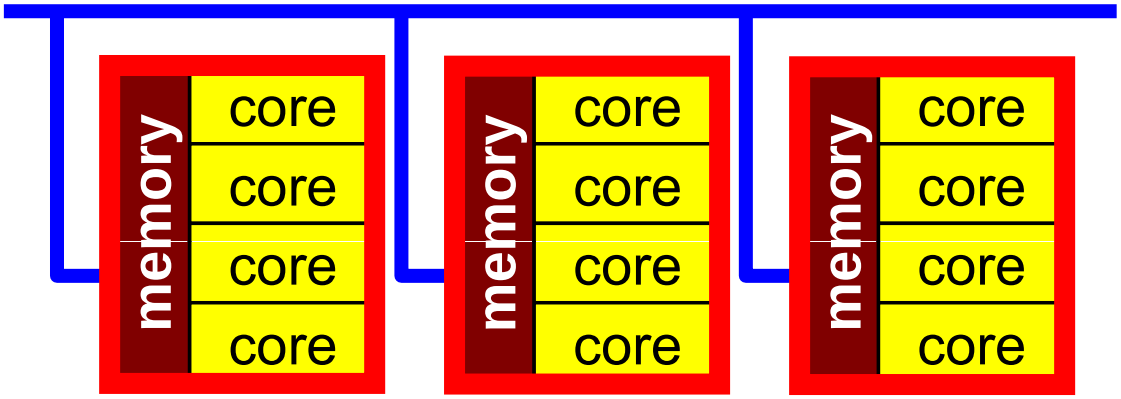


Flat MPI vs. Hybrid

Flat-MPI：Each PE -> Independent



Hybrid：Hierarchal Structure



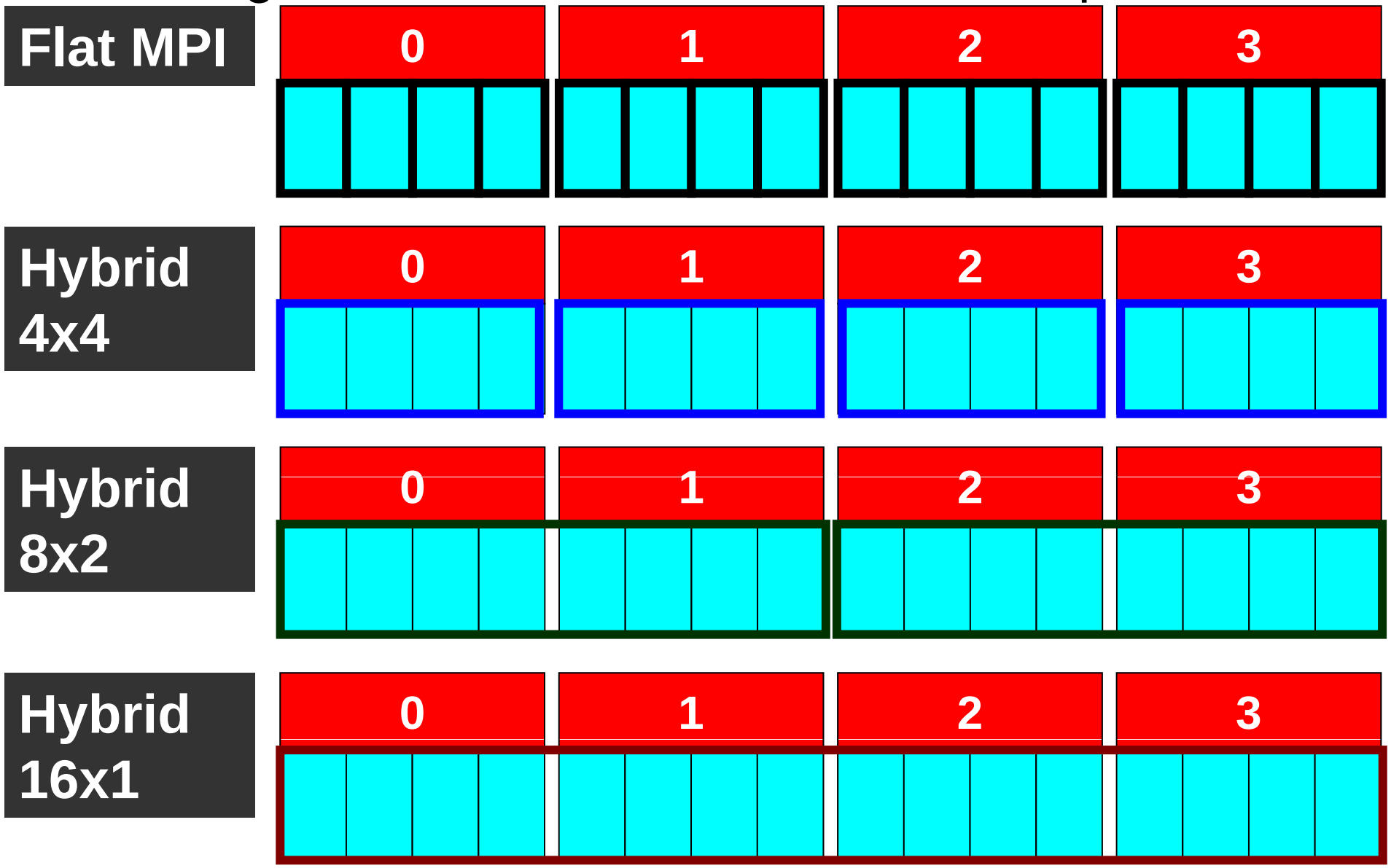
Flat MPI vs. Hybrid

- 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

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

Higher Performance of HB16x1 is important

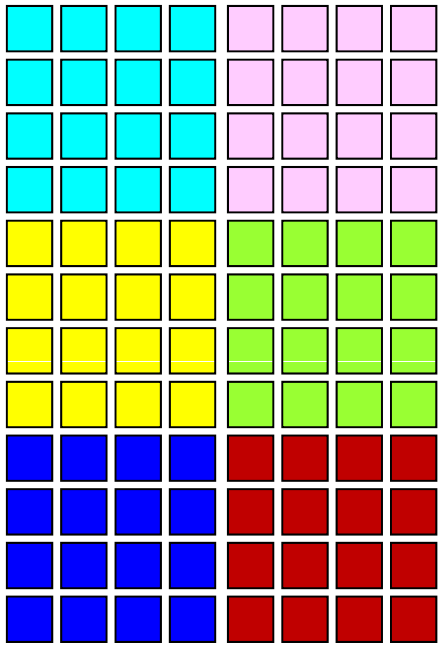


Domain Decomposition

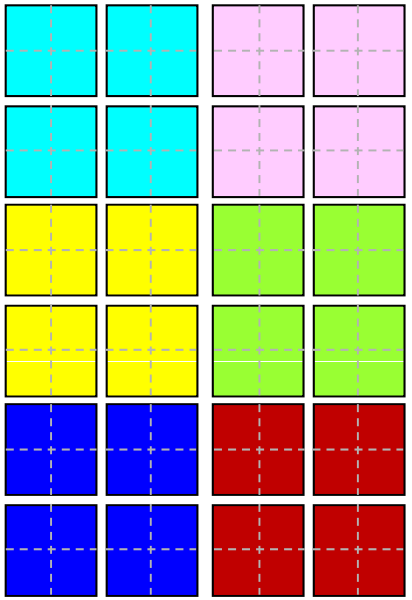
Inter Domain: MPI-Block Jacobi

Intra Domain: OpenMP-Threads (re-ordering)

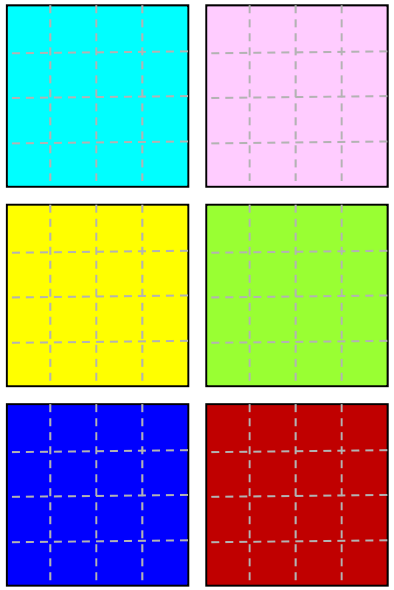
example: 6 nodes, 24 sockets, 96 cores



Flat MPI



HB 4x4



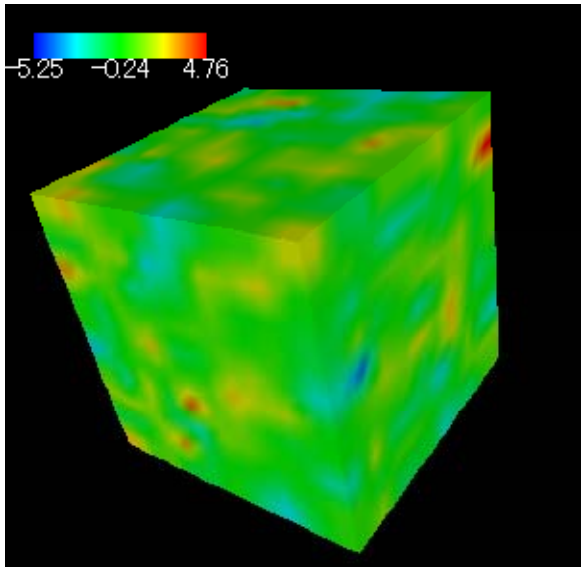
HB 16x1

解析対象

- 透水係数が空間的に分布する三次元地下水流れ
  - ポアソン方程式
  - 透水係数は地質統計学的手法によって決定〔Deutsch & Journel, 1998〕
- 規則正しい立方体ボクセルメッシュを使用した有限体積法
  - 局所細分化を考慮
- 周期的な不均質性: 128^3

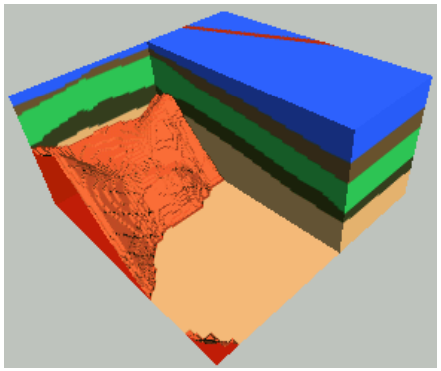
$$\frac{\partial}{\partial x}\left(\lambda\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\lambda\frac{\partial\phi}{\partial y}\right)+\frac{\partial}{\partial z}\left(\lambda\frac{\partial\phi}{\partial z}\right)=q$$

$$\phi=0@x=x_{\max}$$

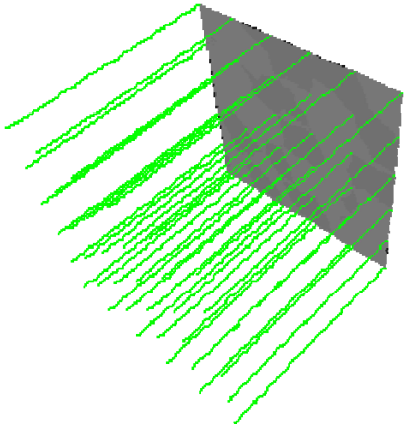


Groundwater Flow through Heterogeneous Porous Media

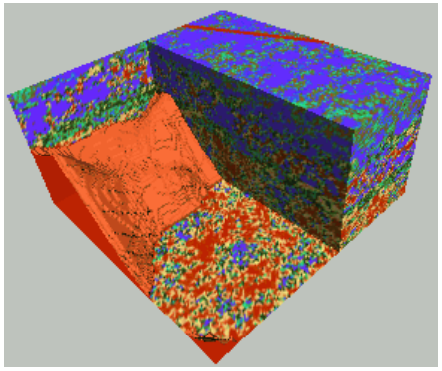
Homogeneous



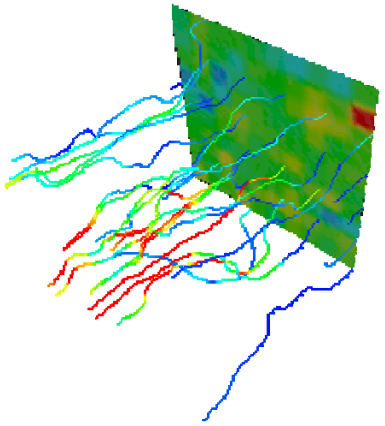
Uniform Flow Field



Heterogeneous

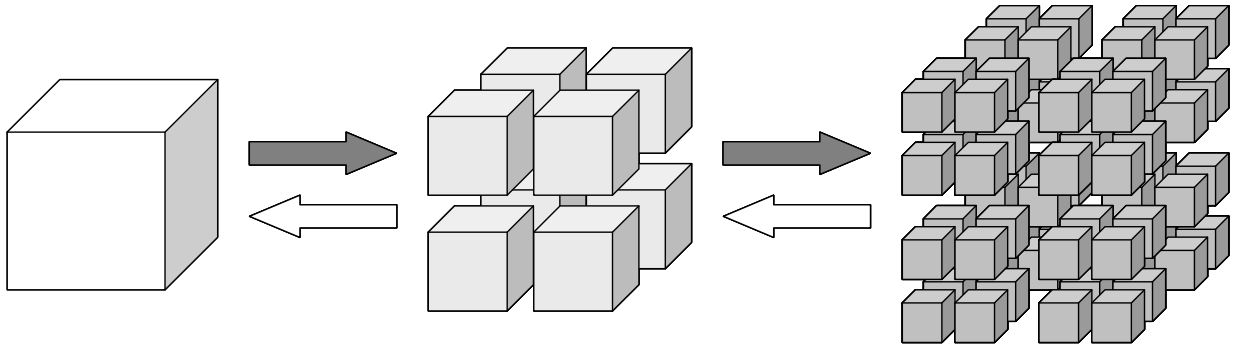


Random Flow Field



線形ソルバーの概要

- 前処理付きCG法
  - Multigrid 前処理
  - IC(0) for Smoothing Operator (Smoother)
  - Additive Schwartz Domain Decomposition
- 並列(幾何学的)多重格子法
  - 当方的な8分木
  - V-cycle
  - 領域分割型:Block-Jacobi局所前処理，階層型領域間通信
  - 最も粗い格子(格子数＝プロセッサ数)は1コアで実施



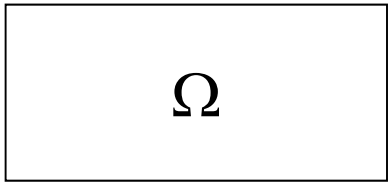
IC(0) as smoother of Multigrid

- IC(0) is generally more robust than GS.
- IC(0) smoother with Additive Schwartz Domain Decomposition (ASDD) provides robust convergence and scalable performance of parallel computation, even for ill-conditioned problems〔KN 2002〕.

Overlapped Additive Schwartz Domain Decomposition Method
for Stabilizing Localized Preconditioning

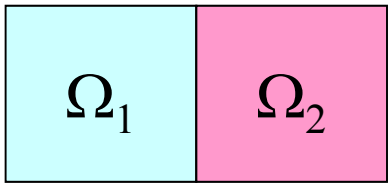
Global Operation

$$Mz = r$$



Local Operation

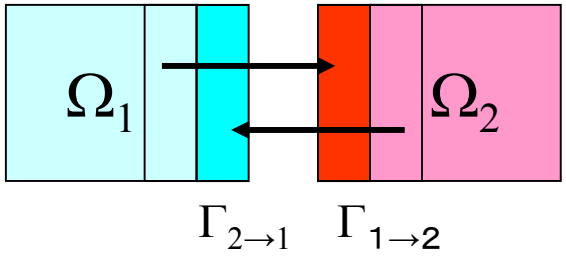
$$M_{\Omega_1} z^n_{\Omega_1} = r_{\Omega_1}, \quad M_{\Omega_2} z^n_{\Omega_2} = r_{\Omega_2}$$



Global Nesting Correction

$$z^n_{\Omega_1} \leftarrow z^{n-1}_{\Omega_1} + M^{-1}_{\Omega_1}\left(r_{\Omega_1} - M_{\Omega_1} z^{n-1}_{\Omega_1} - M_{\Gamma_{2\to1}} z^{n-1}_{\Gamma_{2\to1}}\right)$$

$$z^n_{\Omega_2} \leftarrow z^{n-1}_{\Omega_2} + M^{-1}_{\Omega_2}\left(r_{\Omega_2} - M_{\Omega_2} z^{n-1}_{\Omega_2} - M_{\Gamma_{1\to2}} z^{n-1}_{\Gamma_{1\to2}}\right)$$



Hardware/Software

- T2K/Tokyo
  - up to 512 nodes (8,192 cores)
- Program
  - Hitachi FORTRAN90 + MPI
  - CRS matrix storage
  - CM-RCM Reordering for OpenMP
- $|Ax-b|/|b|=10^{-12}$ for Convergence
- 不均質性
  - 最大最小透水係数の比 $= 10^{10}$ (10^{-5}~10^{+5})
- Multigrid Cycles
  - 1 V-cycle/iteration
  - 2 smoothing iterations for restriction/prolongation at every level
  - 1 ASDD iteration cycle for each resrtiction/prolongation

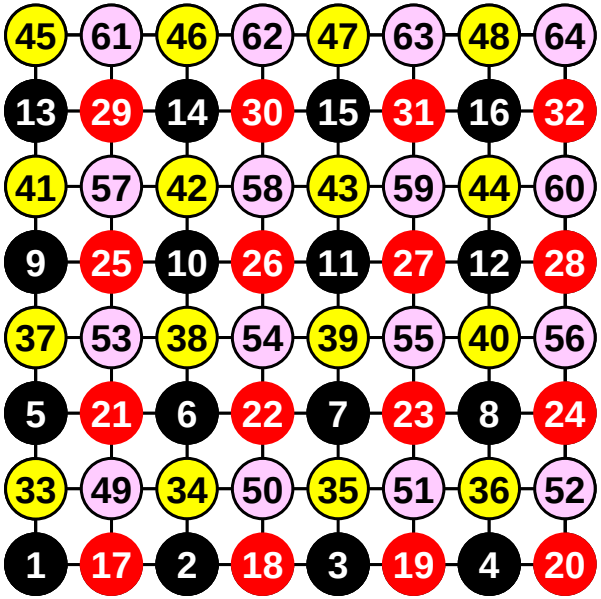
```
for (i=0; i<N; i++) {
  for (k=Index(i-1); k<Index(i); k++{
    Y[i]= Y[i] + A [k]*X[Item[k]];
  }
}
```

前処理付き反復法のSMP/MulticoreでのOpenMPによる並列化

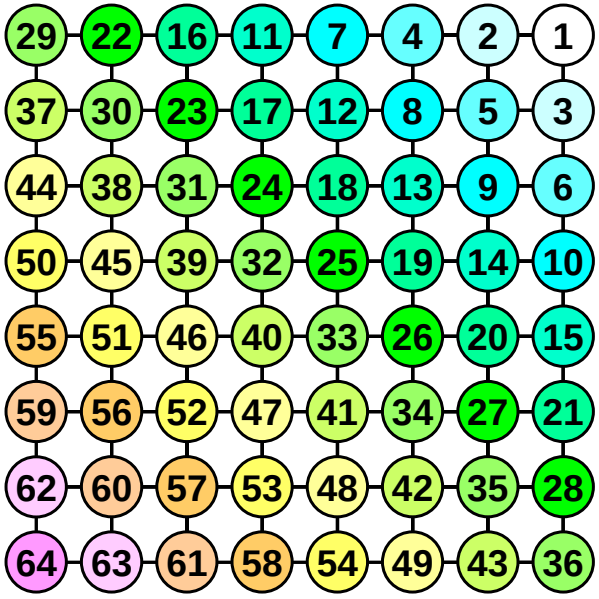
- DAXPY, SMVP, Dot Products
  - 簡単
- 前処理：ILU系分解，前進後退代入
  - 大域的な依存性（Global dependency）
  - 並び替え（Reordering）による並列性の抽出
    - Multicolor Ordering (MC), Reverse-Cuthill-Mckee (RCM)
    - 同じ色内の要素は独立⇒並列化可能
  - 「地球シミュレータ」向け最適化　[KN 2002,2003]
    - 並列及びベクトル性能
  - 並列性高く安定なCM-RCMを採用

Ordering Methods

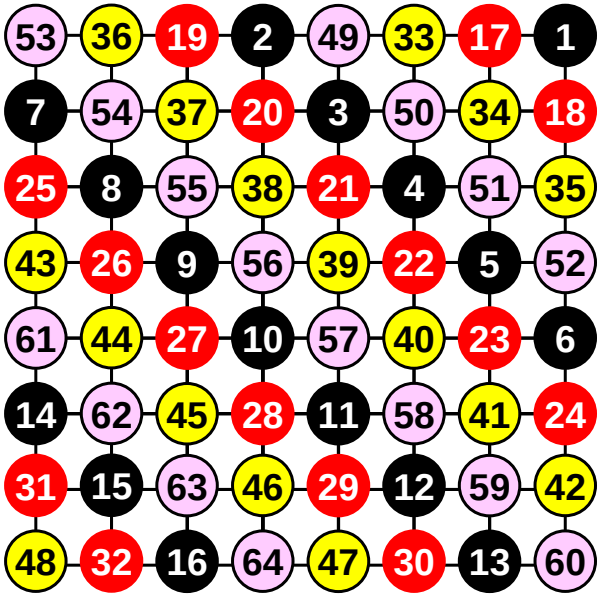
Elements in “same color” are independent: to be parallelized



MC (Color#=4)
Multicoloring



RCM
Reverse Cuthill-Mckee



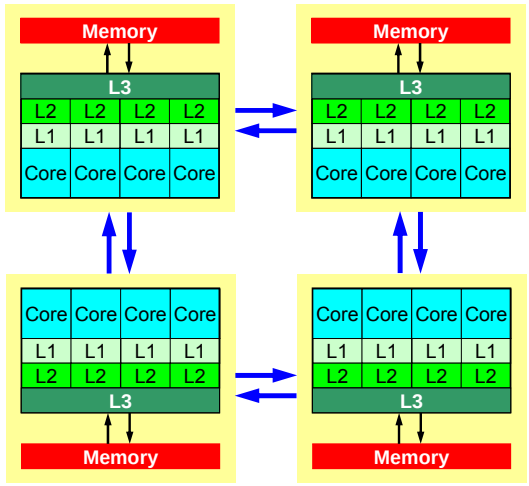
CM-RCM (Color#=4)
Cyclic MC + RCM

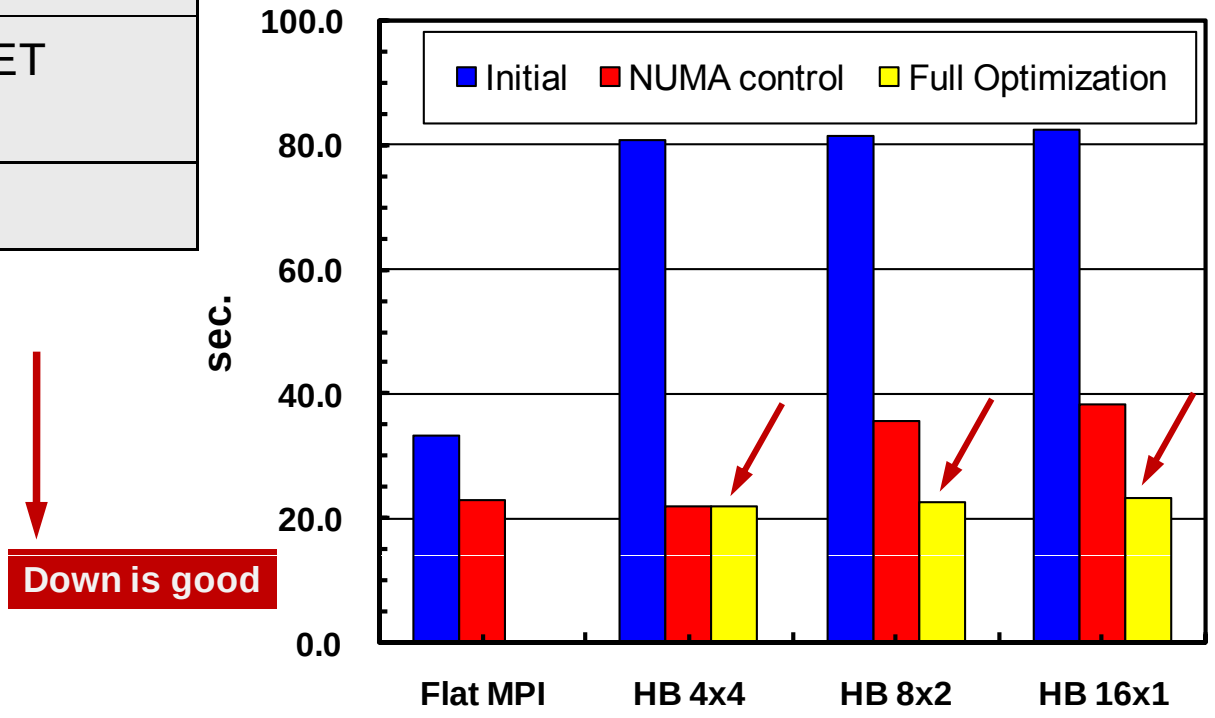
Effect of Optimization

- 64 cores (4 nodes) of T2K/Tokyo
  - 64^3 cells/core
  - 16,777,216 cells
- Full Optimization
  - NUMA Control
  - First Touch Data Placement
  - Further Reordering (with Contiguous/Sequential Memory Access)

NUMA control

Policy ID	Command line switches
0	no command line switches
1	`--cpunodebind=$SOCKET --interleave=all`
2	`--cpunodebind=$SOCKET --interleave=$SOCKET`
3	`--cpunodebind=$SOCKET --membind=$SOCKET`
4	`--cpunodebind=$SOCKET --localalloc`
5	`--localalloc`



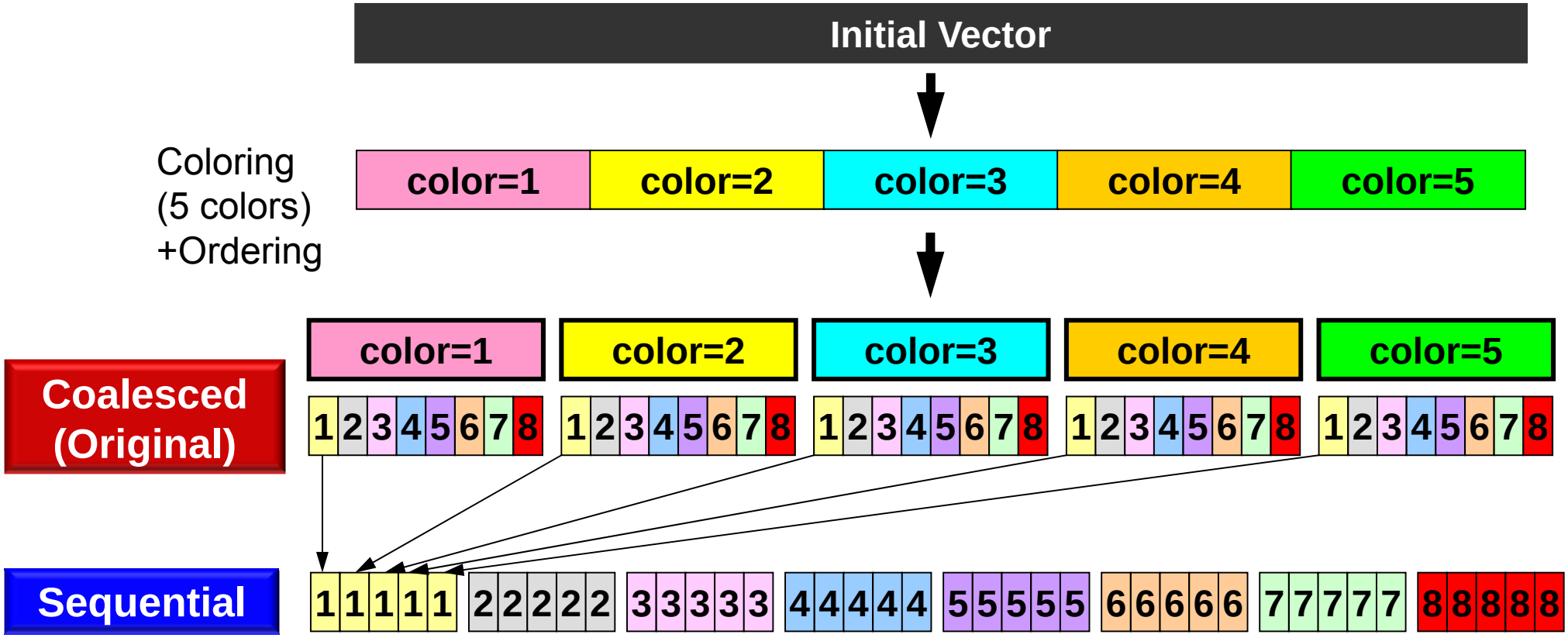


First Touch Data Placement

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

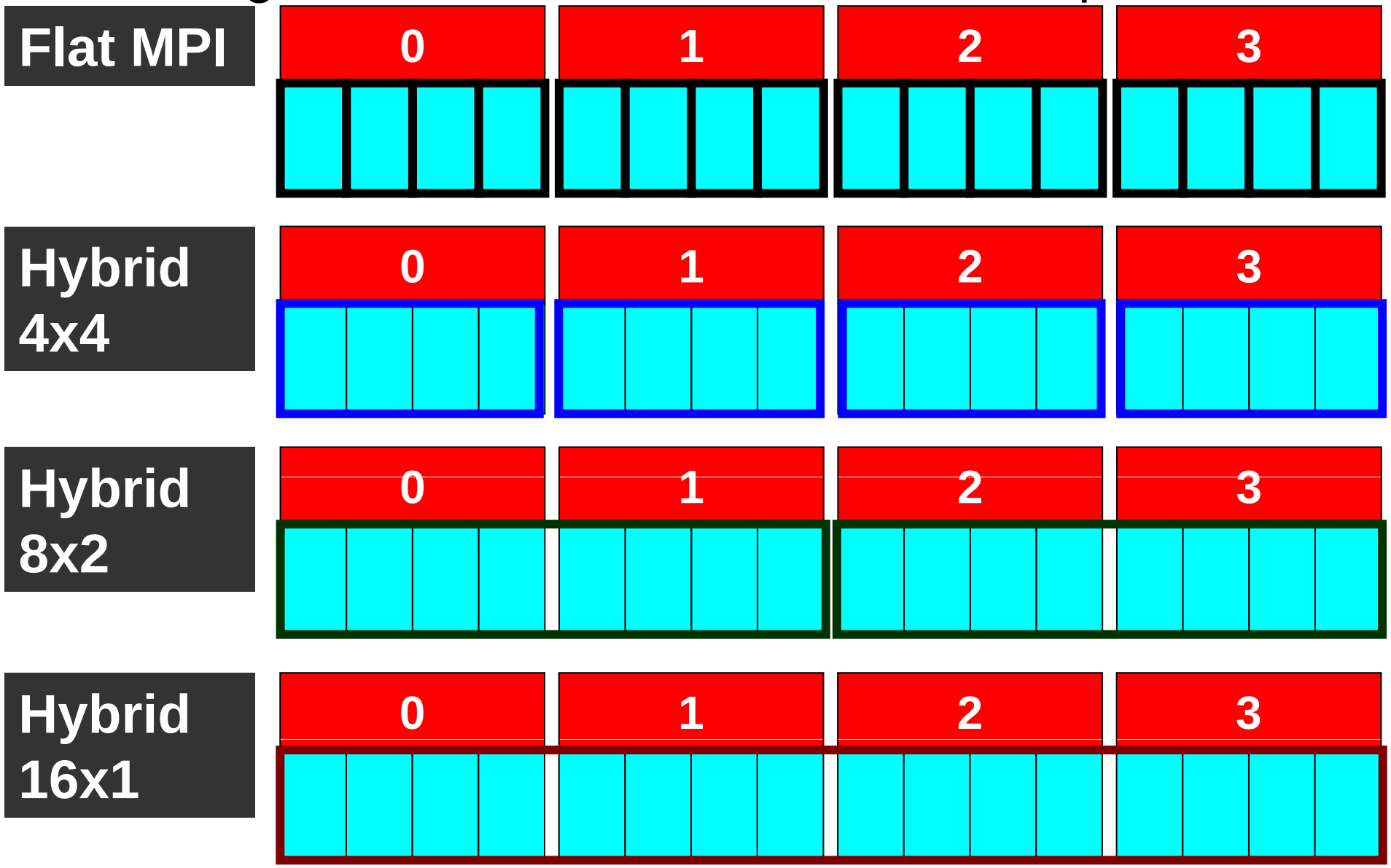
```
  do lev= 1, LEVELtot
    do ic= 1, COLORtot(lev)
!$omp parallel do private(ip,i,j,isL,ieL,isU,ieU)
      do ip= 1, PEsmpTOT
      do i = STACKmc(ip,ic-1,lev)+1, STACKmc(ip,ic,lev)
        RHS(i)= 0.d0; X(i)= 0.d0; D(i)= 0.d0

        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
```

Further Re-Ordering for Continuous Memory Access: Sequential 5 colors, 8 threads



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

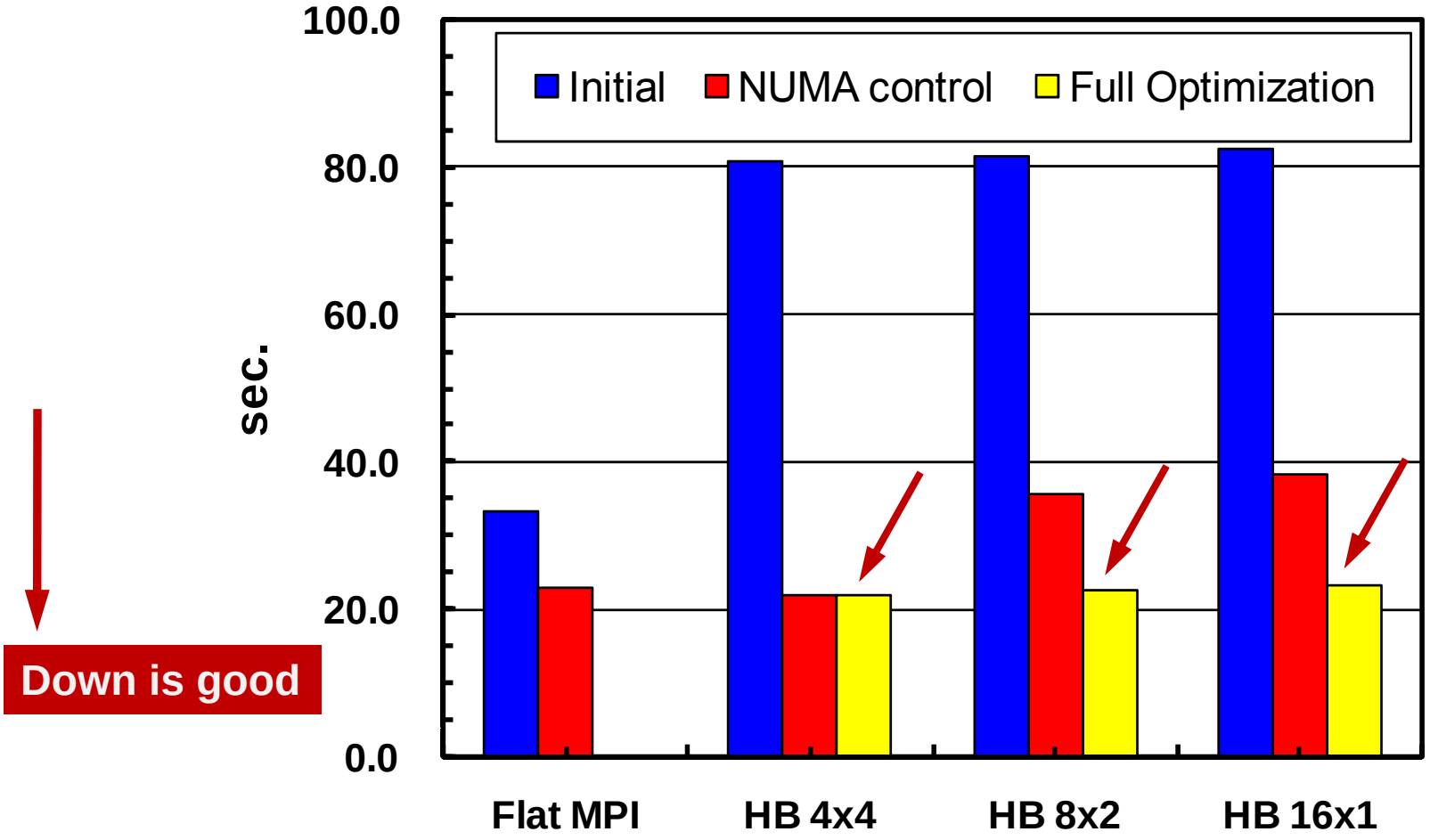
Higher Performance of HB16x1 is important



Effect of F.T. + Sequential Data Access

16,777,216= 64x64^3 cells, 64 cores, CM-RCM(2)

Time for Linear Solvers

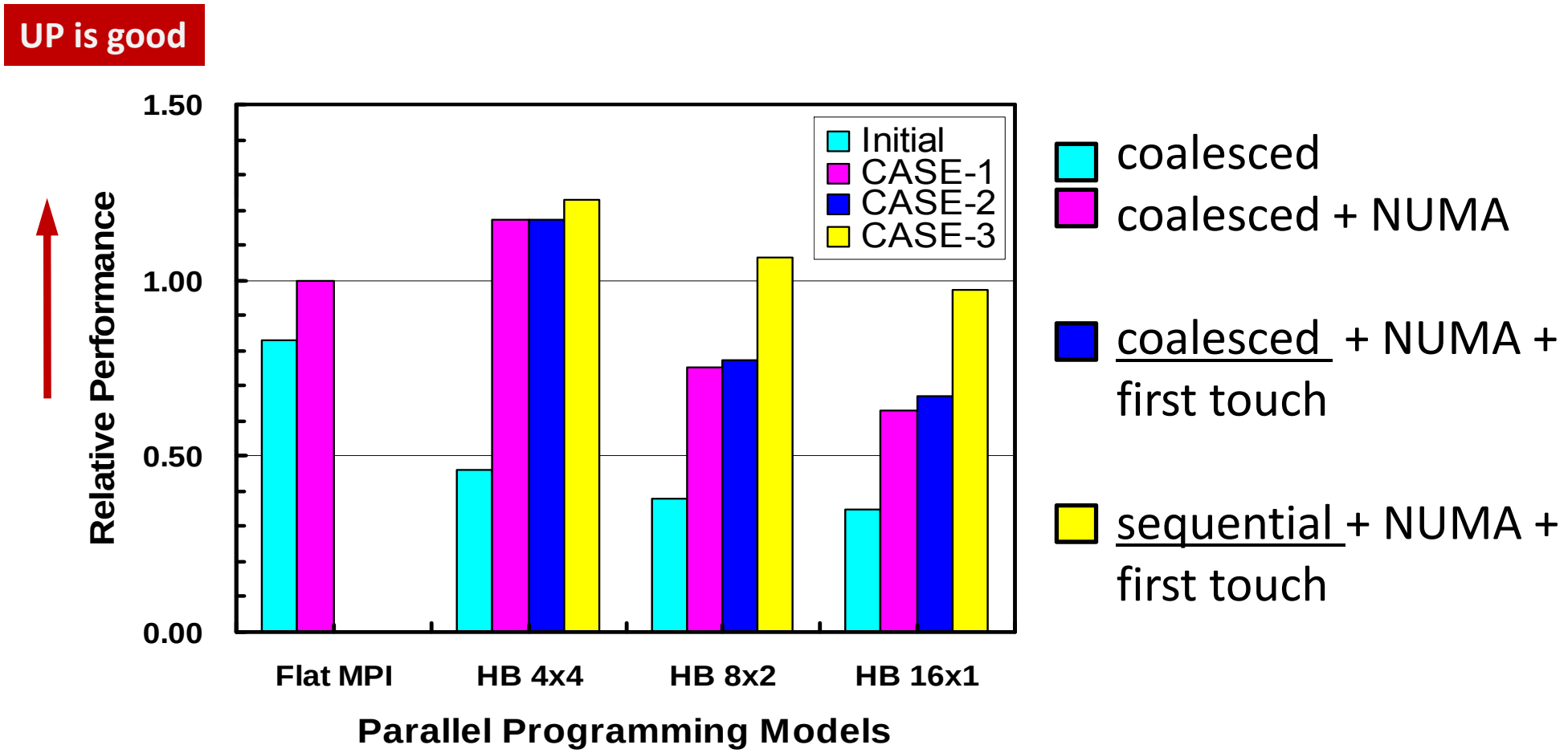


Effect of F.T. + Sequential Data Access

128^3 tri-linear hexahedral elements, 6,291,456 DOF

ICCG Solvers for 3D Linear-Elastic Eqn's, 32 nodes of T2K (512 cores)

Time for Linear Solvers, HB 4x4 is the fastest



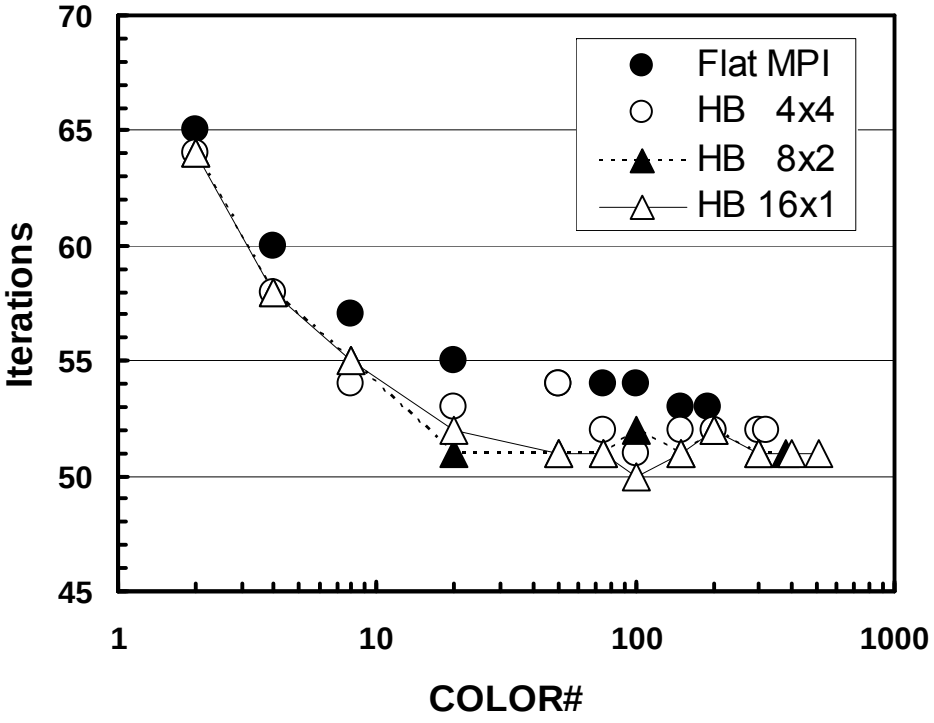
Effect of Number of Colors

色数の効果（CM-RCM）

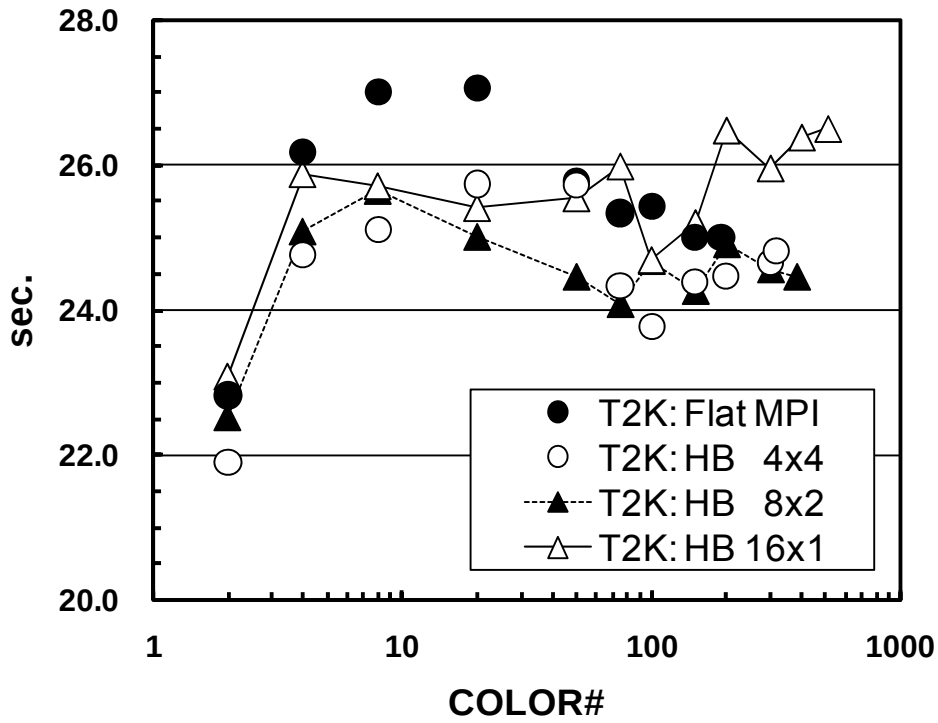
16,777,216= 64x64^3 cells, 64 cores

色数が増えると収束は改善，計算時間は
CM-RCM(2)が最も短い

Iterations



sec.



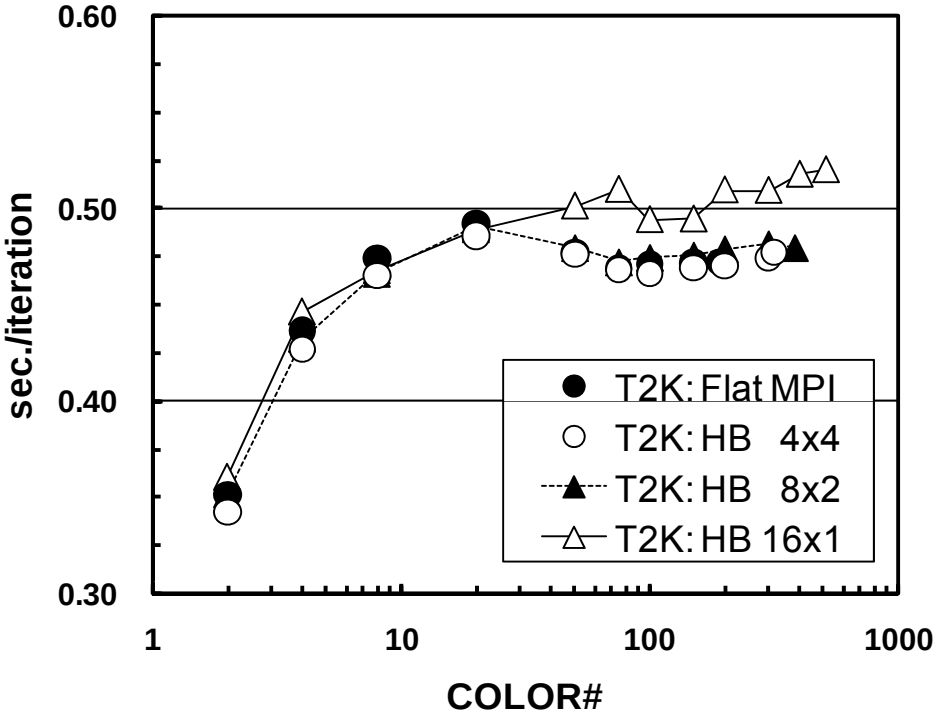
色数の効果（CM-RCM）

16,777,216= 64x64^3 cells, 64 cores

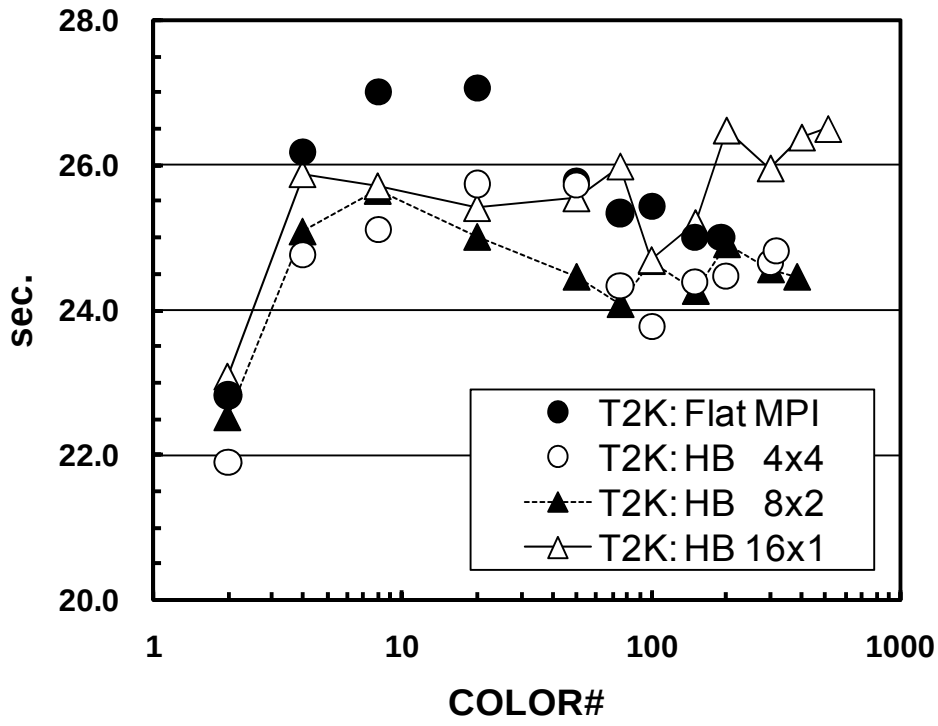
色数が増えると収束は改善，計算時間は

CM-RCM(2)が最も短い：反復あたり計算時間短い

sec./iter



sec.



色数の効果（CM-RCM）

RCM：前進後退代入時に変数値が変わるため，キャッシュラインからメモリに戻されてしまう可能性がある

29	22	16	11	7	4	2	1
37	30	23	17	12	8	5	3
44	38	31	24	18	13	9	6
50	45	39	32	25	19	14	10
55	51	46	40	33	26	20	15
59	56	52	47	41	34	27	21
62	**60**	57	53	48	42	35	28
64	63	**61**	58	54	49	43	36

RCM

45	10	39	5	35	2	33	1
17	46	11	40	6	36	3	34
53	18	47	12	41	7	37	4
24	54	19	48	13	42	8	38
59	25	55	20	49	14	43	9
29	60	26	56	21	50	15	44
63	**30**	61	27	57	22	51	16
32	64	**31**	62	28	58	23	52

CM-RCM(2)

61	29	62	30	63	31	64	32
25	57	26	58	27	59	28	60
53	21	54	22	55	23	56	24
17	49	18	50	19	51	20	52
45	13	46	14	47	15	48	16
9	41	10	42	11	43	12	44
37	**5**	38	**6**	39	**7**	40	8
1	33	2	34	3	35	4	36

MC(2)

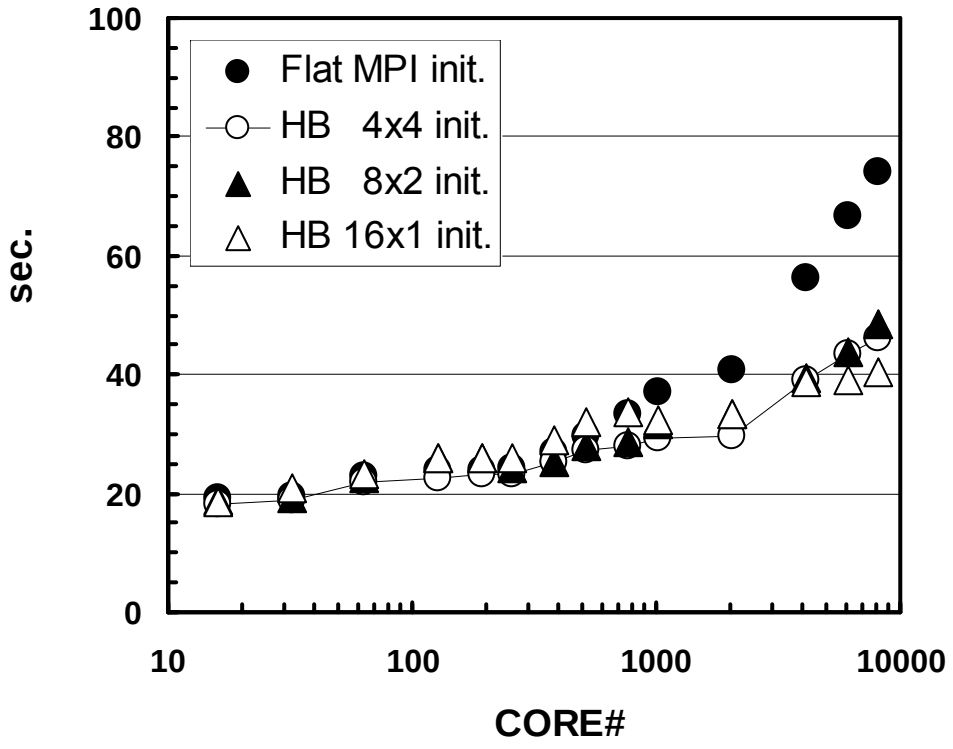
Weak Scaling

- Up to 8,192 cores (512 nodes)
  - 64^3 cells/core
  - 2,147,483,648 cells
- CM-RCM(2)

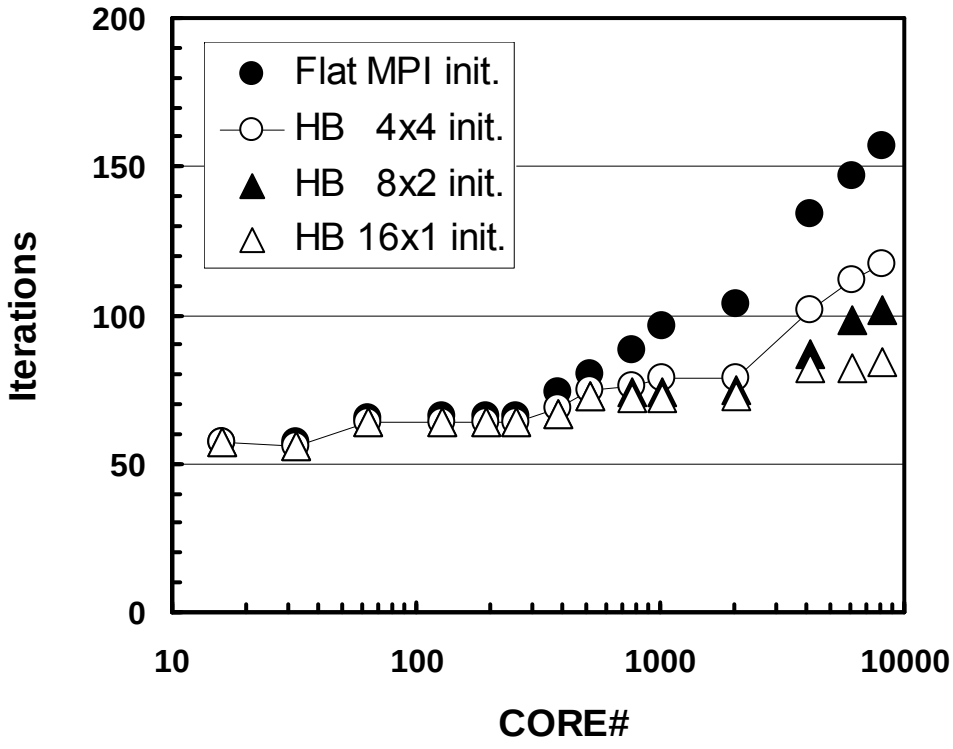
Weak Scaling

64^3cells/core, up to 8,192 cores (2.05×10^9 cells)

sec.

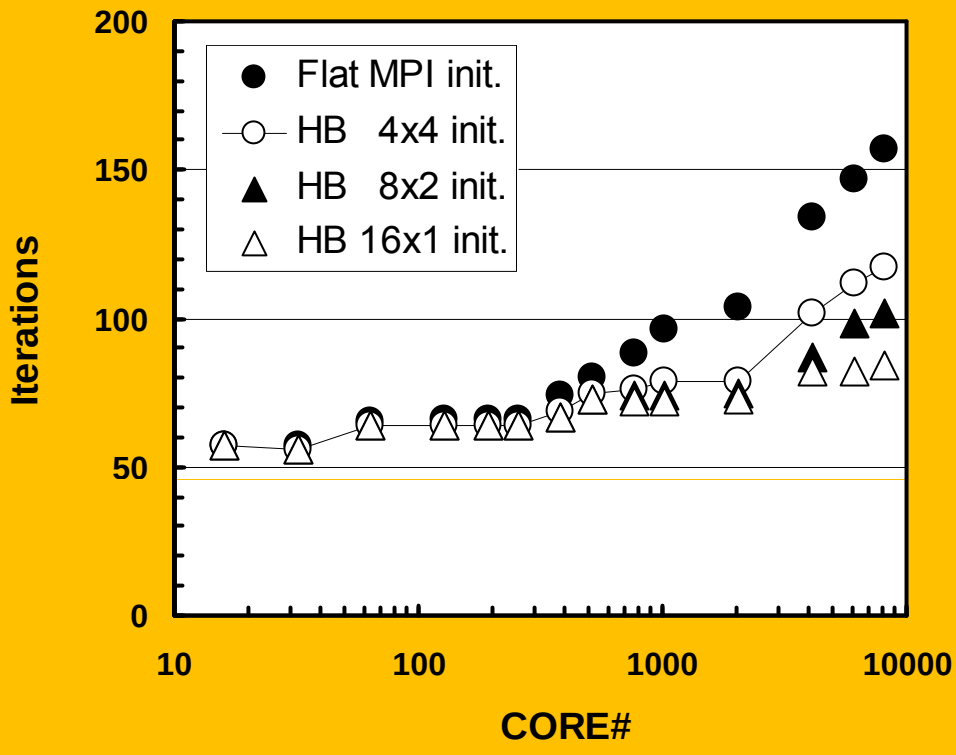


Iterations



Coarse Grid Solverの改良

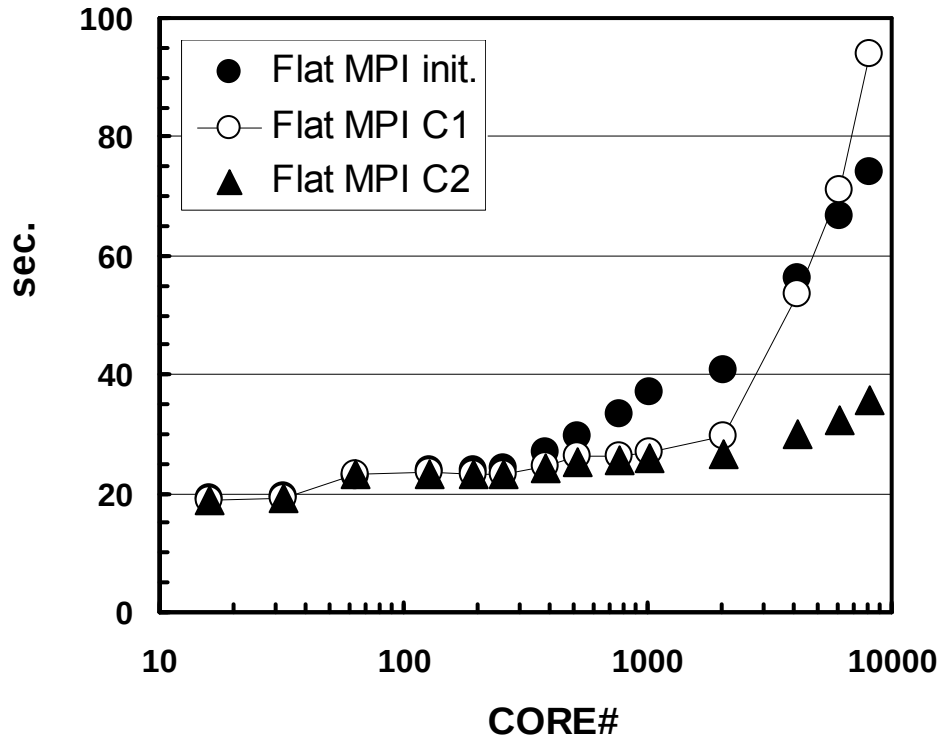
- 領域数が増えると反復回数が増加（特にFlat MPI）
- 最も粗い格子（Coarse Grid Solver）
  - 各領域1メッシュになった状態で1コアに集める
  - IC(0)スムージングを一回施す
- Coarse Grid Solver改良
  - IC(0)スムージングを収束（$\varepsilon=10^{-12}$）まで繰り返す：C1
  - マルチグリッド（V-cycle）を適用し，収束（$\varepsilon=10^{-12}$）まで繰り返す（8,192=32×16×16）：C2



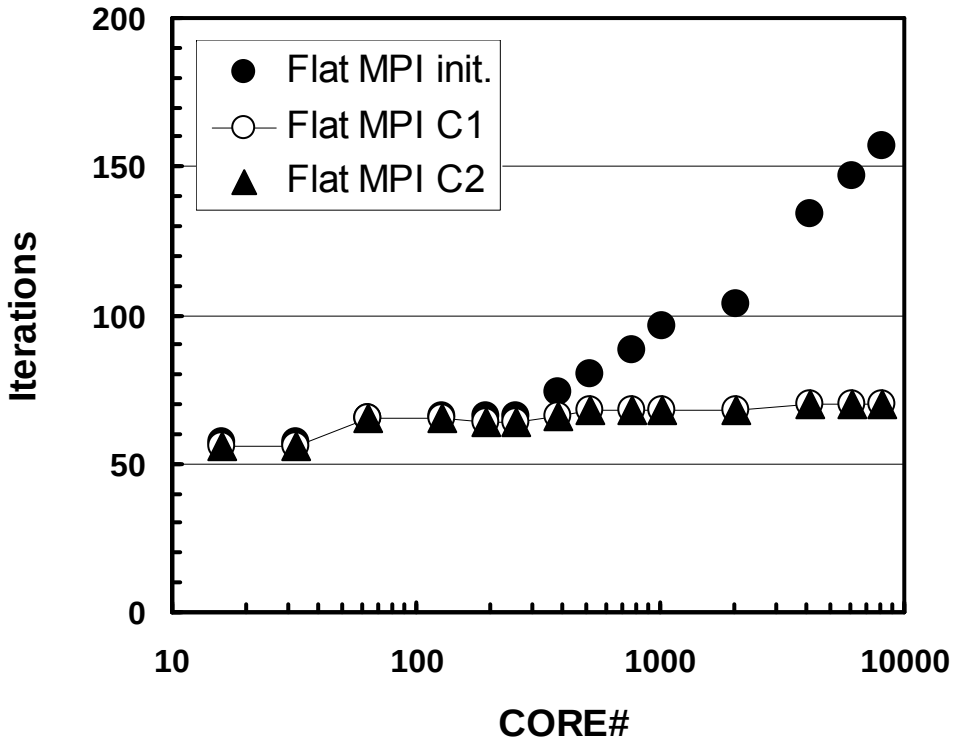
Weak Scaling: Flat MPI

64^{3}cells/core, up to 8,192 cores (2.05×10^9 cells)

sec.



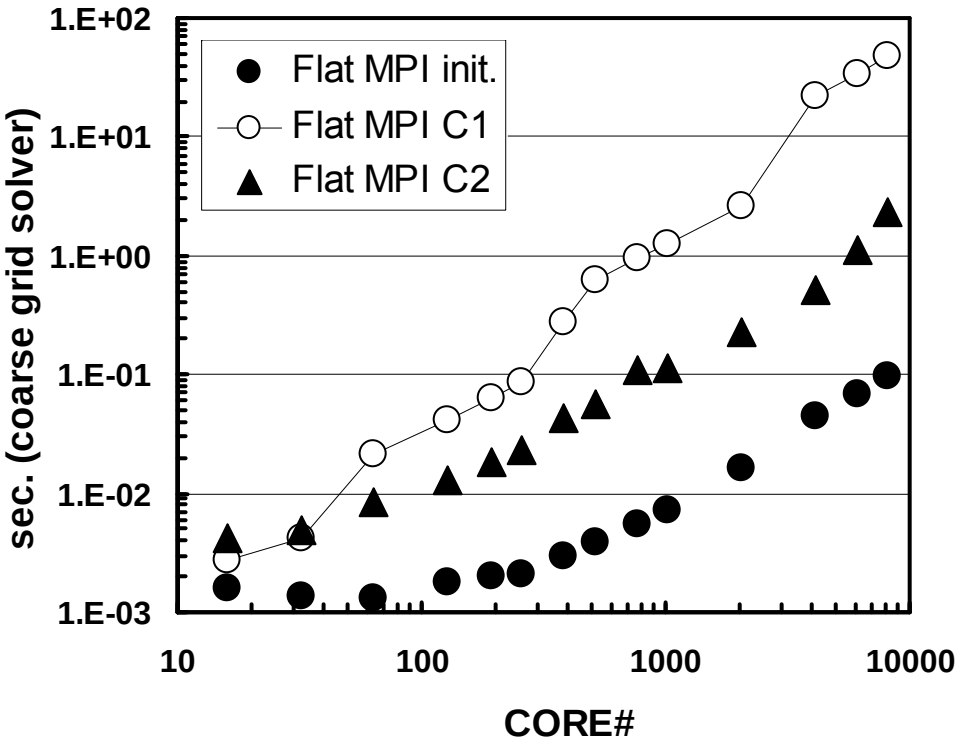
Iterations



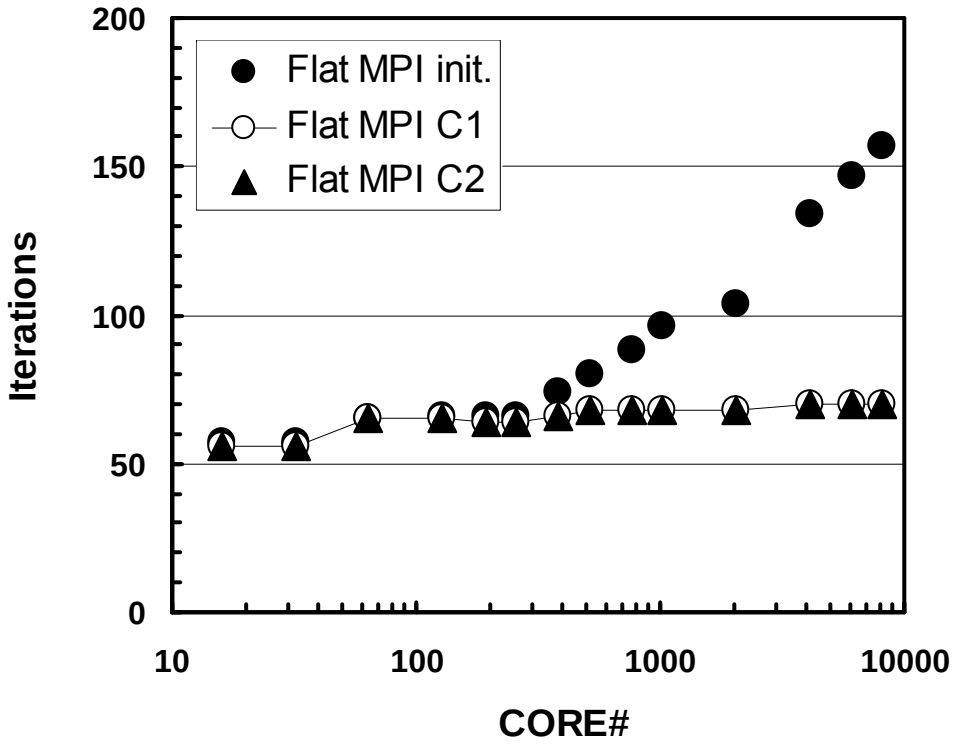
Weak Scaling: Flat MPI

64^{3}cells/core, up to 8,192 cores (2.05×10^9 cells)

Coarse Grid Solver



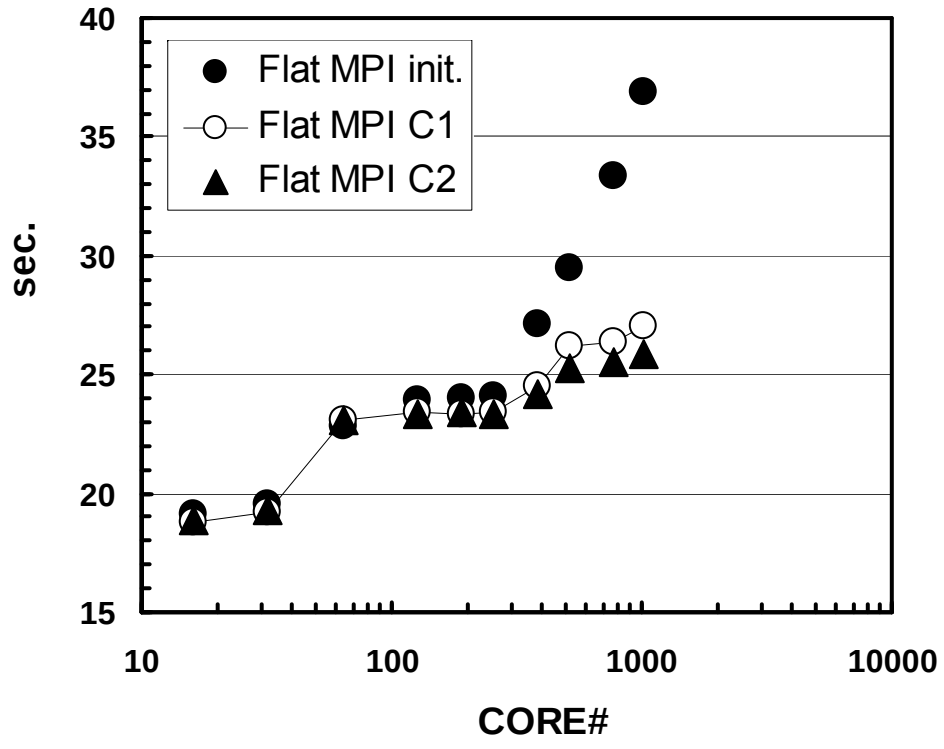
Iterations



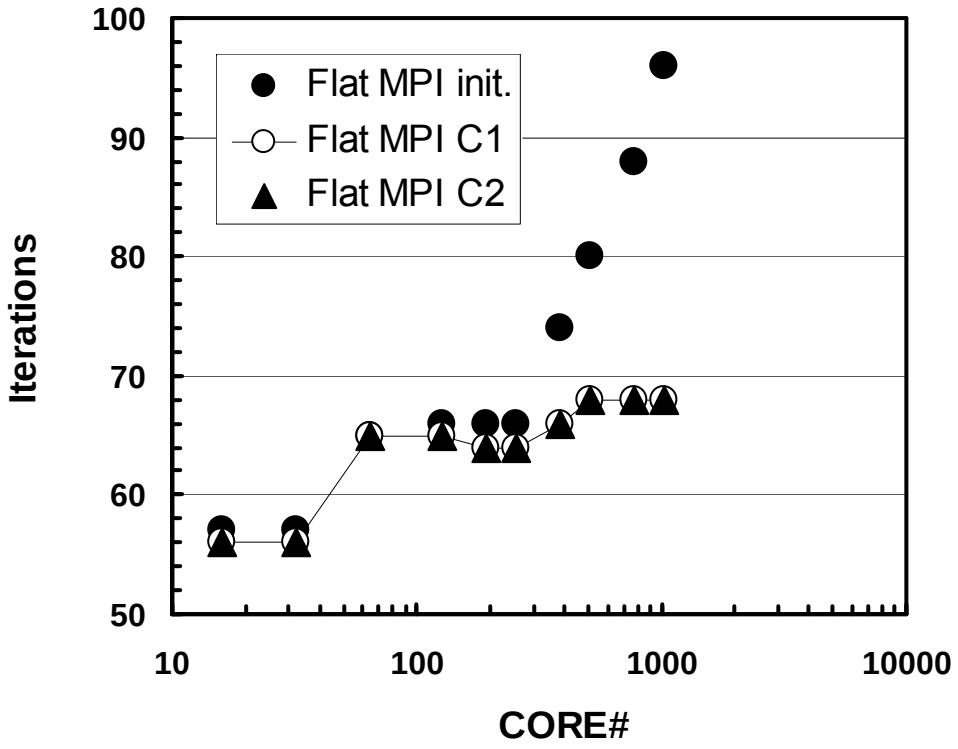
Weak Scaling: Flat MPI

64^{3}cells/core, up to 8,192 cores (2.05×10^9 cells)

sec.



Iterations

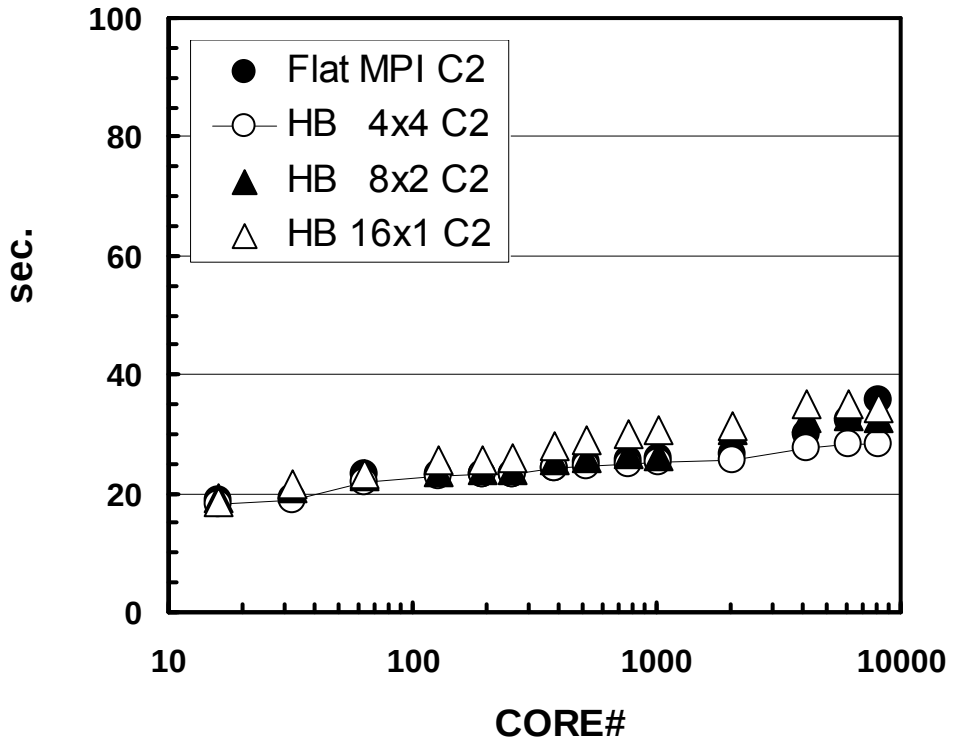


Weak Scaling

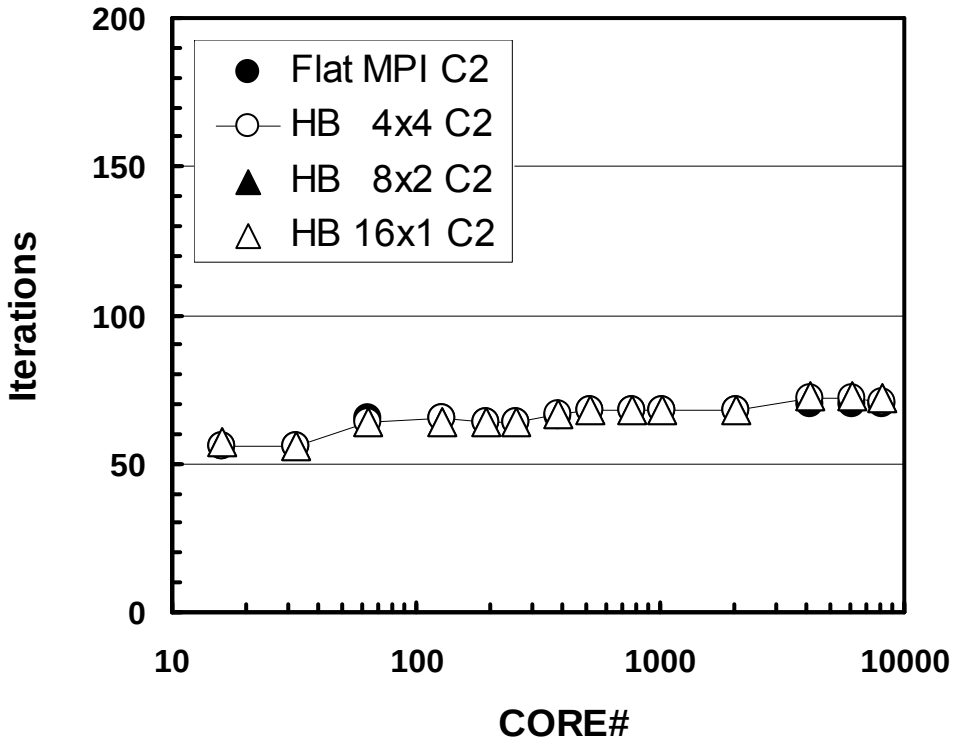
64^3cells/core, up to 8,192 cores (2.05×10^9 cells)

at 8,192 cores: Flat MPI(35.7sec), HB 4x4(28.4), 8x2(32.8), 16x1(34.4)

sec.



Iterations



Strong Scaling

- 512x256x256= 33,554,432 cells
- Up to 1,024 cores (64 nodes)

- CM-RCM(2)

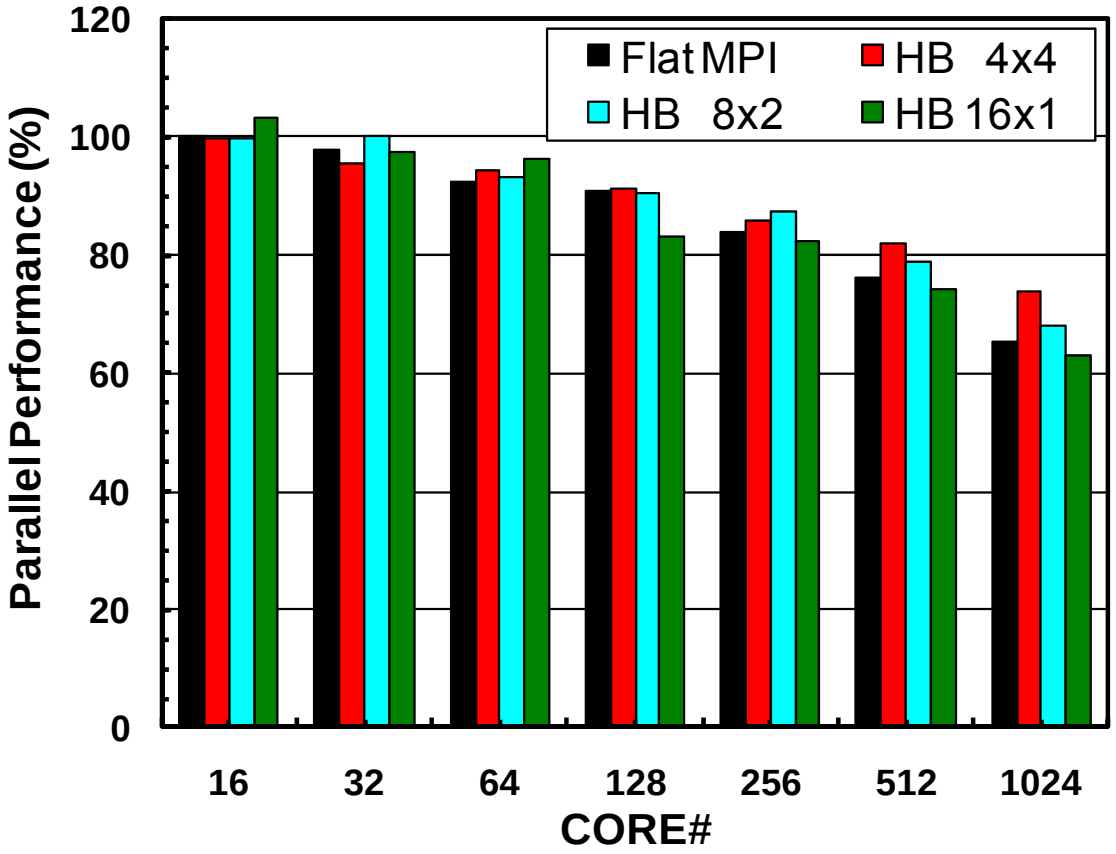
Strong Scale: Parallel Performance

512x256x256= 33,554,432 cells

based on performance of Flat MPI with 16 cores

HB 4x4 at 1,024 cores: 73.7%

Up is good



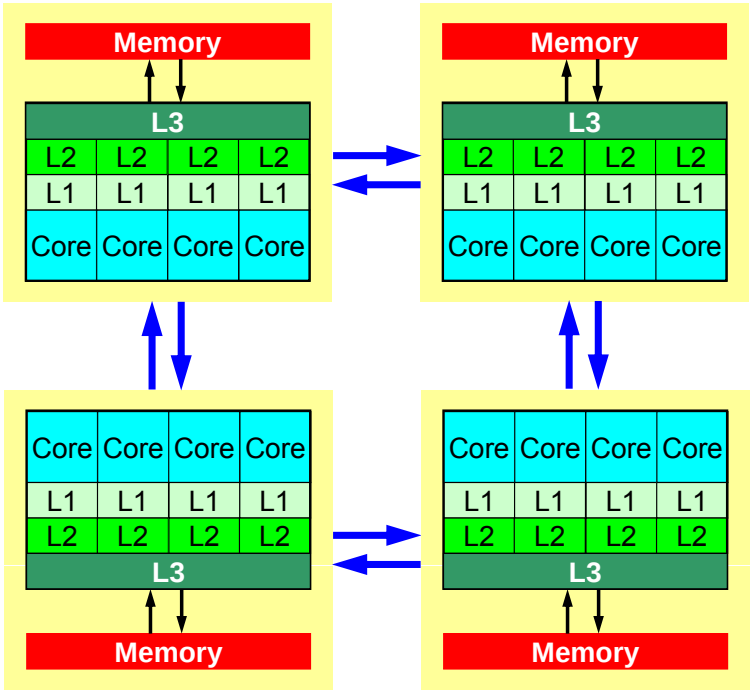
関連研究

- OpenMP/MPI Hybridを並列多重格子法に適用した例は近年特に増加している：Sandia, LLNL

- Alison Baker (LLNL) et al., “On the Performance of an Algebraic Multigrid Solver on Multicore Clusters”, (VECPAR 2010)
  - Hypre Library (BoomerAMG), weak scaling
  - Hera Cluster（T2K東大とほぼ同じアーキテクチャ）
    - ~216 nodes, 3,456コア（発表では>10,000コア）
  - MultiCore SUPport library (MCSup)
  - HB 4×4が最も性能が良い

まとめ

- （多重格子法（MG）前処理＋CG法）
  - 不均質多孔質媒体中の三次元地下水流れ，有限体積法
  - IC(0) smoother + ASDD，幾何学的MG
- OpenMP/MPI Hybrid 並列プログラミングモデル on T2K（東大）
  - NUMA Policy
  - First Touch Data Placement + Sequential Reordering
  - Coarse Grid Solver改良
    - HB 4x4（a single MPI process per socket）が最も効率が良い：メモリを最も効率よく使っている，通信オーバーヘッドも少ない
    - 反復回数は並列プログラミングモデルによってほとんど変化しない



今後の課題

- 粗い格子レベルにおけるコア数の漸減
  - 全体のコア数，領域数が増えると通信オーバーヘッドが増加
- Hybridにおける領域内並べ替え
  - CM-RCM
  - HID
  - 並列化：結構時間がかかる

- Communication Reducing Algorithms
  - 並列MG：とにかく通信多い

Further Re-Ordering for Continuous Memory Access: Sequential 5 colors, 8 threads