Parallel Multigrid Solvers using OpenMP/MPI Hybrid Programming Models on Multi-Core/Multi-Socket Clusters

Kengo Nakajima
Information Technology Center, The University of Tokyo
Japan Science & Technology Agency (JST)

9th International Meeting High Performance Computing for Computational Science
VECPAR 2010
June 22-25, 2010, Berkeley, California, USA

Outline

- Background
- Application
- Results
  - Effect of Optimization
  - Weak Scaling
  - Strong Scaling
    - Further Optimization
- Summary

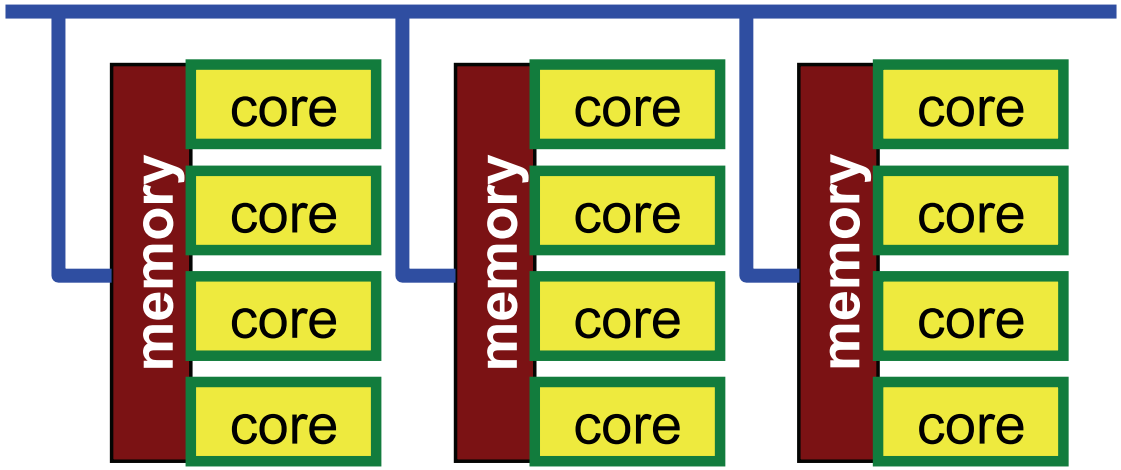
Motivations

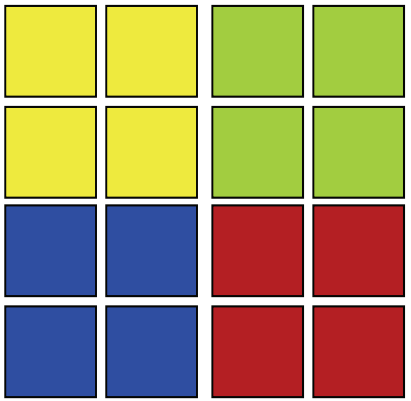
- Parallel Multigrid Solvers for FVM-type appl. on T2K Open Supercomputer
  - Todai Combined Cluster (T2K/Tokyo)

- Flat MPI vs. Hybrid (OpenMP+MPI)
- Expectations for Hybrid
  - Number of MPI processes (and sub-domains) to be reduced
  - O(10^8-10^9)-way MPI might not scale
  - Convergence of parallel iterative solvers with block-Jacobi-type localized IC/ILU preconditioners

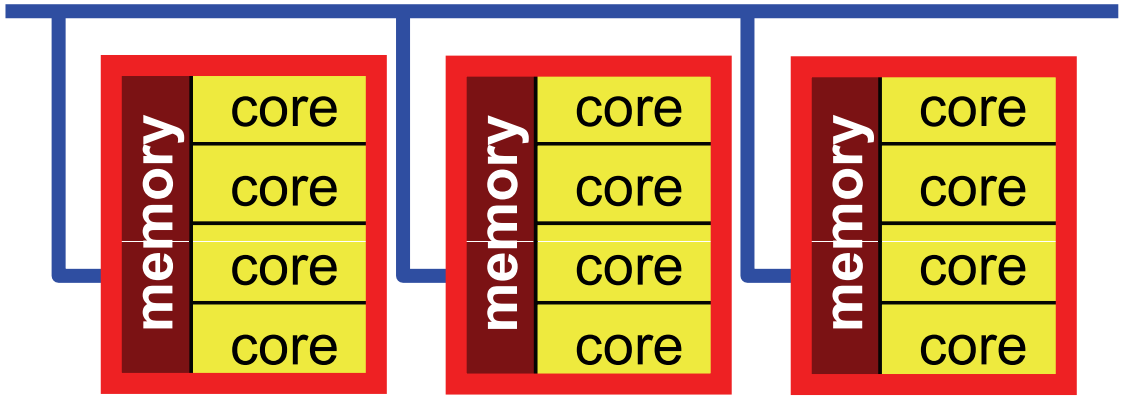
Flat MPI vs. Hybrid

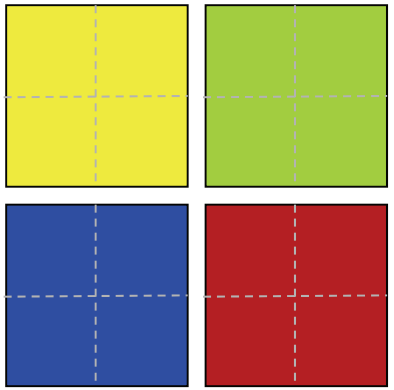
Flat-MPI：Each PE -> Independent





Hybrid：Hierarchal Structure





Memory performance is important

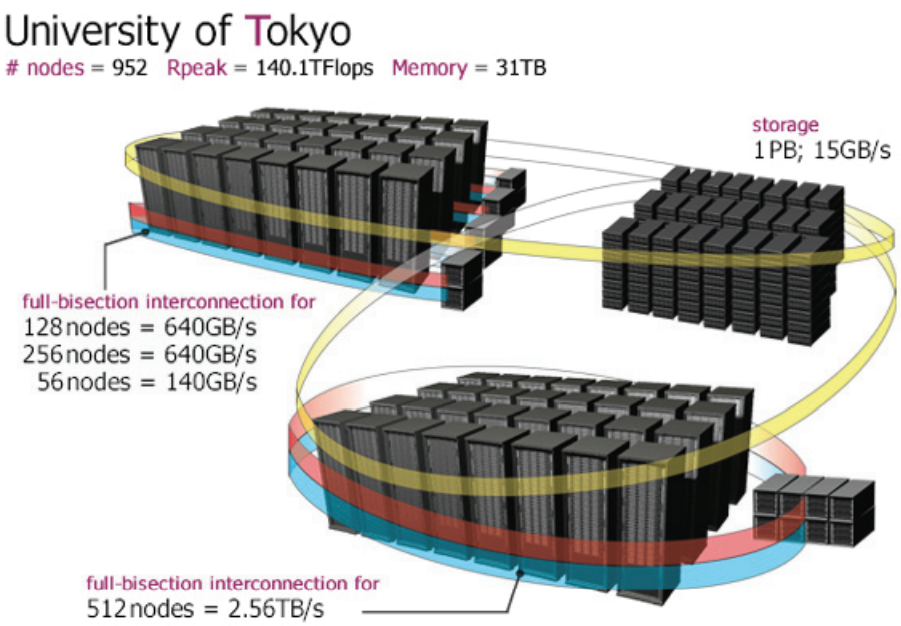
- Sparse Linear Solvers
  - Memory-bound

- First-Touch Data Placement
- Memory Access Pattern
  - Continuous “Sequential” Memory Access

T2K/Tokyo (1/2)

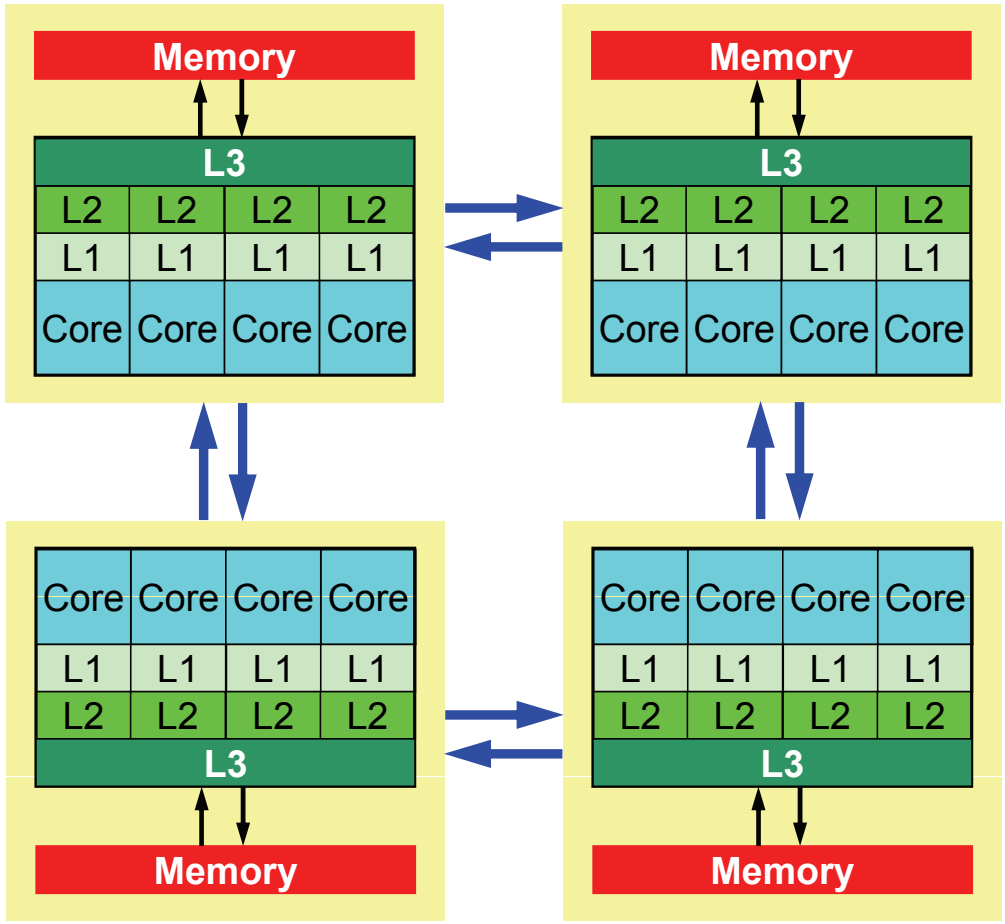
- "T2K Open Supercomputer Alliance"
  - http://www.open-supercomputer.org/
  - Tsukuba, Tokyo, Kyoto

- "T2K Open Supercomputer (Todai Combined Cluster)"
  - by Hitachi
  - op. started June 2008
  - Total 952 nodes (15,232 cores), 141 TFLOPS peak
    - Quad-core Opteron (Barcelona)
  - 53rd in TOP500 (JUN 2010)
  - Fat-Tree with Myrinet-10G



T2K/Tokyo (2/2)

- AMD Quad-core Opteron (Barcelona) 2.3GHz
- 4 “sockets” per node
  - 16 cores/node
- Multi-core, multi-socket system
- cc-NUMA architecture
  - careful configuration needed
    - local data ~ local memory



Flat MPI vs. Hybrid

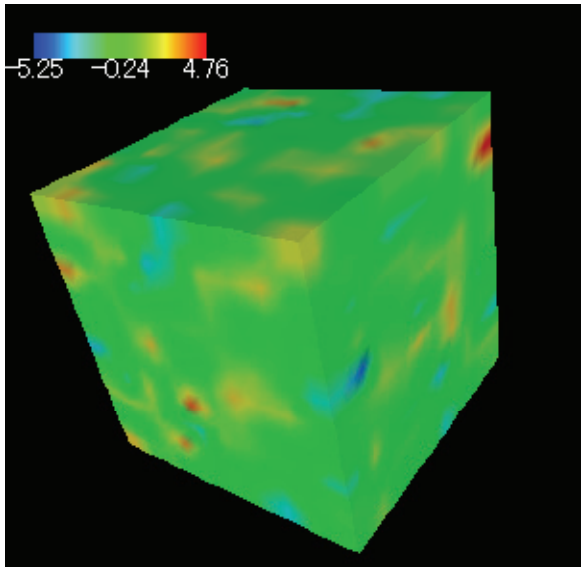
- Performance is determined by various parameters

- Hardware
  - core architecture, performance
  - memory bandwidth, latency
  - network bandwidth, latency
  - their balance
- Applications
  - types: memory or network/communication bound
  - problem size

- Background
- **Application**
- Results
  - Effect of Optimization
  - Weak Scaling
  - Strong Scaling
    - Further Optimization
- Summary

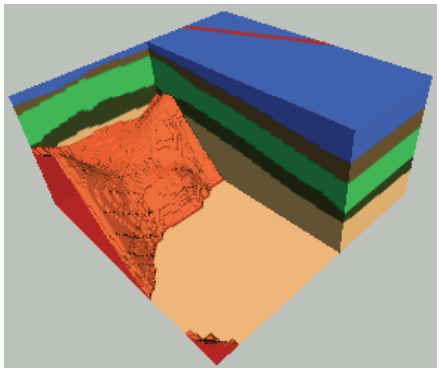
Target Application

- 3D Groundwater Flow Heterogeneous Porous Media
  - Poisson's equation
  - Randomly distributed water conductivity
  - Distribution of water conductivity is defined through methods in geostatics〔Deutsch & Journel, 1998〕
- Finite-Volume Method on Cubic Voxel Mesh
- Cyclic Structure of Heterogeneity for every 128^3 grids in each direction

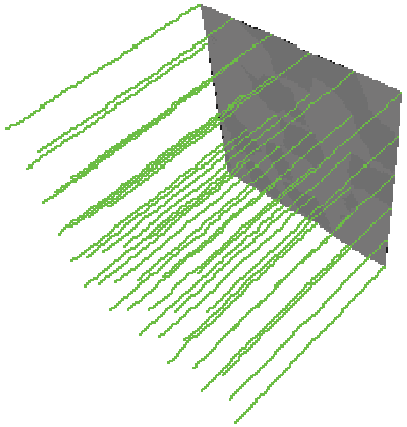


Groundwater Flow through Heterogeneous Porous Media

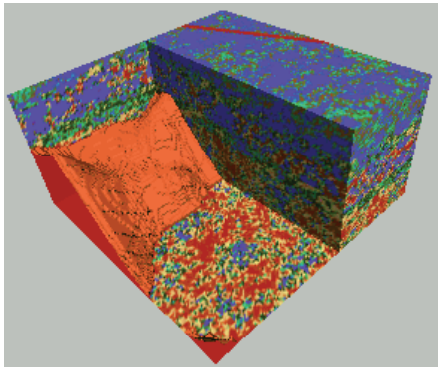
Homogeneous



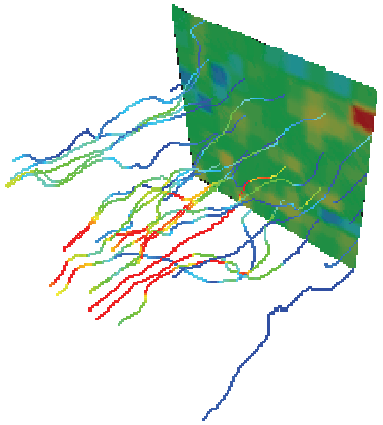
Uniform Flow Field



Heterogeneous

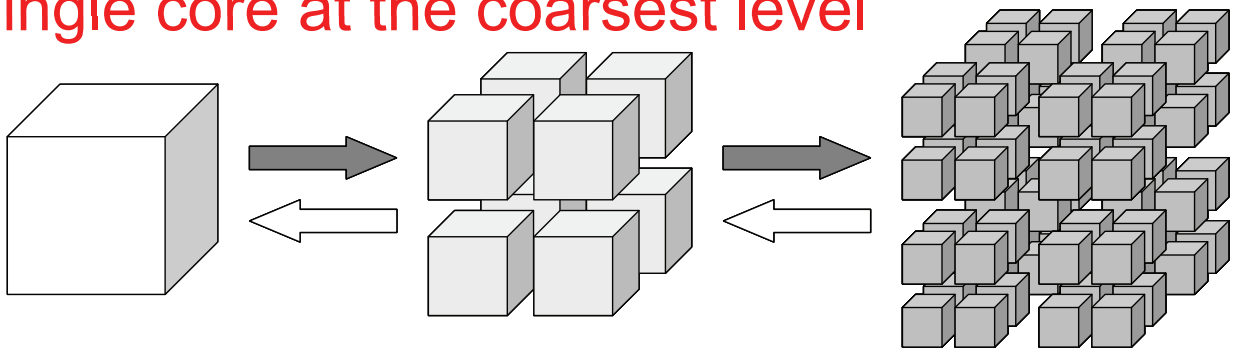


Random Flow Field



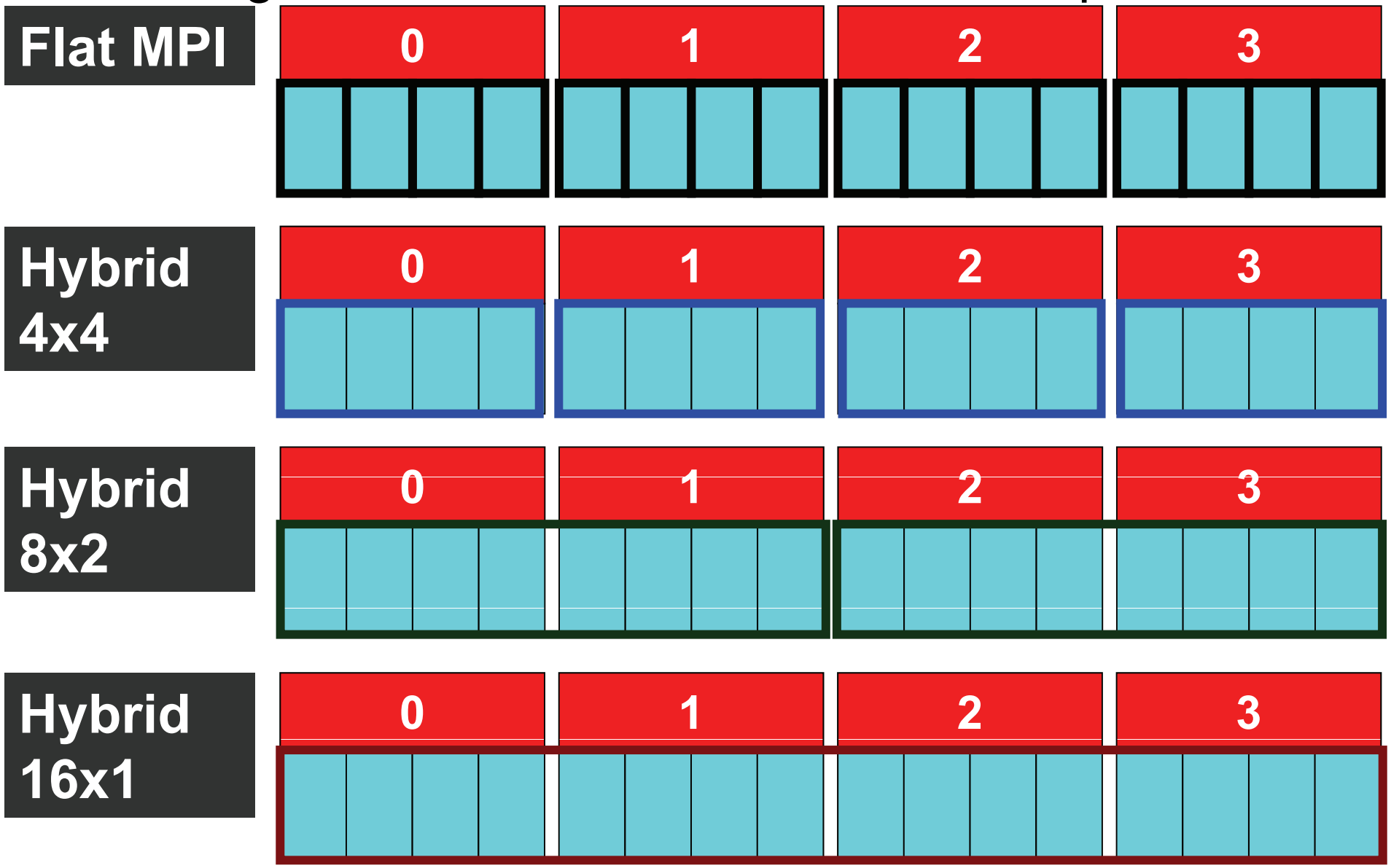
Linear Solvers

- Preconditioned CG Method
  - Multigrid Preconditioning
  - IC(0) for Smoothing Operator (Smoother)
- Parallel Geometric Multigrid Method
  - Scalable method
  - 8 fine meshes (children) form 1 coarse mesh (parent) in isotropic manner (octree)
  - V-cycle
  - Domain-Decomposition-based: Block-Jacobi
  - Operations using a single core at the coarsest level



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

Higher Performance of HB16x1 is important

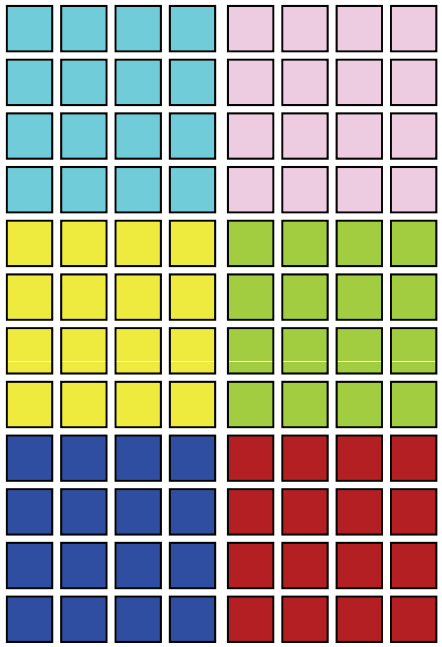


Domain Decomposition

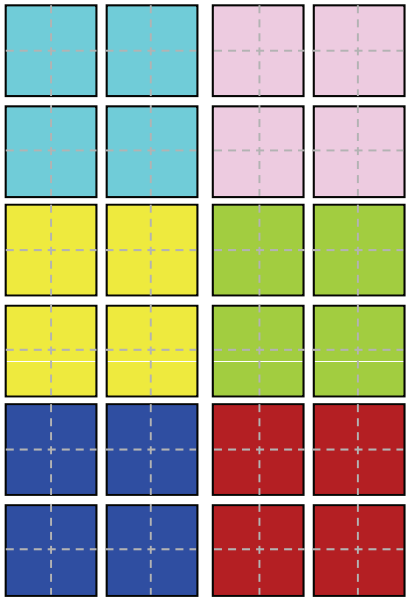
Inter Domain: MPI-Block Jacobi

Intra Domain: OpenMP-Threads (re-ordering)

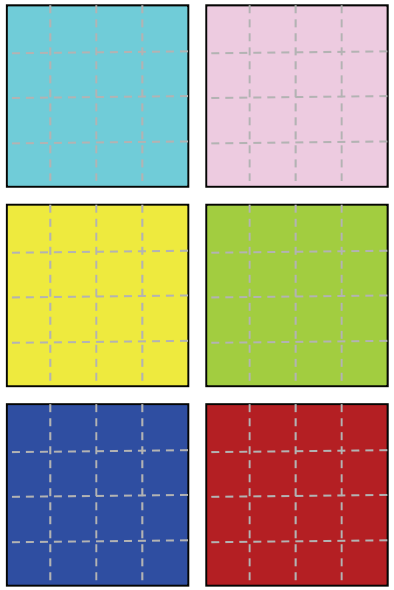
example: 6 nodes, 24 sockets, 96 cores



Flat MPI



HB 4x4



HB 16x1

IC(0) as smoother of Multigrid

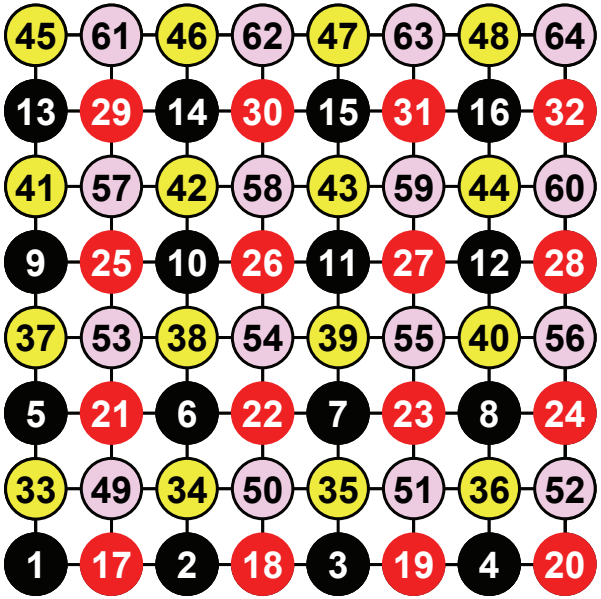
- IC(0) smoother with Additive Schwartz Domain Decomposition (ASDD) provides robust convergence and scalable performance of parallel computation, even for ill-conditioned problems〔KN 2002〕.

Reordering for extracting parallelism in each domain

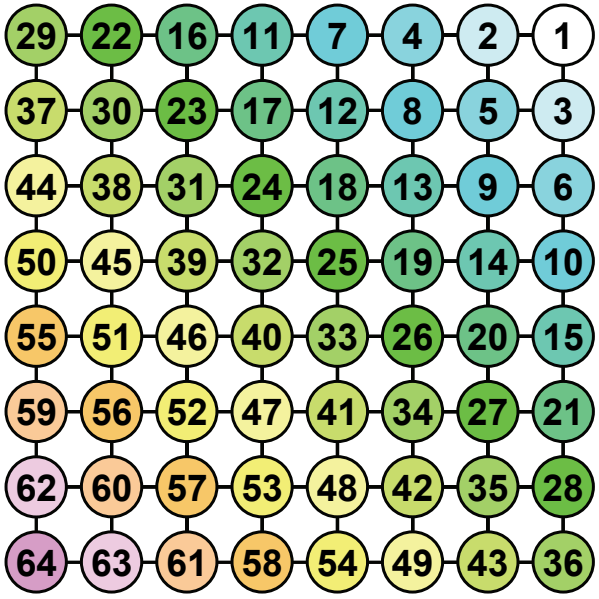
- Krylov Iterative Solvers
  - Dot Products
  - SMVP
  - DAXPY
  - Preconditioning
- IC/ILU Factorization, Forward/Backward Substitution
  - Global Dependency
  - Reordering needed for parallelism ([KN 2003] on the Earth Simulator, KN@CMCIM-2002)
  - Multicoloring, RCM, CM-RCM

Ordering Methods

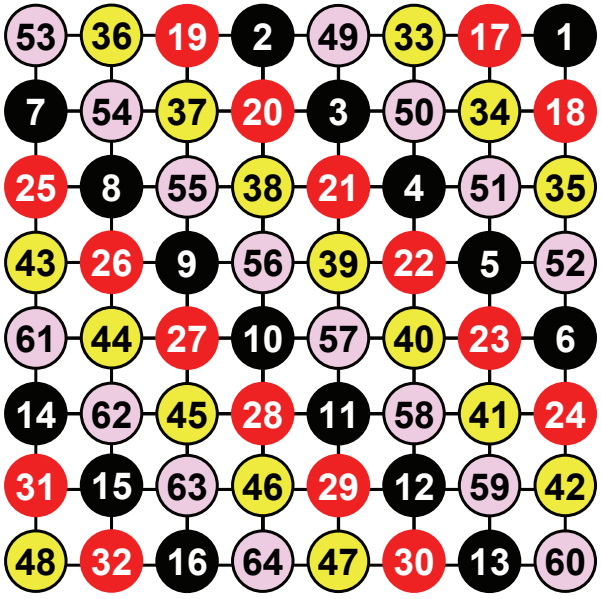
Elements in “same color” are independent: to be parallelized



MC (Color#=4)
Multicoloring



RCM
Reverse Cuthill-Mckee



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

- Background
- Application
- **Results**
  - **Effect of Optimization**
  - Weak Scaling
  - Strong Scaling
    - Further Optimization
- Summary

Hardware/Software

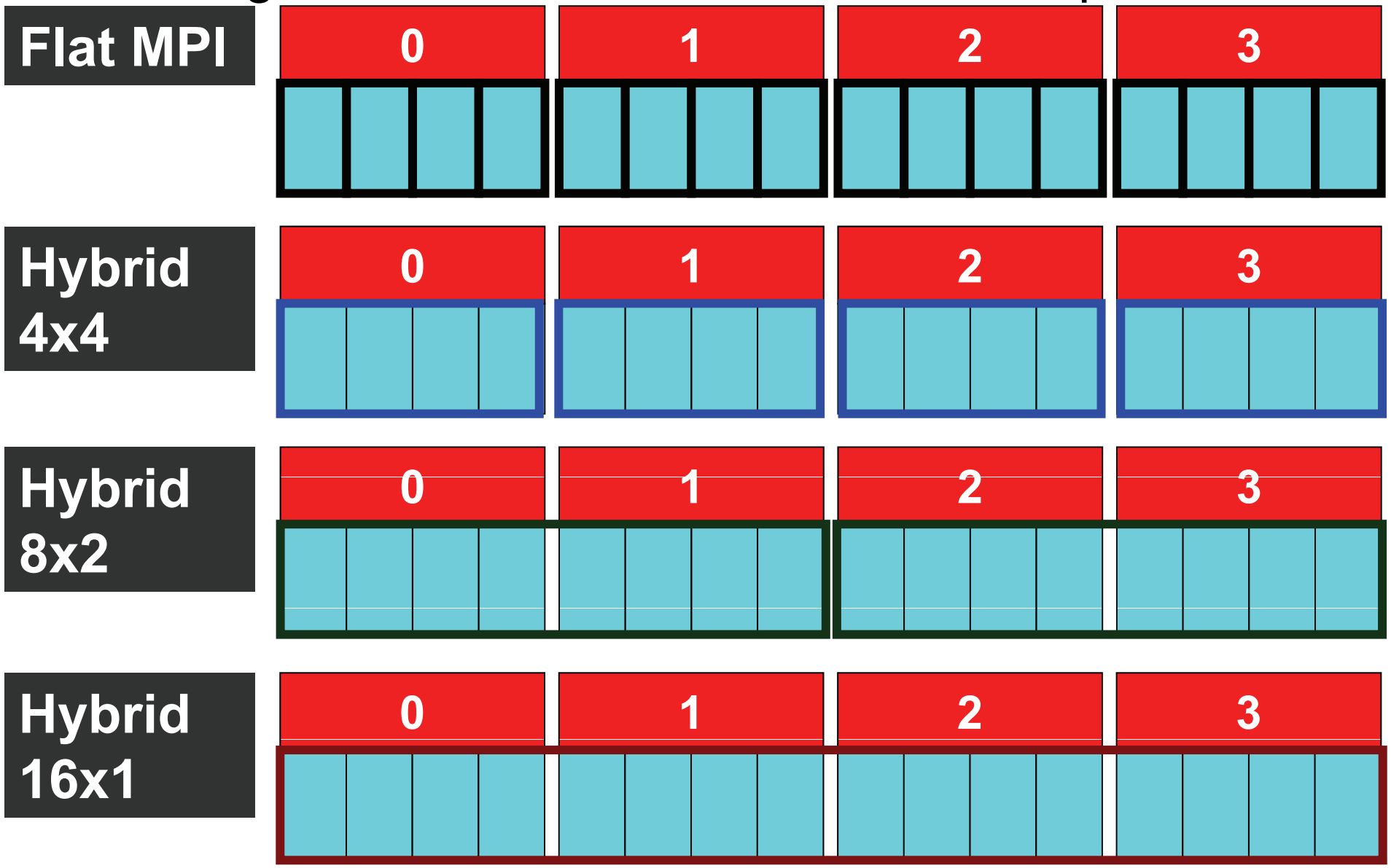
- T2K/Tokyo
  - up to 64 nodes (1,024 cores)
- Program
  - Hitachi FORTRAN90 + MPI
  - CRS matrix storage
  - CM-RCM Reordering for OpenMP

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

- $|Ax-b|/|b|=10^{-12}$ for Convergence
- Heterogeneity
  - Ratio of MAX/MIN water conductivity= 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 restriction/prolongation

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

Higher Performance of HB16x1 is important



Effect of Optimization

- 64 cores (4 nodes) of T2K/Tokyo
  - 64^3 cells/core
  - 16,777,216 cells

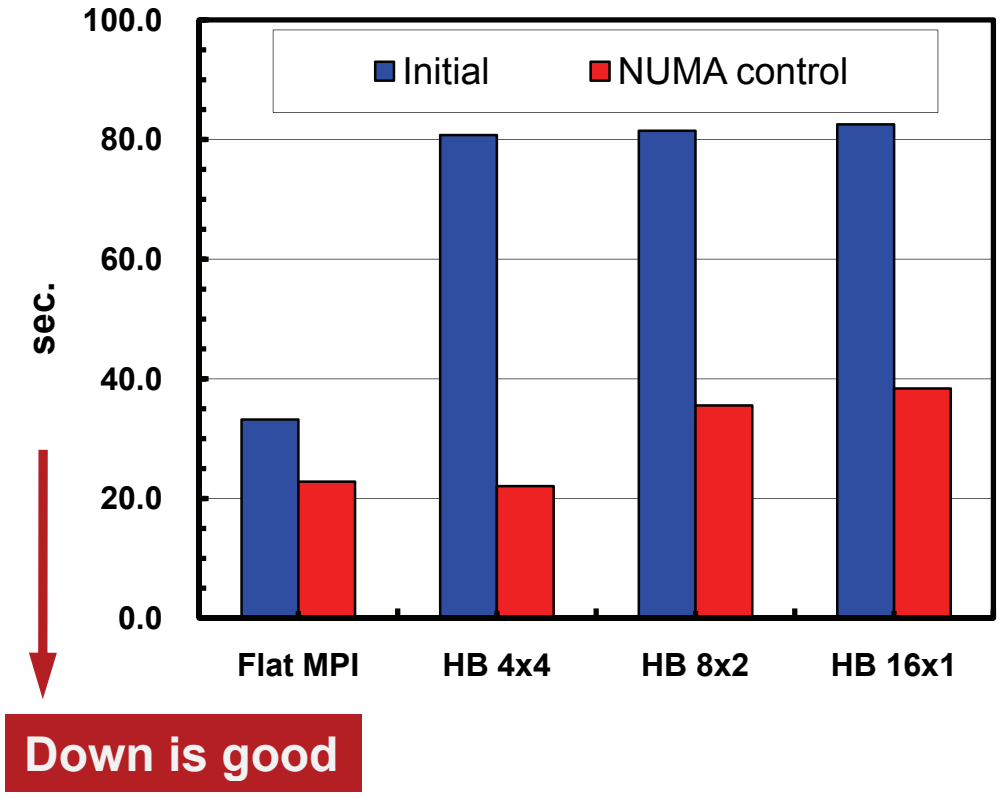
- Full Optimization
  - NUMA Control
  - First Touch Data Placement
  - Further Reordering (with Contiguous Memory Access)

Effect of NUMA Control

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

Time for Linear Solvers

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`

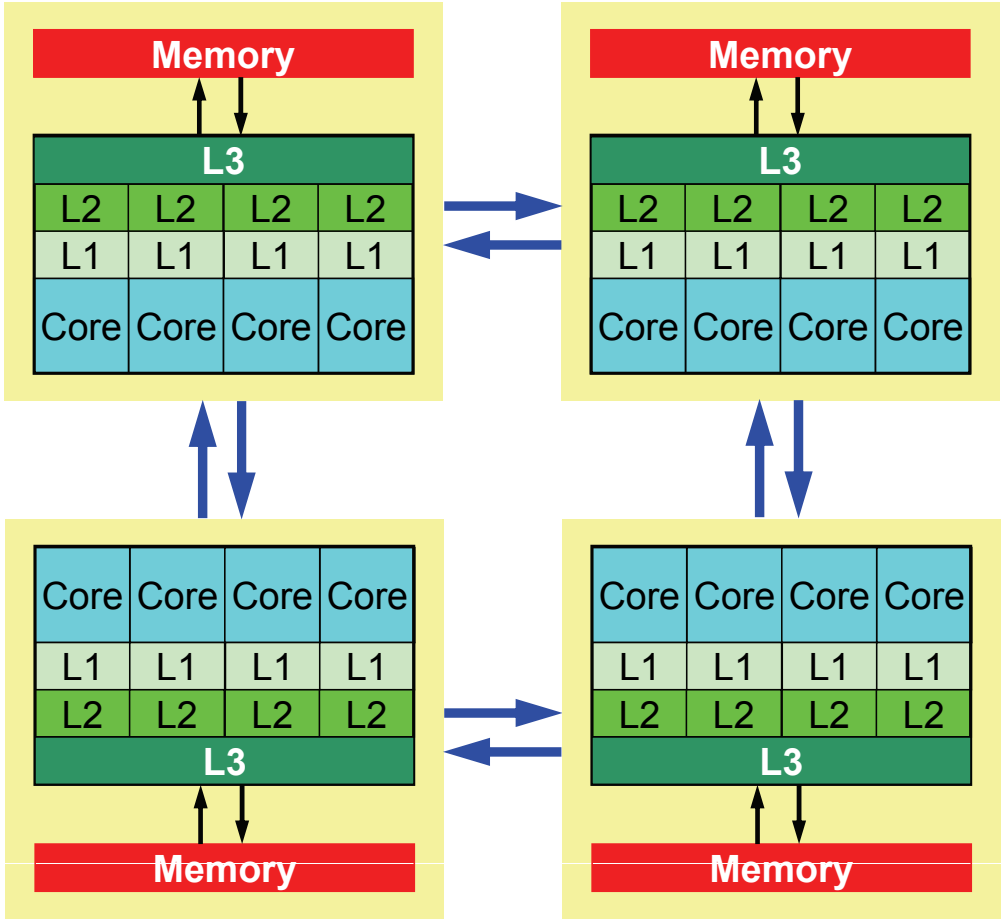


First Touch Data Placement

Local Data – Local Memory

The most common NUMA page-placement algorithm is the "first touch" algorithm, in which the PE first referencing a region of memory will have the page holding that memory assigned to it.

A very common technique in OpenMP program is to initialize data in parallel using the same loop schedule as will be used later in the computations.



First Touch Data Placement

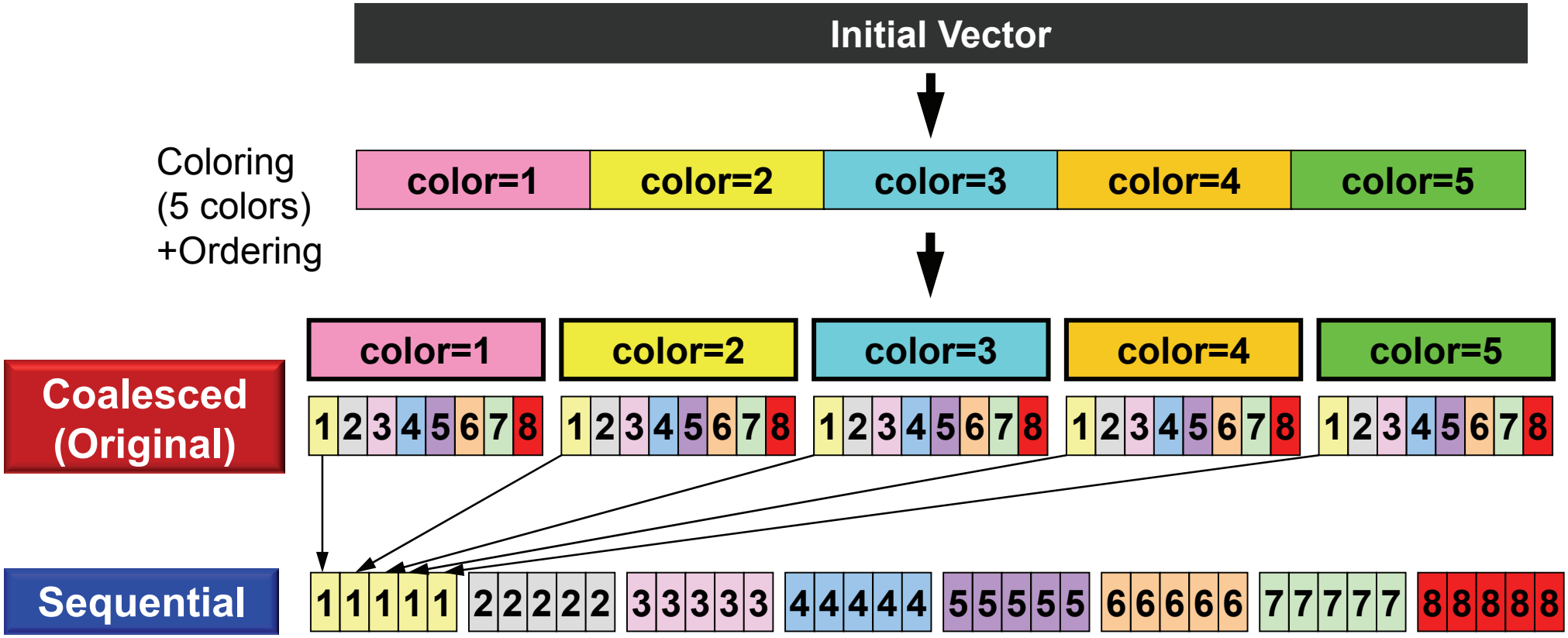
Method of Initialization
to be initialized as computation

```
  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
```

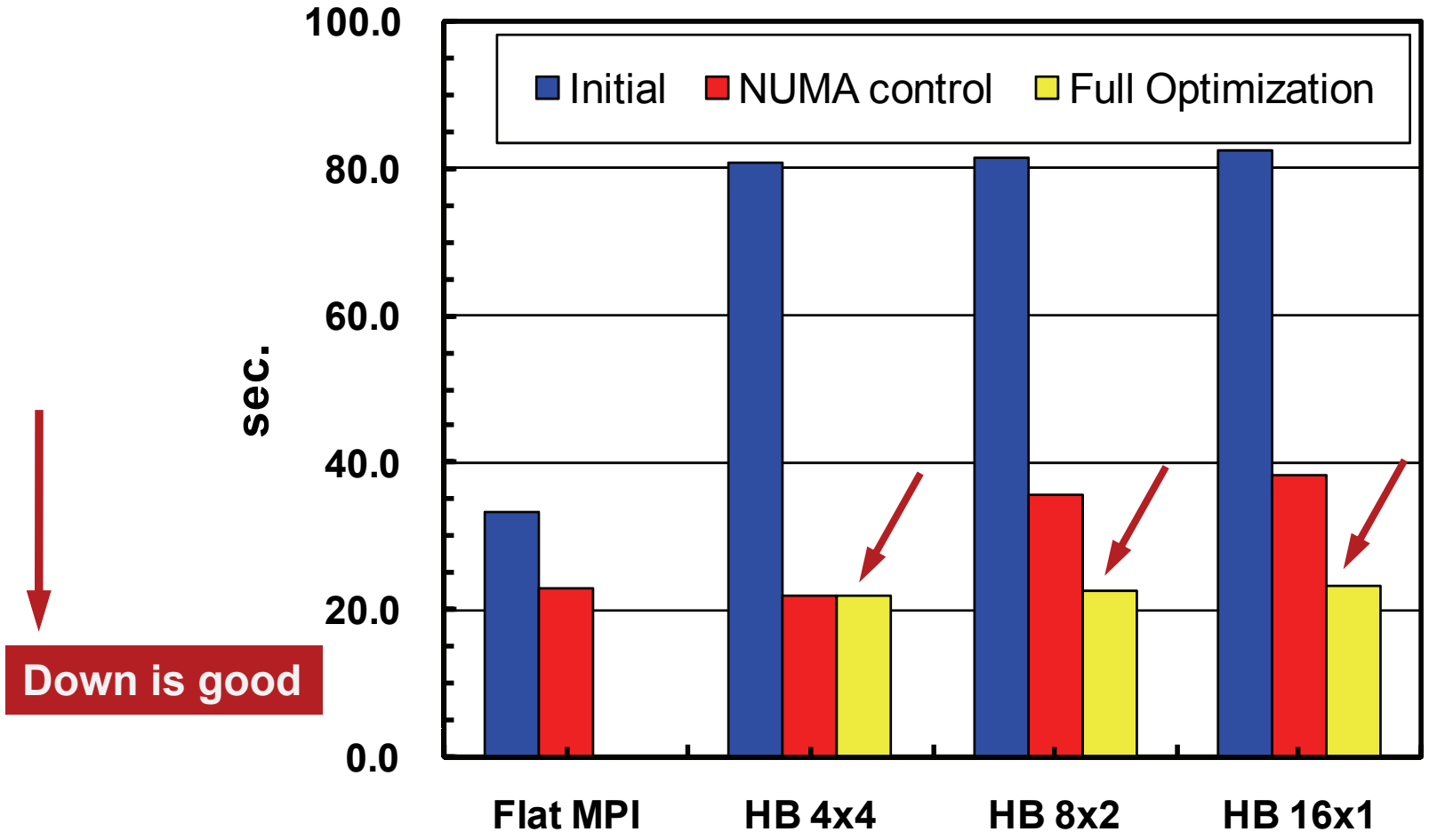
Further Re-Ordering for Continuous Memory Access: Sequential 5 colors, 8 threads



Effect of F.T. + Sequential Data Access

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

Time for Linear Solvers, HB 4x4 is the fastest



- Background
- Application
- **Results**
  - Effect of Optimization
  - **Weak Scaling**
  - **Strong Scaling**
    - **Further Optimization**
- Summary

Weak Scaling

- Up to 1,024 cores (64 nodes)
  - 64^3 cells/core
  - 268,435,456 cells

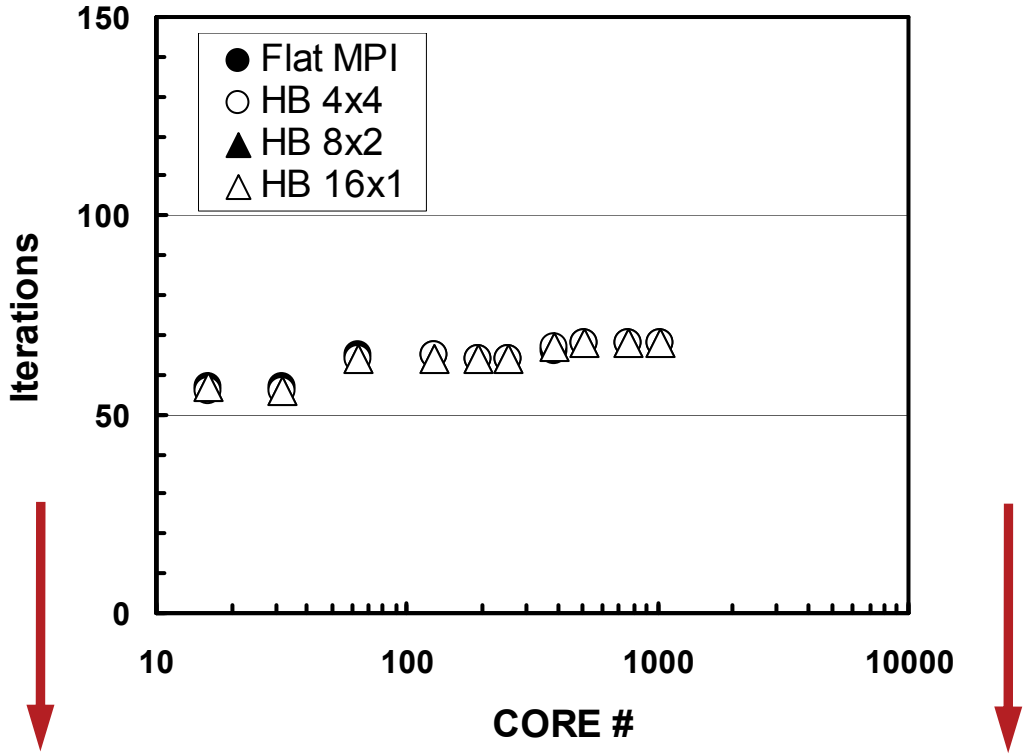
- CM-RCM(2)

Improved Coarse-Grid Smoother

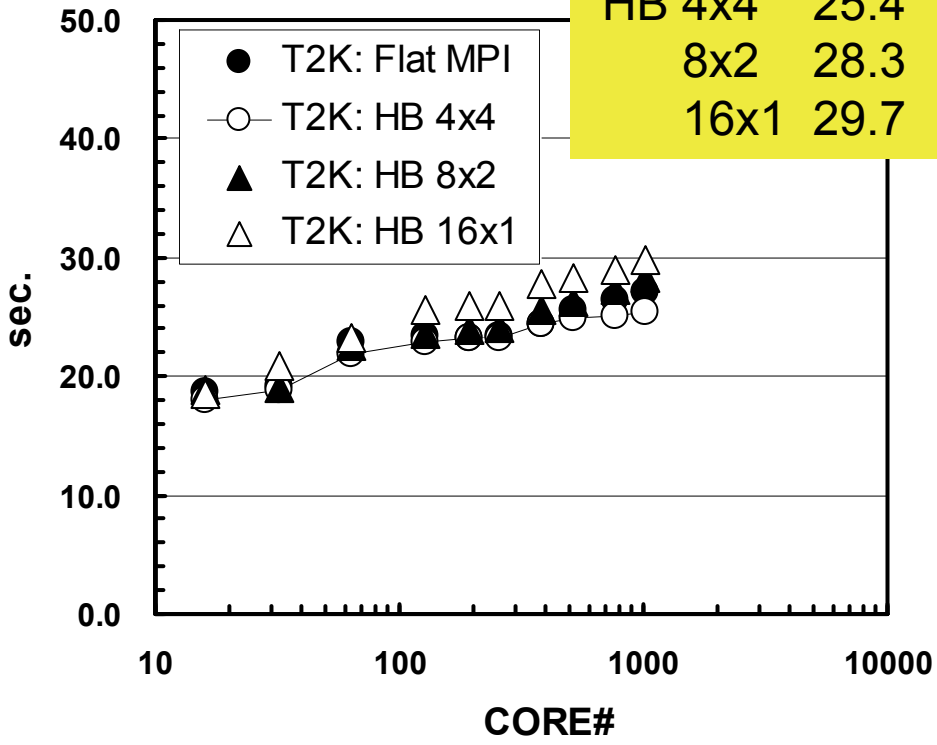
64^3cells/core, up to 1,024 cores/268,435,456 cells

HB 4x4 provides the best performance

Iterations



sec.



At 1,024 cores

Flat MPI: 27.1 sec

HB 4x4 25.4

8x2 28.3

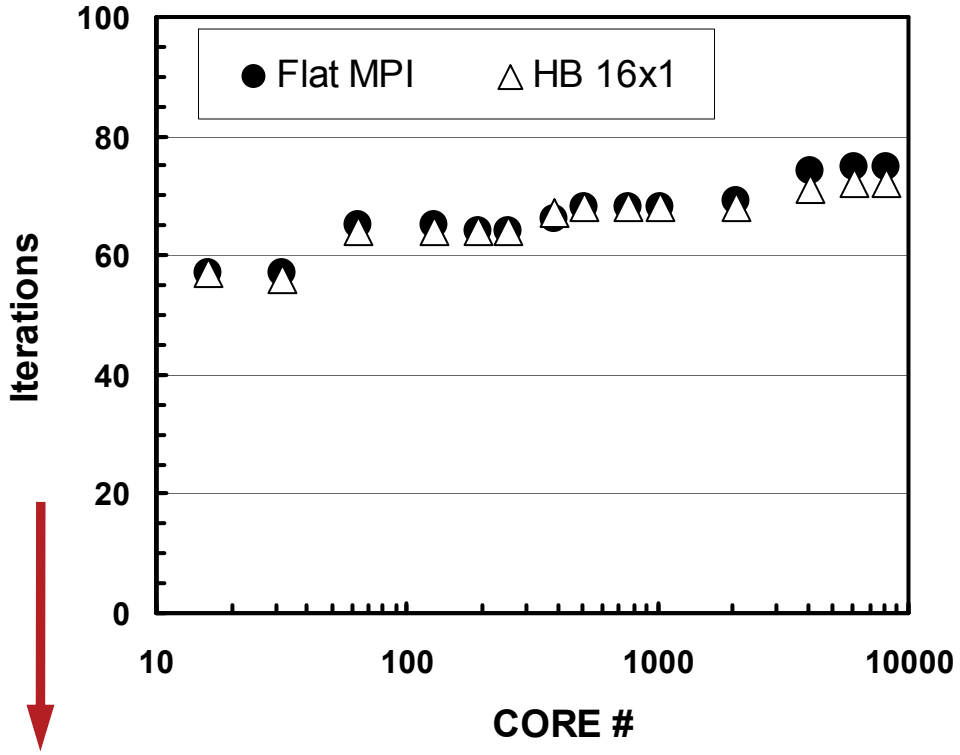
16x1 29.7

Down is good

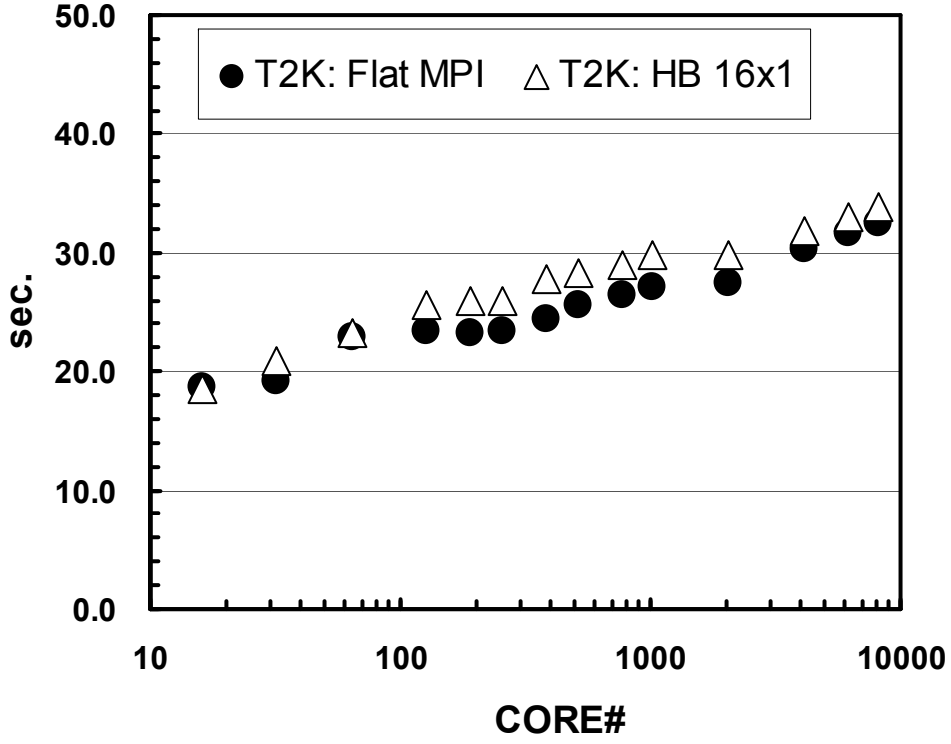
More Cores …

64^3cells/core, up to 8,192 cores/2,147,483,648 cells

Iterations



sec.



Down is good

Strong Scaling

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

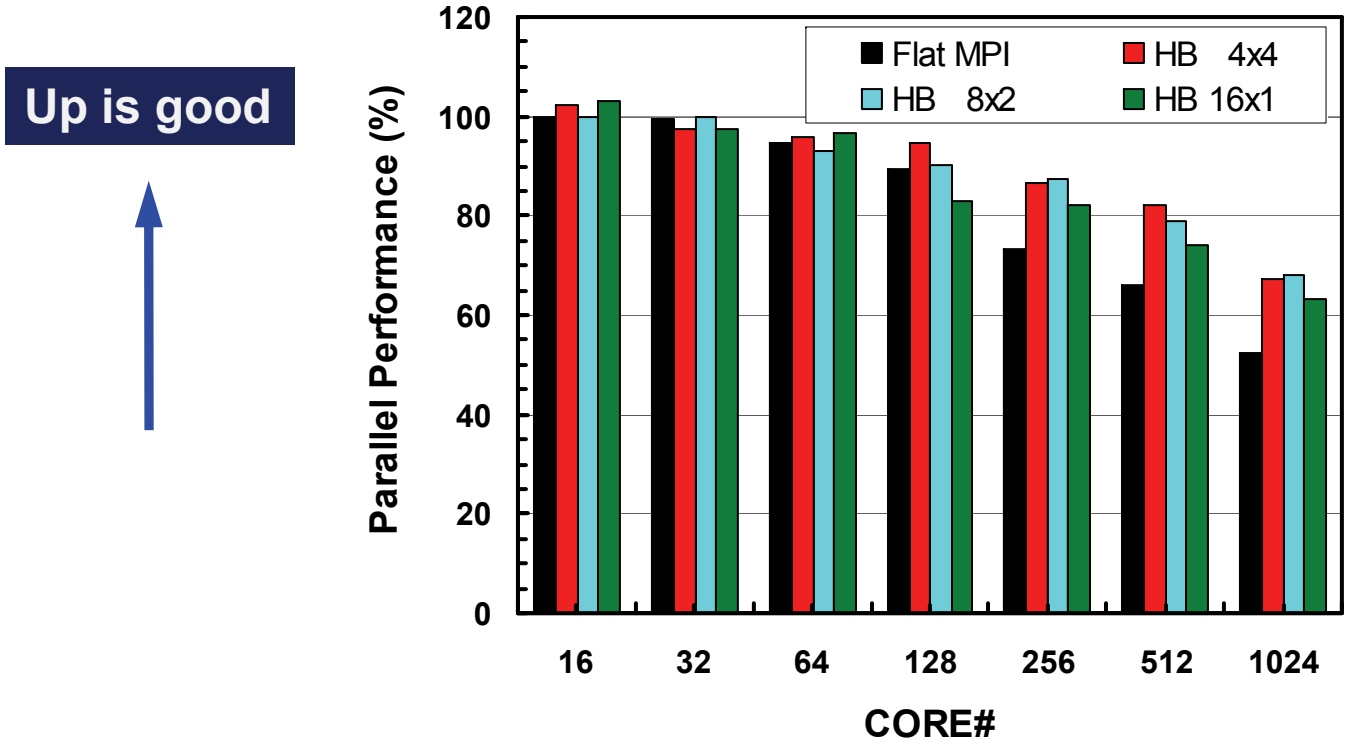
- CM-RCM(2)

Strong Scale: Parallel Performance

512x256x256= 33,554,432 cells

based on performance of Flat MPI with 16 cores

(Improved coarse-grid smoother does not work well because cost per iteration is larger for many-core cases)

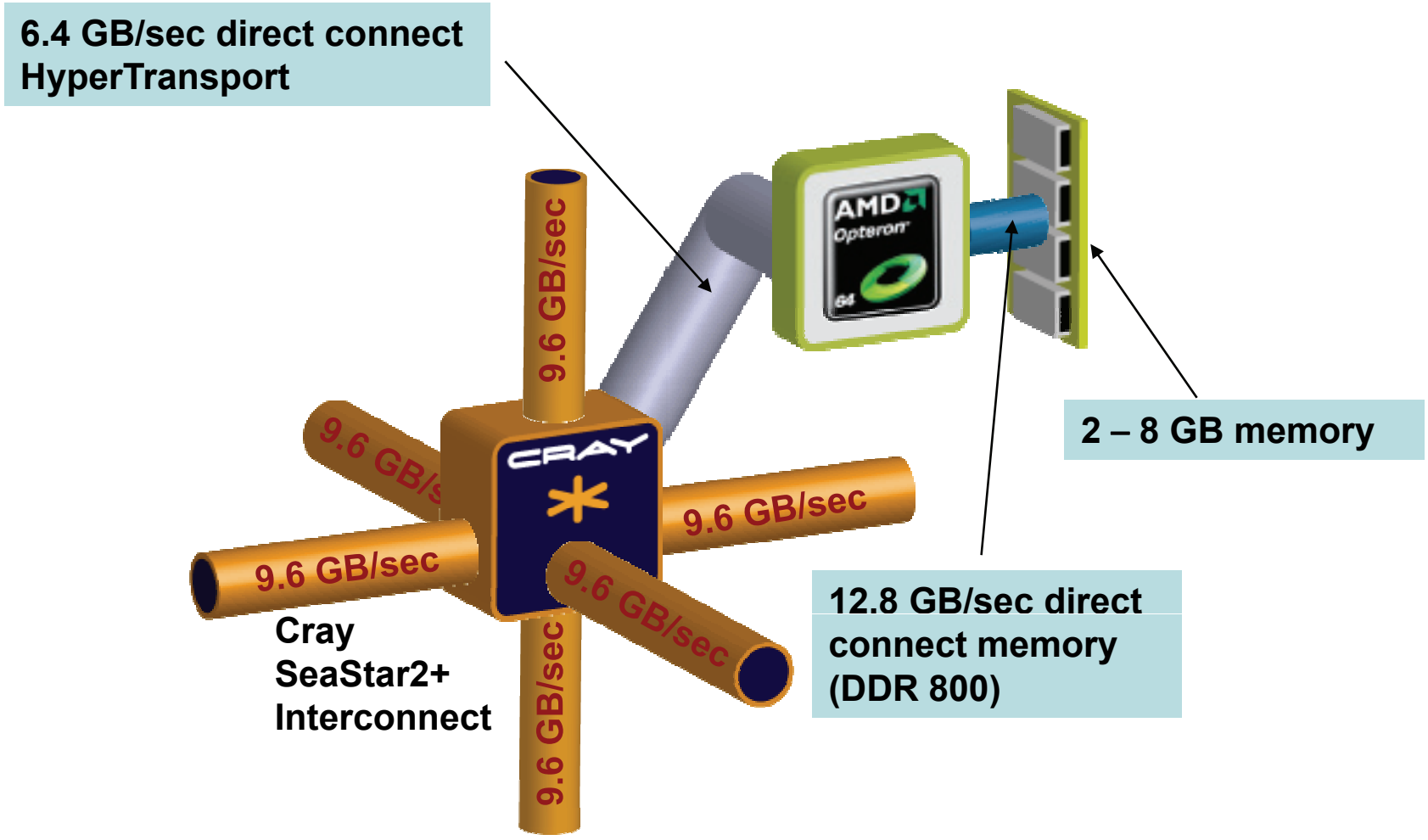


- Background
- Application
- Results
  - Effect of Optimization
  - Weak Scaling
  - Strong Scaling
    - Further Optimization
- **Summary**

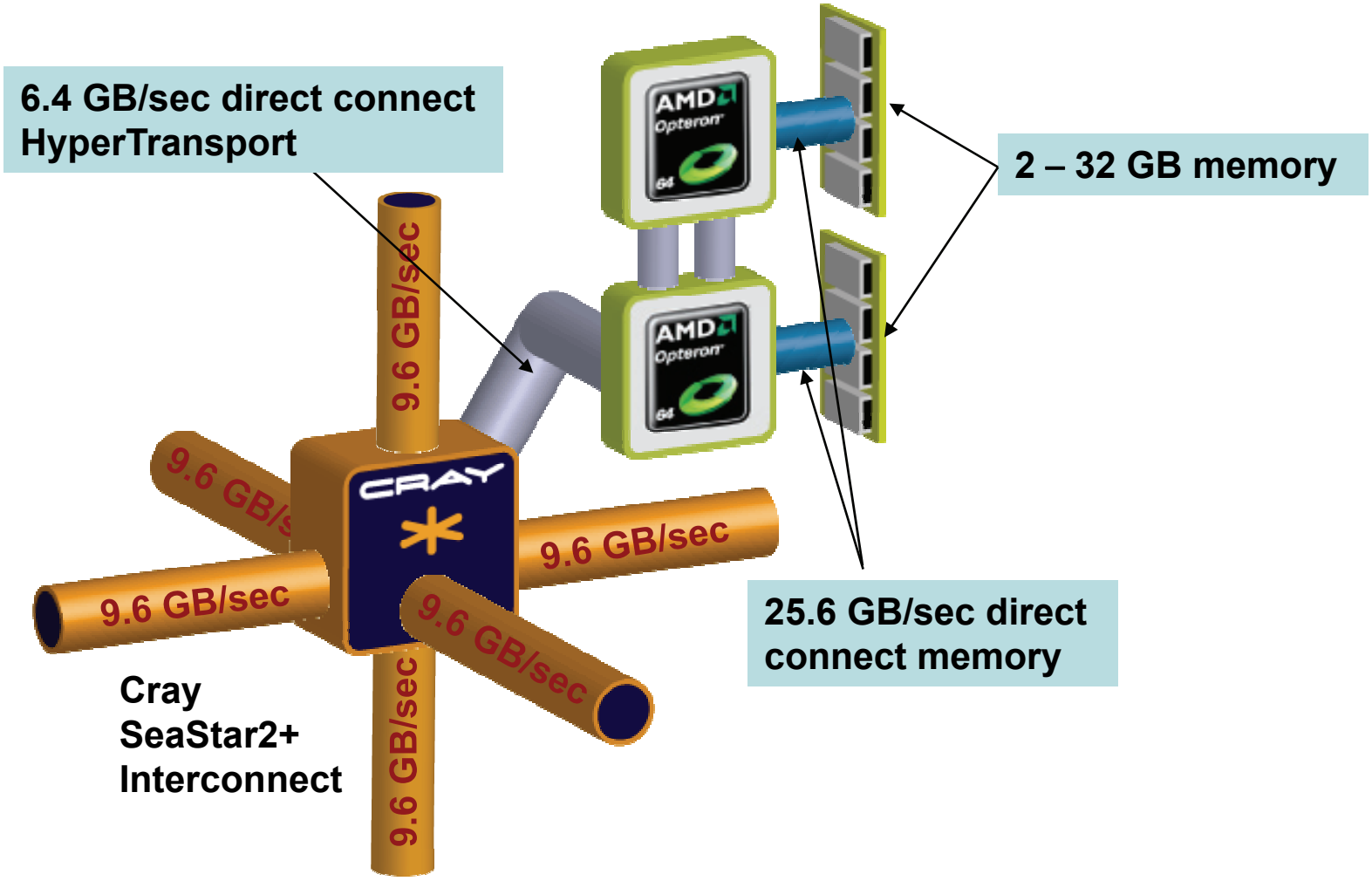
Summary

- (Multigrid + CG) solvers
  - 3D FVM code for GW flow through heterogeneous porous media
  - IC(0) smoother + ASDD
  - Geometric multigrid
- OpenMP/MPI Hybrid Parallel Programming Model
  - T2K/Tokyo
  - NUMA Policy, First Touch Data Placement , Sequential Reordering
  - Coarse-Grid Smoother
  - HB 4x4 provides the best performance
    - HB is better for strong-scaling cases
- Future Works
  - Robustness of MG: HID, Coarse-Grid Smoother
  - Cray XT4/XT5 in NERSC/LBNL

Cray XT4 Processing Element SeaStar2 Interconnect



Cray XT5 Processing Element SeaStar2 Interconnect



STREAM Benchmarks 16 cores, Flat MPI

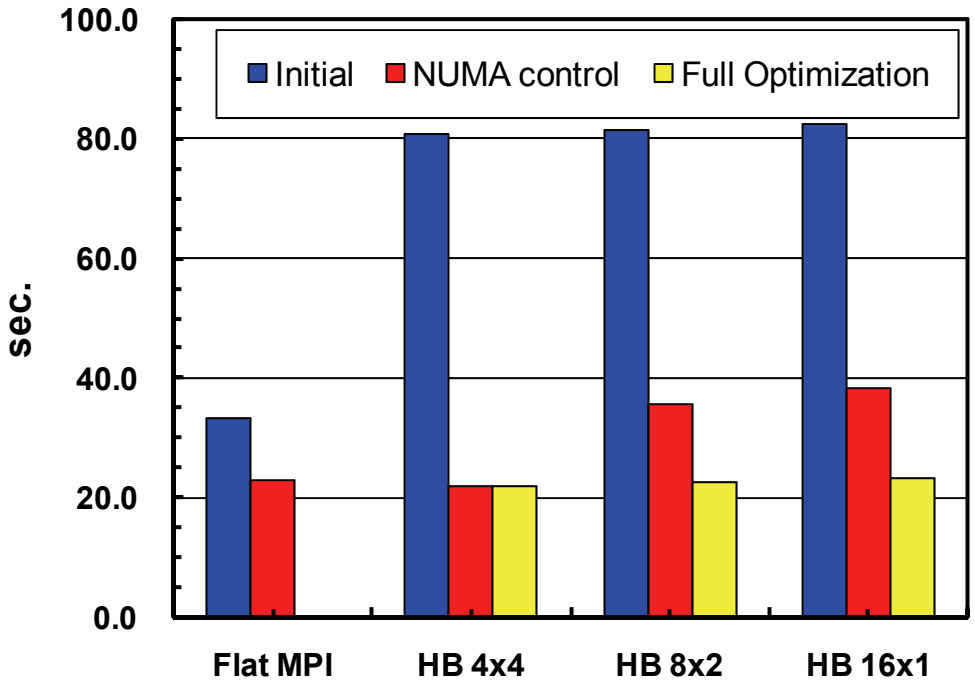
	T2K 2.3GHz	Franklin XT4/LBNL 2.3GHz	Hopper XT5/LBNL 2.4GHz
Copy	18408	38275	41280
Scale	18106	21708	28666
Add	18720	24516	31421
Triadd	19686	24577	31692
GeoFEM benchmark	138.1 sec. (1.00)	79.88 sec. (1.73)	70.23 sec. (1.97)

Effect of F.T. + Sequential Data Access

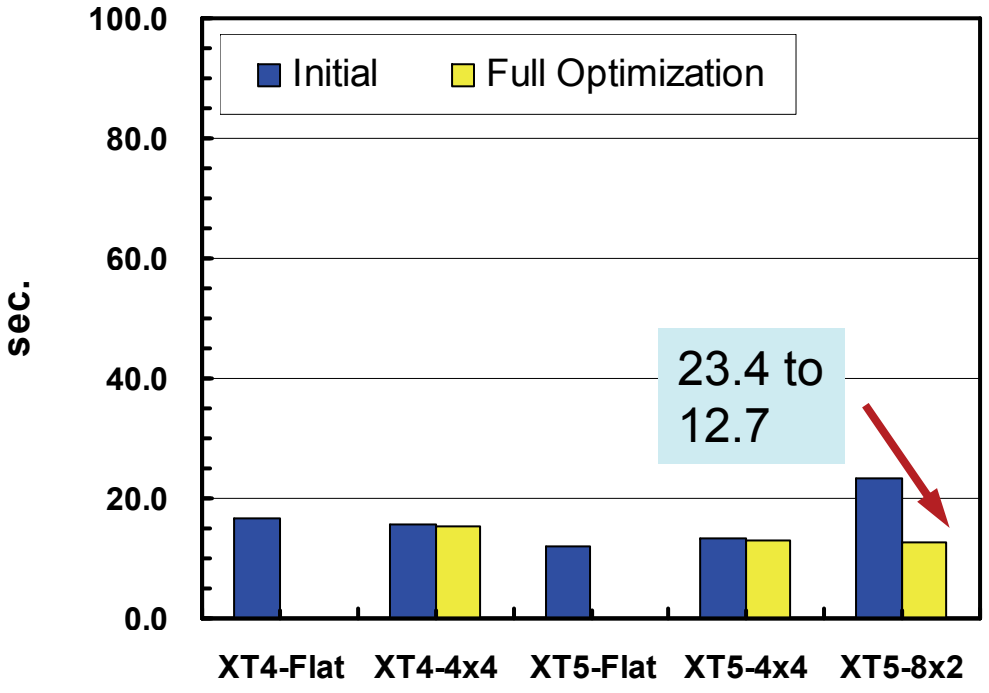
16,777,216= 64x64^3 cells, 64 cores, CM-RCM(2)

Time for Linear Solvers

T2K/Tokyo



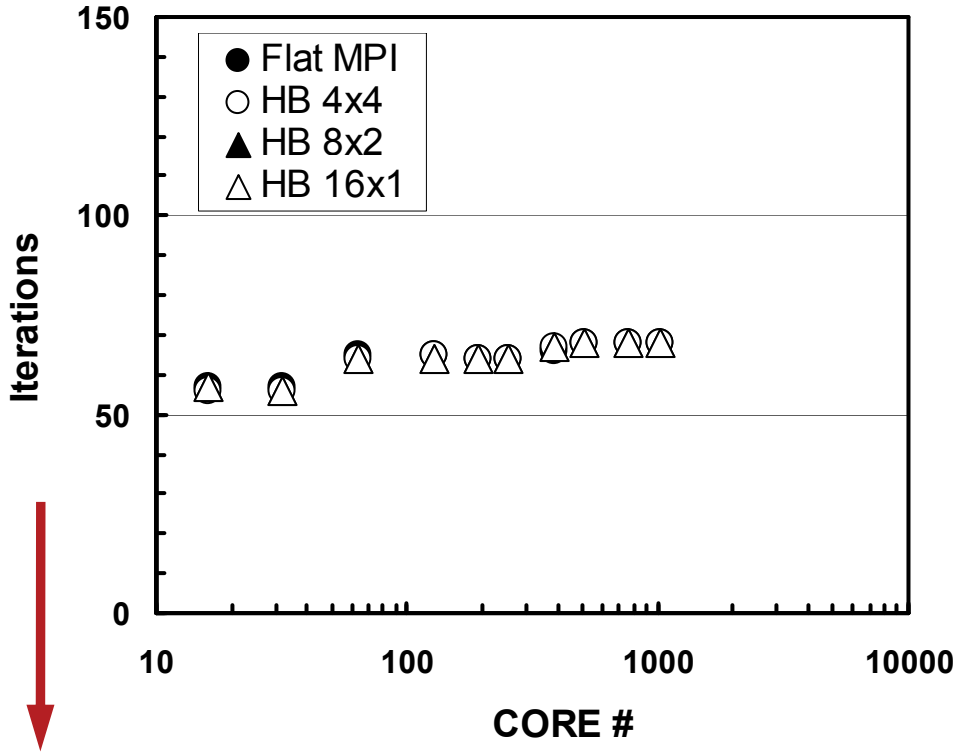
Cray XT4/5



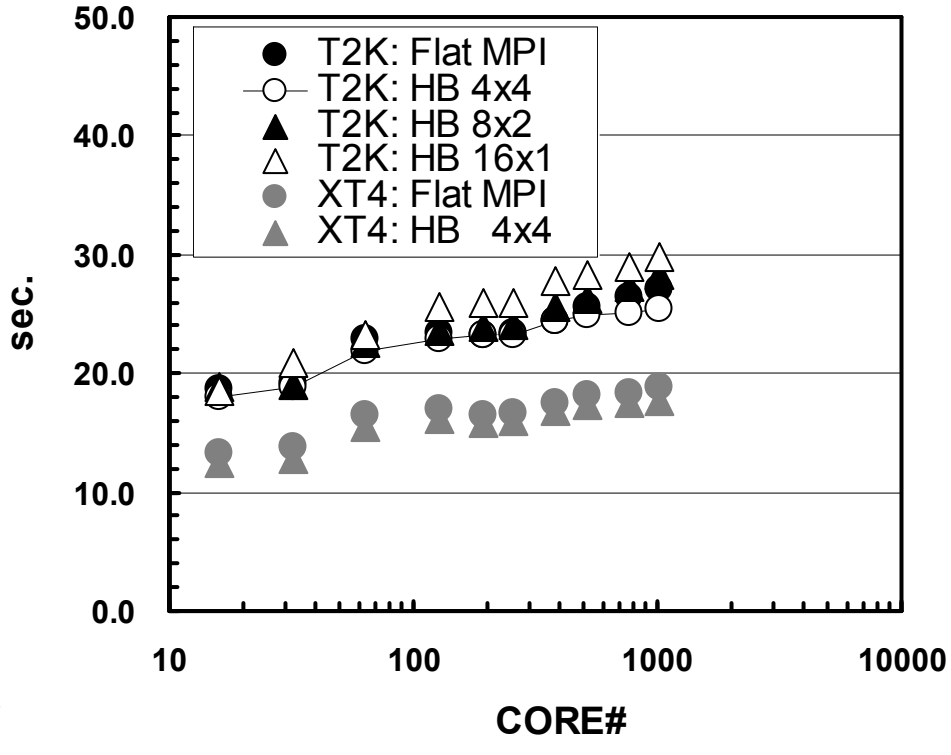
Weak Scaling

64^3cells/core, up to 268,435,456 cells

Iterations



sec.



Down is good