Efficient Parallel Multigrid Solver on Intel Xeon Phi Cluster

IXPUG
INTEL EXTREME PERFORMANCE USERS GROUP

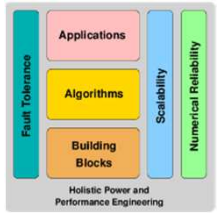
Kengo Nakajima[*1,*2], **Balazs Gerofi**[*2], **Yutaka Ishikawa**[*2], **Masashi Horikoshi**[*3]

*1: Information Technology Center, University of Tokyo,
*2: RIKEN R-CCS, *3: Intel

IXPUG HPC Asia 2021
January 22, 2021 (on-line)

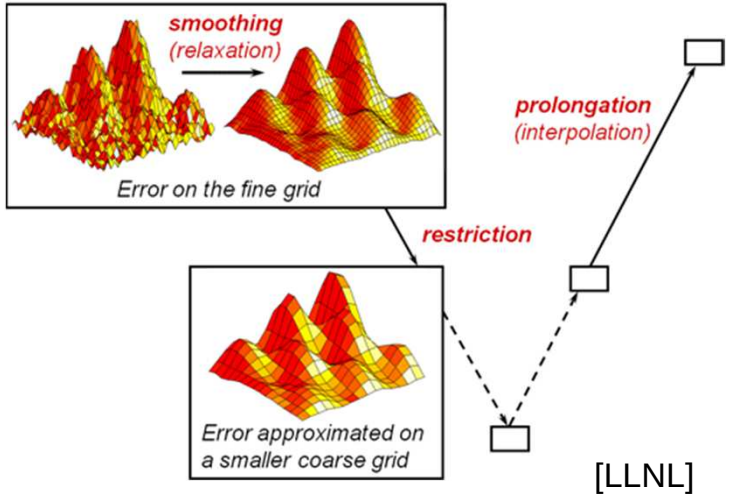
Acknowledgements

- JST/CREST
- DFG/SPPEXA
- **JHPCN (jh200037-NAH, jh200041-NAH)**
- JSPS Grant-in-Aid for Scientific Research (S): 19H05662
- JCAHPC



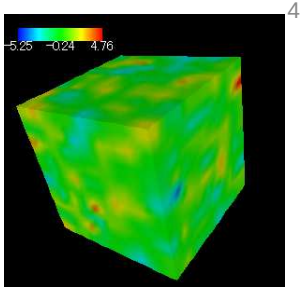
Features of Multigrid (MG) Methods

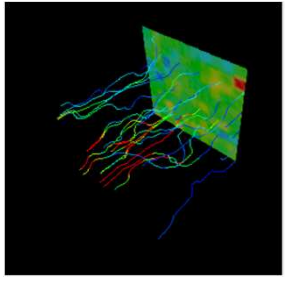
- ✓ **Scalable multilevel method for solving linear equations**
- ✓ GMG (Geometrical Multigrid) and AMG (Algebraic Multigrid)
- ✓ **Number of iterations until convergence for multigrid method is kept constant as the problem size changes, Comp. Time = O(N)**
- ✓ The parallel multigrid method is expected to be one of the most powerful tools on exa-scale systems.
- ✓ Applied to rather well-conditioned problems (e.g. Poisson's eqn's)
  - ☐ Many sophisticated methods for real-world applications are under development (next presentation)
- ✓ **MG is scalable, but there are many things to be done towards exascale computing**



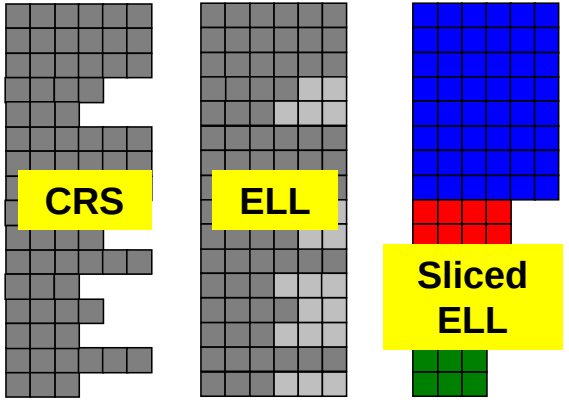
[LLNL]

pGW3D-FVM: Target Application
[KN IEEE ICPADS 2014] (Best Paper Award)





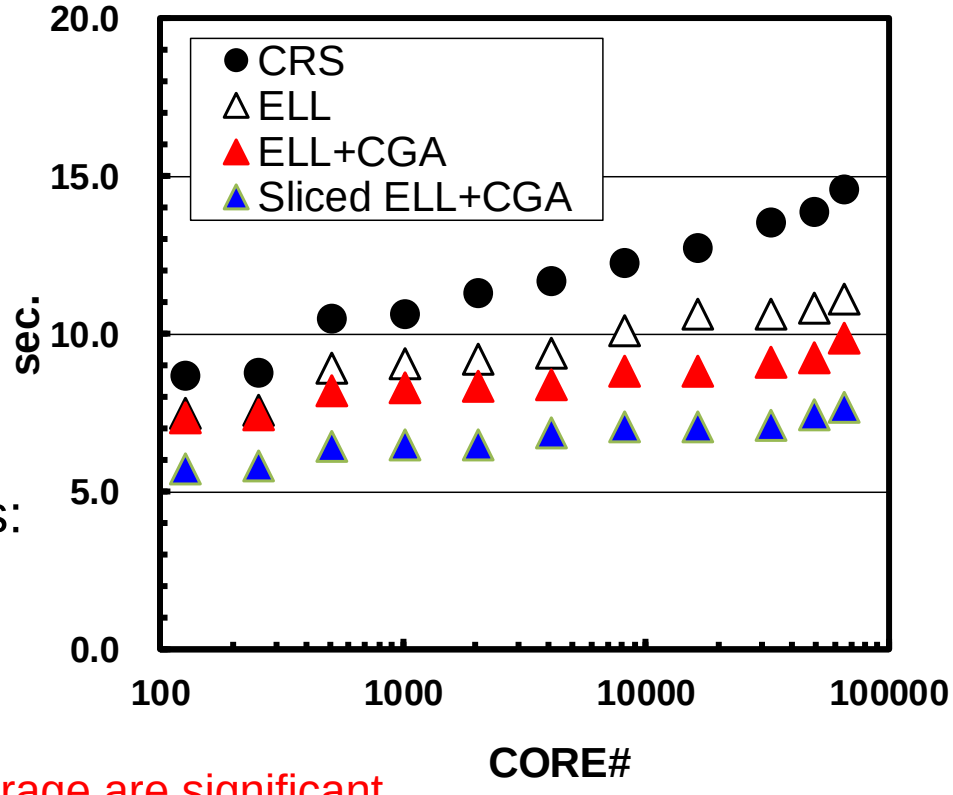
- 3D Groundwater Flow via Heterogeneous Porous Media
- Finite-Volume Method on Structured Cubic Voxel Mesh
- Poisson's equation $\nabla \cdot (\lambda(x,y,z)\nabla\phi) = q$
  - Randomly distributed water conductivity (λ)
  - $\lambda=10^{-5}\sim10^{+5}$, Average: 1.00
- **Conjugate Gradient preconditioned by Multigrid (MGCG)**
  - **Geometric Multigrid (GMG): Octree-based**
  - **IC(0) Smoother**
  - **V-Cycle**
  - **Additive Schwartz Domain Decomposition**
- **Sliced ELL for Storage of Sparse Matrices**



Nakajima, K., Optimization of Serial and Parallel Communications for Parallel Geometric Multigrid Method, Proceedings of the 20th IEEE International Conference for Parallel and Distributed Systems (ICPADS 2014) 25-32, Hsin-Chu, Taiwan, 2014

Effects of Sliced-ELL on MGCG/pGW3D-FVM [KN ICPADS 2014]

- Fujitsu PRIMEHPC FX10
  - up to 4,096-nodes, 655,536-cores
  - max 17,179,869,184 DOF
  - HB 8x2
- 1.9x performance@655,536-cores: (Sliced ELL+CGA) over CRS
- MGCG: Sparse Linear Solver
  - Typical Memory-Bound Procedure
  - Effects of Memory Access/Matrix Storage are significant

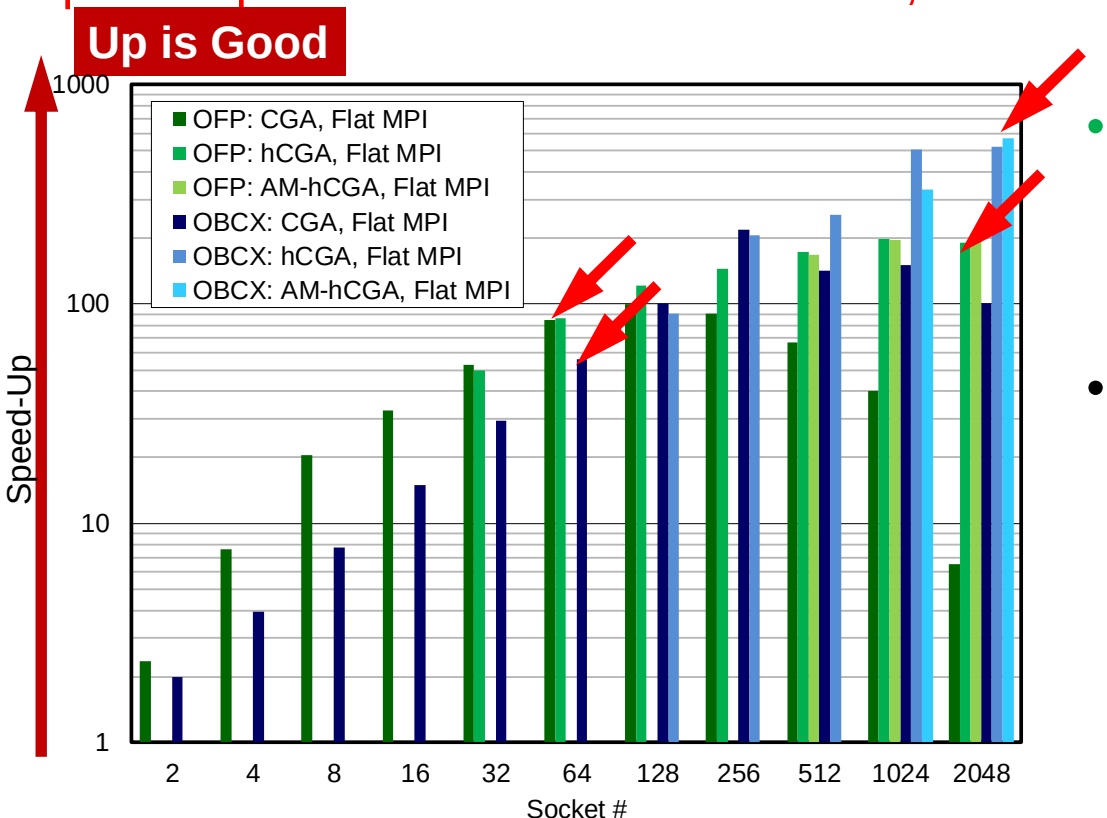


Nakajima, K., Optimization of Serial and Parallel Communications for Parallel Geometric Multigrid Method, Proceedings of the 20th IEEE International Conference for Parallel and Distributed Systems (ICPADS 2014) 25-32, Hsin-Chu, Taiwan, 2014

Strong Scaling: Parallel Performance

[KN IXPUG@HPC Asia 2020]

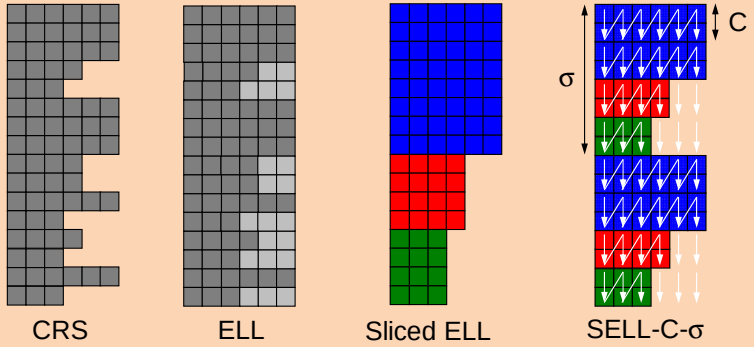
Speed-Up= 2.00 for OBCX at 2-Soc's, Flat MPI, CGA



- **OFP (KNL)** is rather faster if Soc.# is smaller, but **OBCX (CLX)** outperforms at more Soc.#
- Effects of *h*-CGA/AM-*h*CGA is very significant on OBCX with more than 1,024 sockets.

Nakajima, K., Parallel Multigrid Method on Multicore/Manycore Clusters, IXPUG@HPC Asia 2020

Future Works [KN IXPUG@HPC Asia 2020]



- Pipelined Algorithms
- SELL-C-σ
- Lower/Mixed Precision
- Preliminary Results: Double/Single Precision
  - Number of Iterations does not change
  - Computation Time for MGCG
    - 0.85 for OFP (Single/Double Ratio)
    - 0.60 for Intel BDW Cluster (Single/Double Ratio)
  - Further Vectorization Needed on OFP
- Larger Problems using More Cores

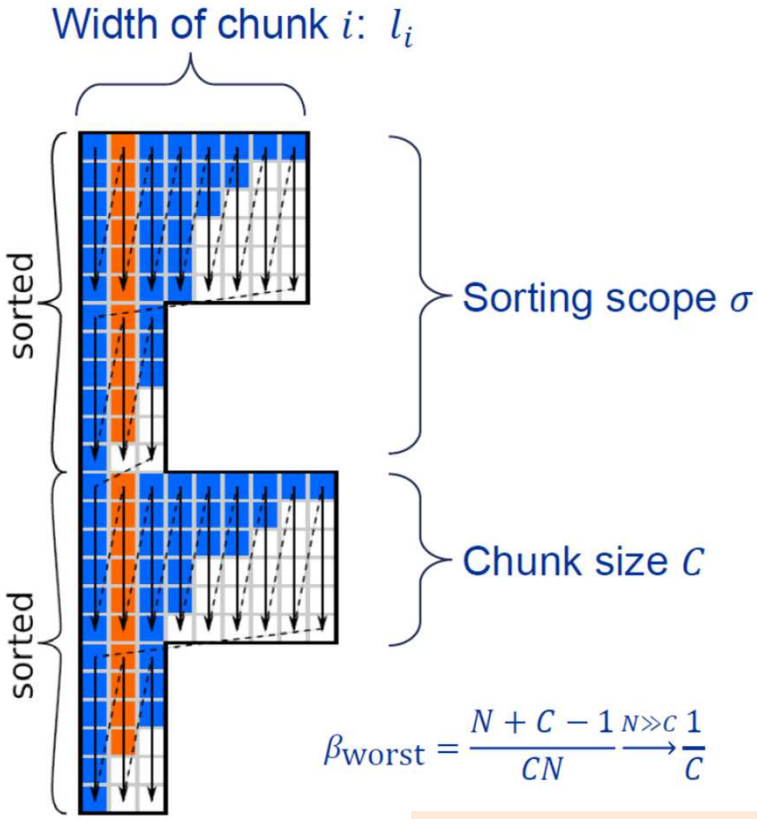
Nakajima, K., Parallel Multigrid Method on Multicore/Manycore Clusters, IXPUG@HPC Asia 2020

Constructing SELL-C-σ

1. Pick chunk size C (guided by SIMD/T widths)
2. Pick sorting scope σ
3. Sort rows by length within each sorting scope
4. Pad chunks with zeros to make them rectangular
5. Store matrix data in "chunk column major order"

"Chunk occupancy": fraction of "useful" matrix entries

$$\beta = \frac{N_{nz}}{\sum_{i=0}^{N_c} C \cdot l_i}$$

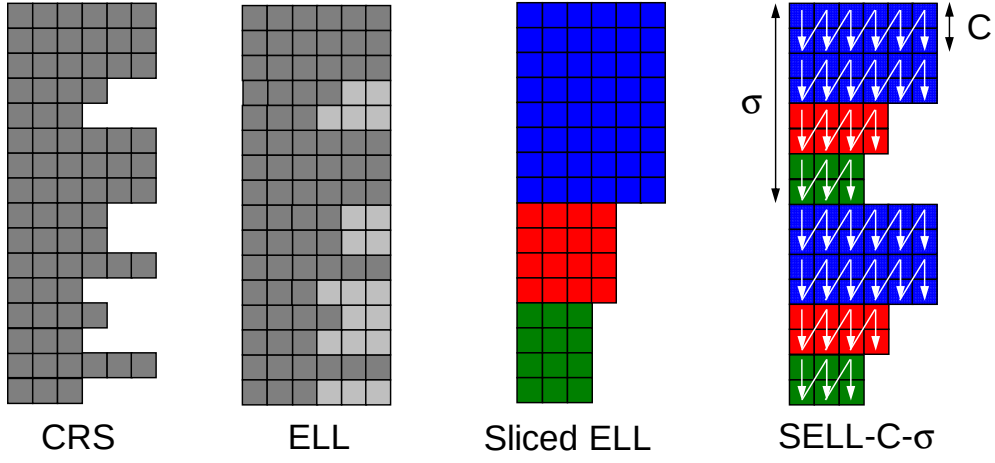


SELL-6-12
β=0.66

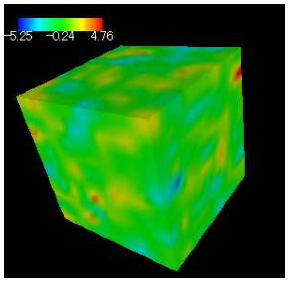
[Kreutzer, Hager, Wellein, SIAM SISC 2014]

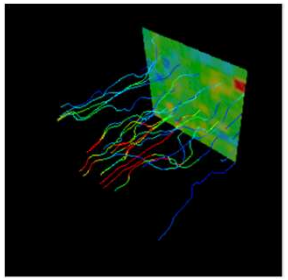
Purpose of the Present Work: SELL-C-σ

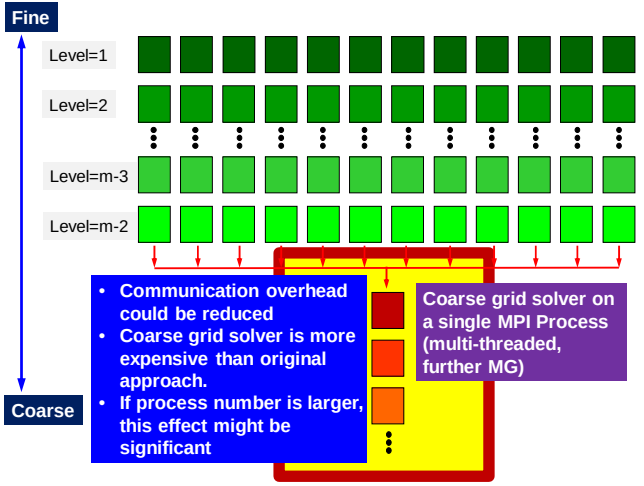
- **Applying SELL-C-σ to MGCG Solvers in pGW3D-FVM**
  - **Currently with Sliced ELL**
- Sliced ELL and SELL-C-σ have been mostly applied to SpMV, or Gauss-Seidel Iterative Solvers/Smoothers
- Implementation to Forward/Backward Substitution in ILU-type Smoothers is very difficult
  - First example of Sliced ELL [KN IEEE ICPADS 2014]
  - **This is the first example of SELL-C-σ**



Overview of the Present Work



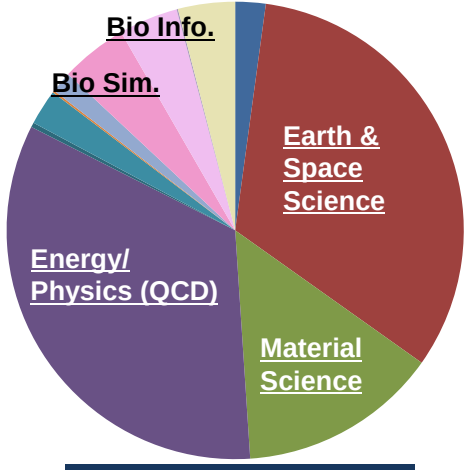




- pGW3D-FVM, Weak Scaling
  - CGA（Coarse Grid Aggregation）: Single Level
- Oakforest-PACS (OFP), ~1,024 nodes
  - Flat, MC-DRAM only
  - 64-cores on each node
    - Flat MPI, HB 2x32 (2-threads x 32-proc's)
    - HB 4x16, HB 8x8
  - Problem Size: 64x32x32 on each core (max: 4,294,967,296 DOF), Best for 5-Runs
  - IHK/McKernel
- Comparison between (CRS, Sliced ELL) and SCS (SELL-C-σ, C=σ=8)
  - 64-bit (Double Precision) x 8 = 512 bit
  - SCS: Switching to Sliced-ELL if the problem size is smaller than (C (=8)) for each color/thread
  - **Sliced ELL only for Coarse Grid Solver**
  - 20% improvement is expected by SCS over Sliced ELL, based on preliminary results

Target System: Oakforest-PACS (OFP)

- **Intel Xeon Phi (Knights Landing, KNL), OPA**, Fujitsu
- IHK/McKernel
- 8,208 nodes, 25+PF, 22nd in TOP 500 (November 2020)
  - 2nd Largest KNL System in Asia
- Operated by JCAHPC (U.Tsukuba & U.Tokyo)



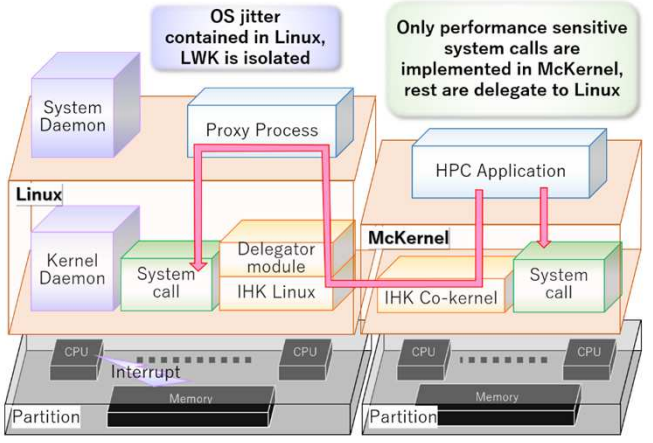
Usage in FY.2019

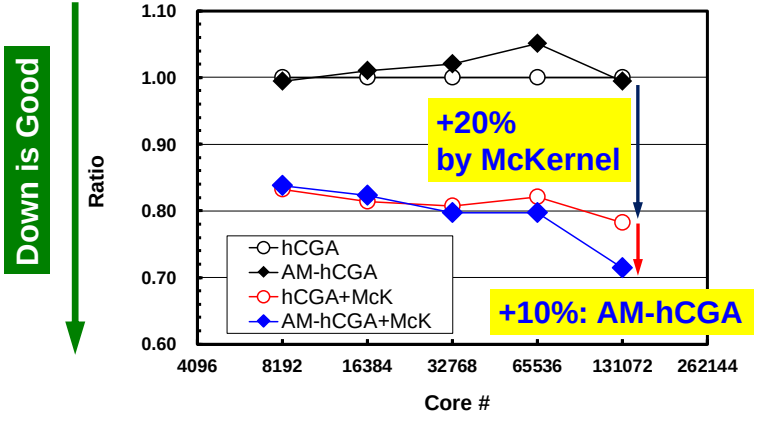
- Engineering
- Earth/Space
- Material
- Energy/Physics
- Info. Sci. : System
- Info. Sci. : Algrorithms
- Info. Sci. : AI
- Education
- Industry
- Bio
- Bioinformatics
- Social Sci. & Economics
- Data

JCAHPC 筑波大学 University of Tsukuba 東京大学 THE UNIVERSITY OF TOKYO

IHK/McKernel

- Lightweight Multi-Kernel OS for HPC by RIKEN R-CCS [Gerofi et al. IPDPS 2016]
- McKernel implements only a small set of performance sensitive system calls and the rest of the OS services are delegated to Linux: (Linux+McKernel)
- Same binary on pure Linux can be used
- Lower Noise/Communication Overhead than the Pure Linux Environment
- Significant improvement of MGCG solver (20%) on OFP with 131,072-cores [KN, BG, MH, YI, ScalA19@SC19]
- to be installed on the Fugaku

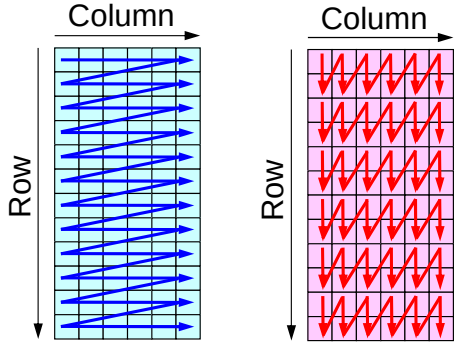




Forward Substitution: 2nd Color of CM-RCM(2)

Sliced-ELL Row-wise

```
!$omp parallel private (icol,…)
(…)
    icol= NCOLORtot
!$omp do
    do icel= IND(icol-1,lev)+1,IND(icol,lev)
      VAL= B(icel)
      do j= 1, 6
        VAL= VAL - AL(j,icel)*X(IAL(j,icel))
      enddo
      Xmg(icel)= VAL*DDmg(icel)
    enddo
!$omp end parallel
```



SCS-a Row-wise

```
!$omp parallel private (icol,…)
(…)
    icol= NCOLORtot
!$omp do
    do ip= 1, PEsmpTOT
      ib0= BLOCKindex(ip-1,icol,lev)+1
      ib1= BLOCKindex(ip  ,icol,lev)
    do ib= ib0, ib1
      ic0= (ib-im2-1)*8 + im1
!$omp simd
      do k= 1, 8
        icel= ic0 + k
        VAL= B(icel)
        do j= 1, 6
          isL= SELL_CSL(ib-1)+(j-1)*8 + k
          VAL= VAL - ALs(isL)*X(IALs(isL))
        enddo
        X(icel)= VAL*DD(icel)
      enddo
    enddo
    enddo
!$omp end parallel
```

SCS-b Column-wise more expensive

```
!$omp parallel private (icol,…)
(…)
    icol= NCOLORtot
!$omp do
    do ip= 1, PEsmpTOT
      ib0= BLOCKindex(ip-1,icol,lev)+1
      ib1= BLOCKindex(ip  ,icol,lev)
    do ib= ib0, ib1
      ic0= (ib-im2-1)*8 + im1
!$omp simd
      do k= 1, 8
        icel= ic0 + k
        W(icel)= B(icel)
      enddo
      do j= 1, 6
!$omp simd
      do k= 1, 8
        icel= ic0 + k
        isL= SELL_CSL(ib-1)+(j-1)*8 + k
        W(icel)= W(icel)-ALs(isL)*X(IALs(isL))
      enddo
      enddo
!$omp simd
      do k= 1, 8
        icel= ic0 + k
        X(icel)= W(icel)*DD(icel)
      enddo
    enddo
    enddo
!$omp end parallel
```

Coalesced，CM-RCM(2)

- CM-RCM with 2 colors: CM-RCM(2)
  - Number of iterations will increase with 2-colors, compared to RCM (current method)
  - Performance on OFP with multithreading is better with fewer colors
    - Loop length
  - Implementation of SELL-C-σ is easier than RCM
- Coloring part is not parallelized, but implementation of CM-RCM(2) is easy

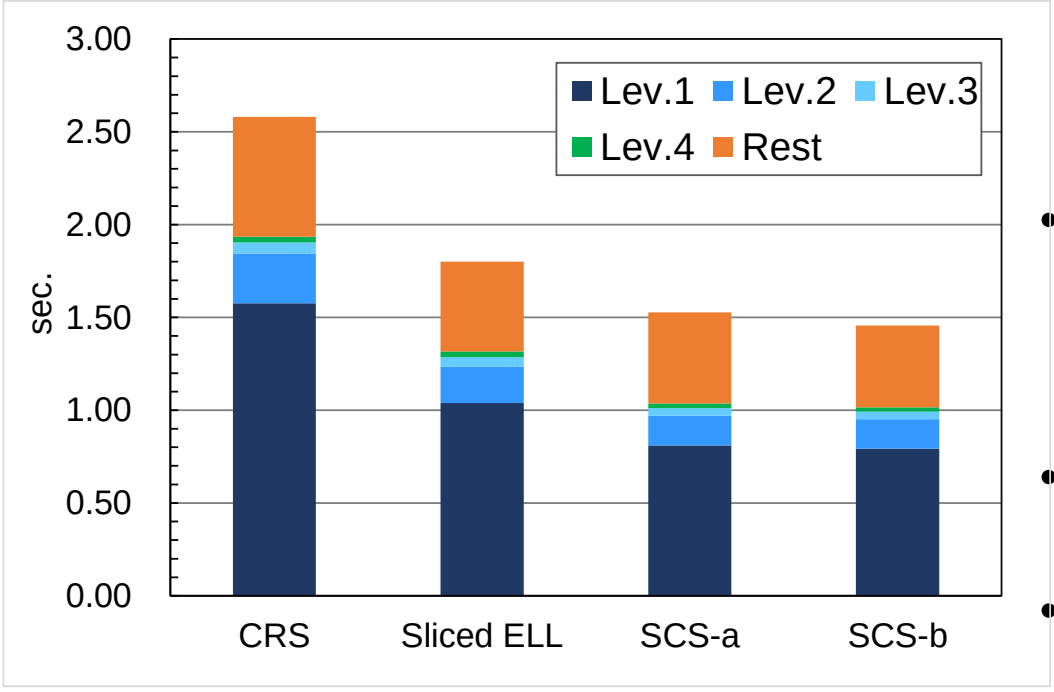
Red-Black，2色のMC

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

CM-RCM（2），Coalesced

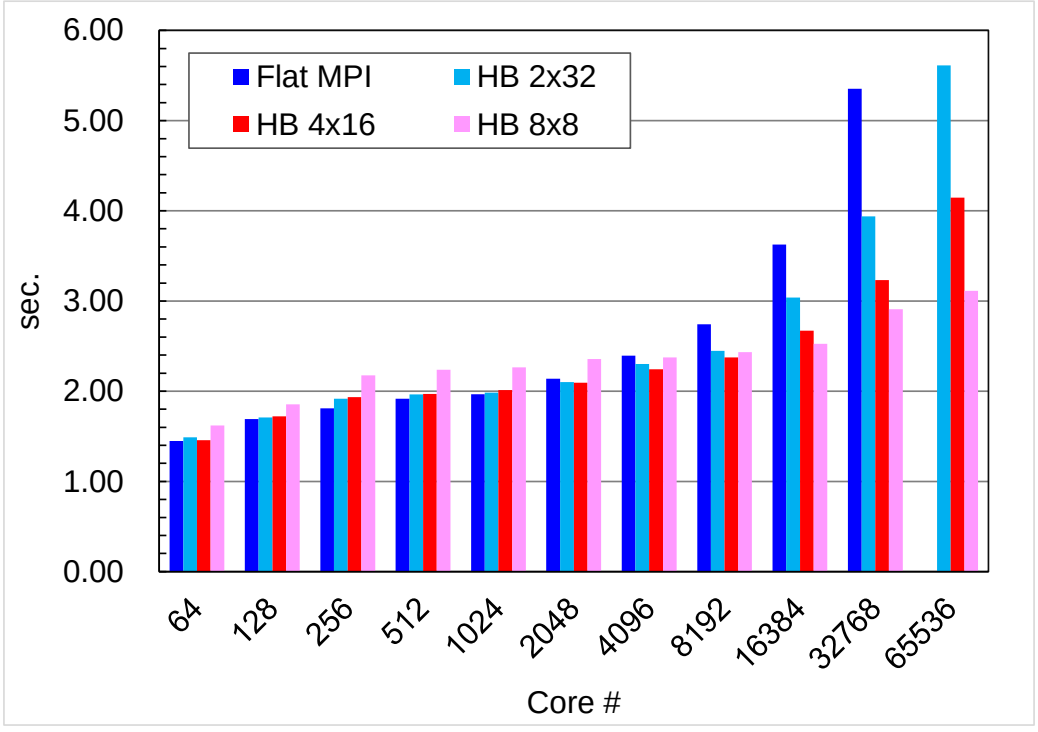
13	42	7	37	3	34	1	33
49	14	43	8	38	4	35	2
21	50	15	44	9	39	5	36
56	22	51	16	45	10	40	6
27	57	23	52	17	46	11	41
61	28	58	24	53	18	47	12
31	62	29	59	25	54	19	48
64	32	63	30	60	26	55	20

Results: Time for MGCG 8-nodes, HB 4×16



- SCS up to the 5th Level
- Rest
  - Coarser Level Smoothers
  - CG except MG (SpMV etc.)
  - Communication
  - Coarse Grid Solver
- Improvement over CRS
  - Sliced ELL：43.3%
  - SCS-a：68.9%
  - SCS-b：77.1%
- Level-1 (Finest)
  - 51.5%, 94.7%, 98.6%
- Sliced ELL⇒SCS
  - SCS-a：28.5%
  - SCS-b：31.1%

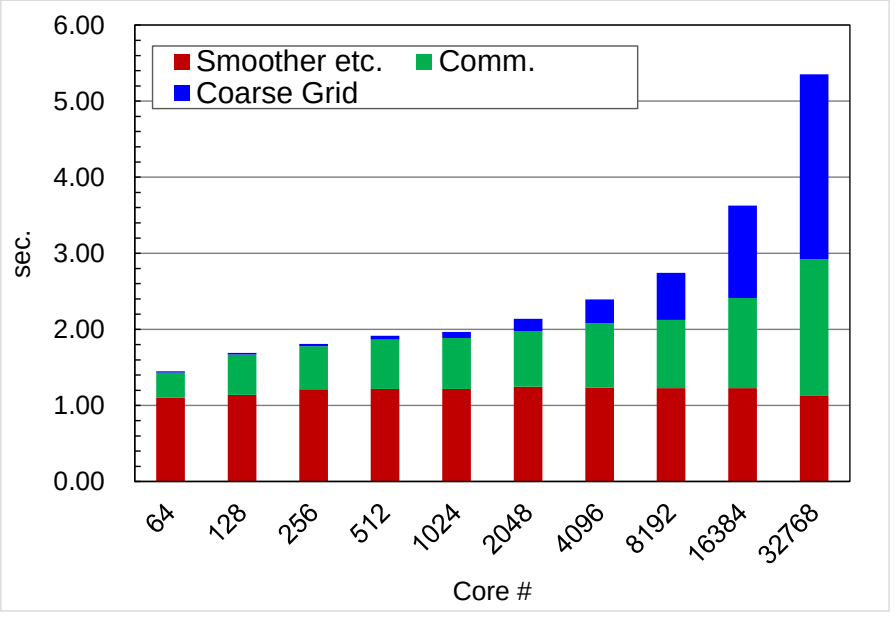
Weak Scaling: up to 1,024-nodes (65,536-cores) SCS-b, Time for MGCG, Down is Good



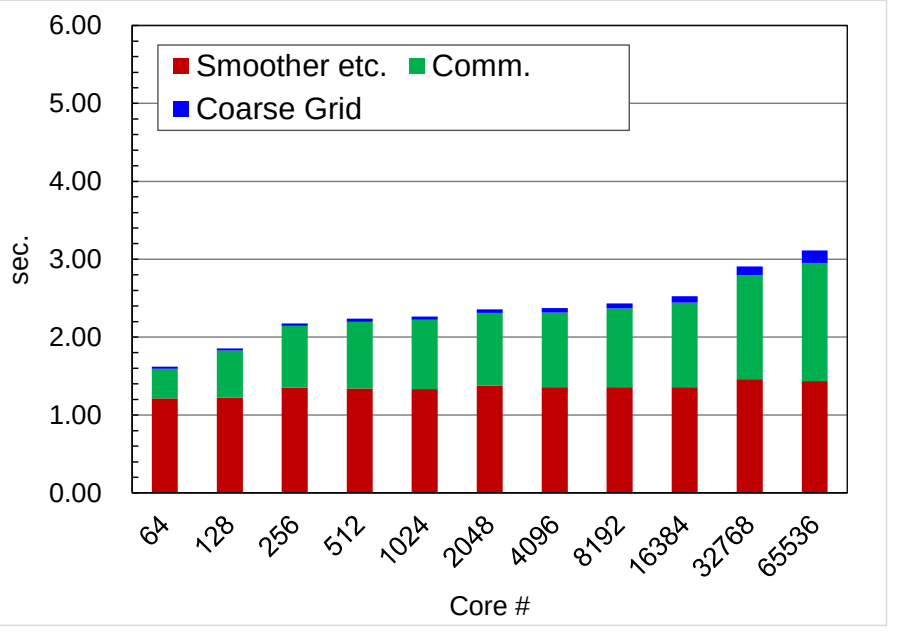
- Flat MPI is better with fewer nodes
- Flat MPI is slower with larger number of nodes
  - Effects of Coarse Grid Solver
  - Problem size of the Coarse Grid Solver is proportional to # of MPI Processes
    - # of MPI processes in Flat MPI is larger than HB cases

Weak Scaling: up to 1,024-nodes (65,536-cores)
SCS-b, Time for MGCG, Down is Good

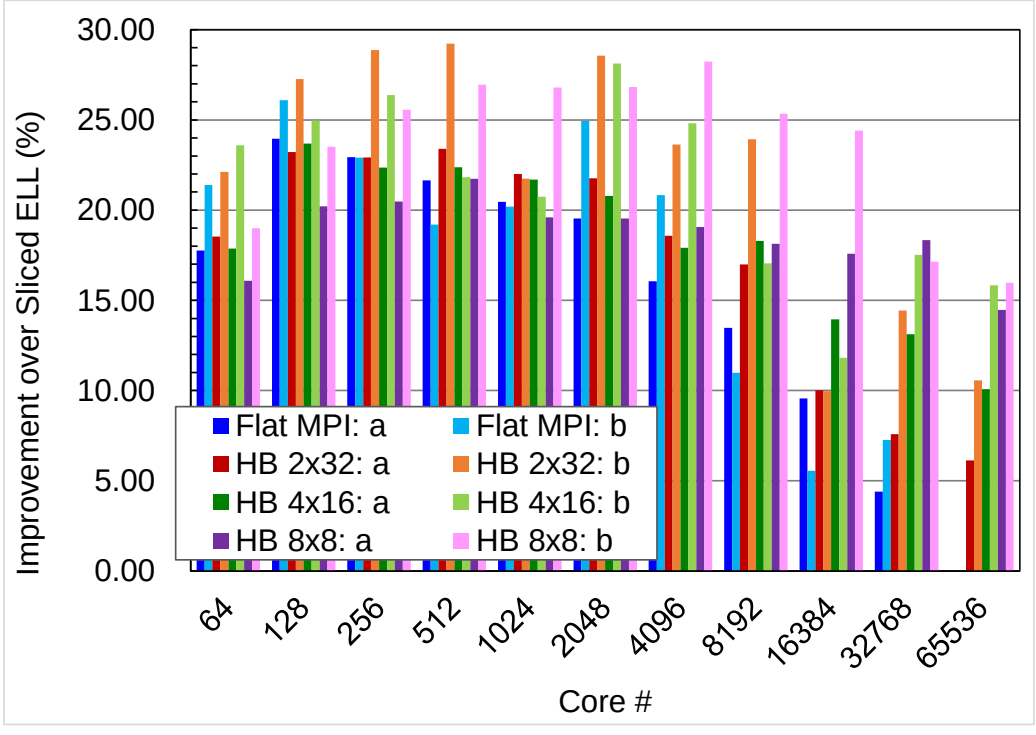
Flat MPI



HB 8x8

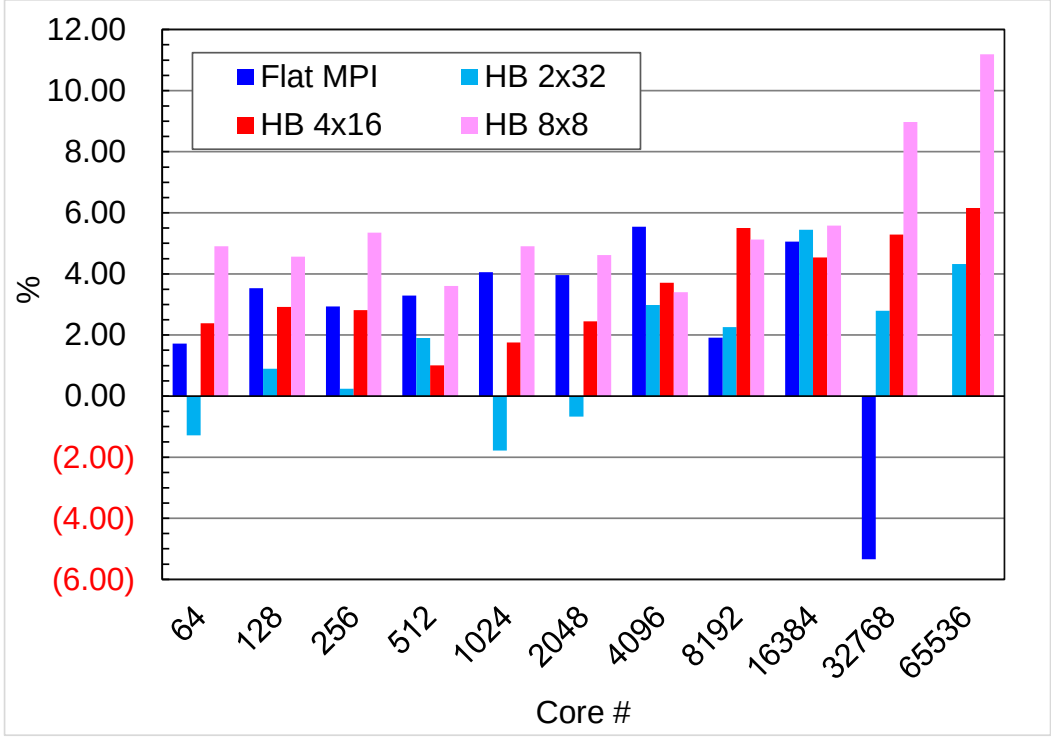


Ratio of Improvement over Sliced ELL SCS-a, SCS-b



- Generally 20+ %
- Although cost of computations in SCS-b is larger than that in SCS-a, performance of SCS-b is better
- Improvement ratio decreases for larger number of nodes, because Sliced-ELL is applied to the Coarse Grid Solver
  - More significant in Flat MPI

Effect of McKernel：SCS-b Performance Improvement



- Random behavior, especially in Flat MPI
- Improvement of performance is more significant for larger nodes, as was expected
- HB 8x8 was most effective @1,024-nodes (not Flat MPI)
- Problem size per core is large (close to the limit of MC-DRAM)
  - Effects will be more significant for smaller problems

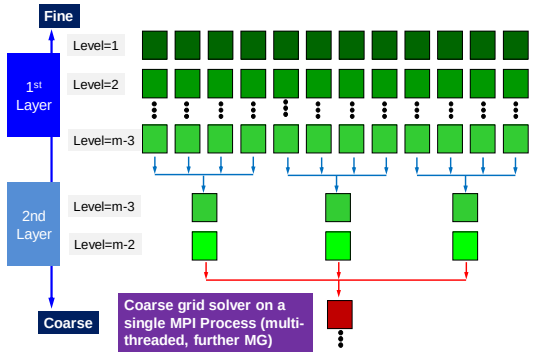
Summary

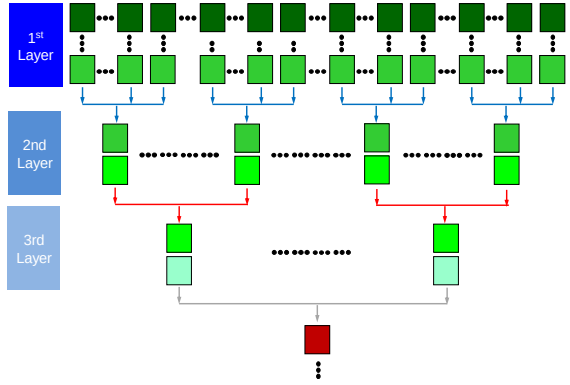
- Effects of SELL-C-σ are significant

- Performance Improvement with 30% on OFP compared to Sliced ELL
- Good Scalability up to 1,024 nodes for various types of parallel programming models
- IHK/McKernel

Future Works

- *h*CGA, AM-*h*CGA
- Coarse Grid Solver with SELL-C-σ
- Automatic selection of optimum method for matrix storage, optimum number of threads and etc. varies according to level of multigrid procedures
  - Optimum parameters should be adopted, including C/σ
  - Previous work on switching between single and multi threads [KN VECPAR 2012]
- Single Precision, Mixed Precision
  - Preliminary investigations show SP/MP provides efficient and robust results
- Extension to Various Types of Architectures
  - Intel Xeon CPU's, A64FX: Optimum C/σ should be larger than 8 (e.g. 32, 64…) [Alappat, Hager, Wellein et al. SC20 WS] (developer of SELL-C-σ)





一般社団法人 情報処理学会
Information Processing Society of Japan

sighpc

RIKEN

R-CCS

HPC Asia 2022 in Kobe

KOBE UNIVERSITY

Stepping forward to the Post Moore Era together !!

http://sighpc.ipsj.or.jp/HPCAsia2022/




Looking forward to seeing all of you in Kobe next year !!

Kengo Nakajima
General Chair
The University of Tokyo
RIKEN R-CCS

Miwako Tsuji
General Vice-Chair
RIKEN R-CCS