Preconditioned Iterative Solvers in ppOpen-HPC/pKOpen-HPC for ESSEX-II

Kengo Nakajima
Information Technology Center
The University of Tokyo

Parallel Programming Models – Productivity and Applications (2nd Edition)
University of Versailles - October 18, 2017

Acknowledgements

- Sponsors
  - ✓ CREST-JST, Japan
  - ✓ SPPEA-DFG, Germany
- Collaborators, Colleagues
  - ✓ Akihiro Ida (ITC/U.Tokyo)
  - ✓ Masatoshi Kawai (ITC/U.Tokyo)
  - ✓ Takahiro Katagiri (Nagoya U.)
  - ✓ Masaki Satoh (AORI/U.Tokyo)
  - ✓ Takashi Furumura (ERI/U.Tokyo)
  - ✓ Takeshi Iwashita (Hokkaido U.)
  - ✓ Satoshi Ohshima (Kyushu U.)
  - ✓ Toshihiro Hanawa (ITC/U.Tokyo)
  - ✓ Tetsuya Hoshino (ITC/U.Tokyo)
  - ✓ Hajime Yamamoto (Taisei)

- **ppOpen-HPC**
- ESSEX-II: Preconditioned Iterative Solvers for Eigenvalue Problems in Quantum Physics
- Other Collaborations in ESSEX-II
  - SELL-C-σ
  - CRAFT

ppOpen-HPC: Overview

- Application framework with automatic tuning (AT)
  - ✓ "pp" : post-peta-scale
- Five-year project (FY.2011-2015) (since April 2011)
  - ✓ P.I.: Kengo Nakajima (ITC, The University of Tokyo)
  - ✓ Part of "Development of System Software Technologies for Post-Peta Scale High Performance Computing" funded by JST/CREST (Supervisor: Prof. Mitsuhisa Sato, RIKEN AICS)
- Target: Oakforest-PACS (Original Schedule: FY.2015)
  - ✓ could be extended to various types of platforms
- Team with 7 institutes, >50 people (5 PDs) from various fields: Co-Design
- Open Source Software
  - ✓ http://ppopenhpc.cc.u-tokyo.ac.jp/
  - ✓ English Documents, MIT License

独立行政法人 科学技術振興機構 Japan Science and Technology Agency

CREST 戦略的創造研究推進事業 Core Research for Evolutionary Science and Technology

Oakforest-PACS

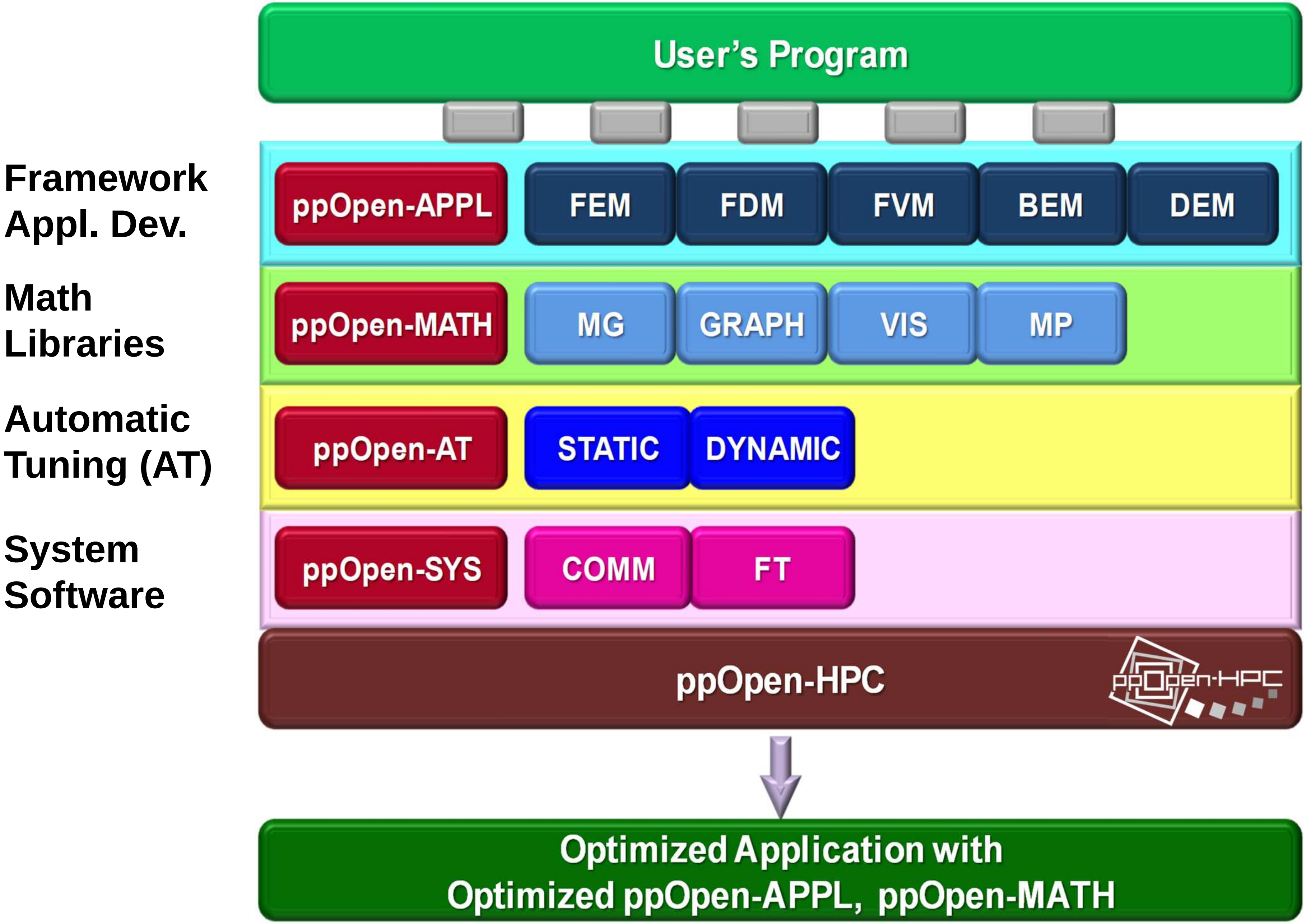
The Fastest Supercomputer in Japan

- Full Operation started on December 1, 2016
- 8,208 Intel Xeon/Phi (KNL), 25 PF Peak Performance
  - Fujitsu
- **TOP 500 #7 (#1 in Japan), HPCG #5 (#2) (June 2017)**
- **JCAHPC: Joint Center for Advanced High Performance Computing)**
  - University of Tsukuba
  - University of Tokyo
    - New system will installed in Kashiwa-no-Ha (Leaf of Oak) Campus/U.Tokyo, which is between Tokyo and Tsukuba
  - http://jcahpc.jp

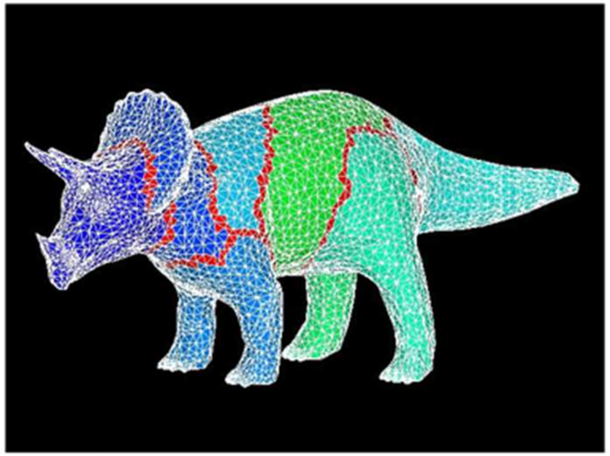
JCAHPC

東京大学 THE UNIVERSITY OF TOKYO

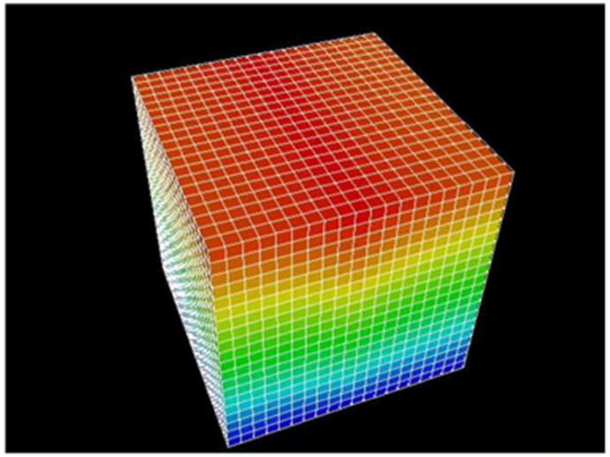
筑波大学 University of Tsukuba


User's Program
Framework
Appl. Dev.
ppOpen-APPL
FEM
FDM
FVM
BEM
DEM
Math
Libraries
ppOpen-MATH
MG
GRAPH
VIS
MP
Automatic
Tuning (AT)
ppOpen-AT
STATIC
DYNAMIC
System
Software
ppOpen-SYS
COMM
FT
ppOpen-HPC
Optimized Application with
Optimized ppOpen-APPL, ppOpen-MATH

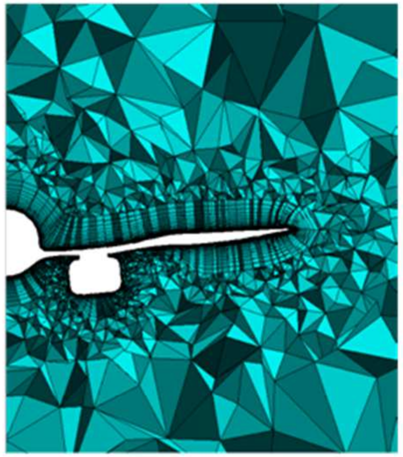
ppOpen-HPC covers …



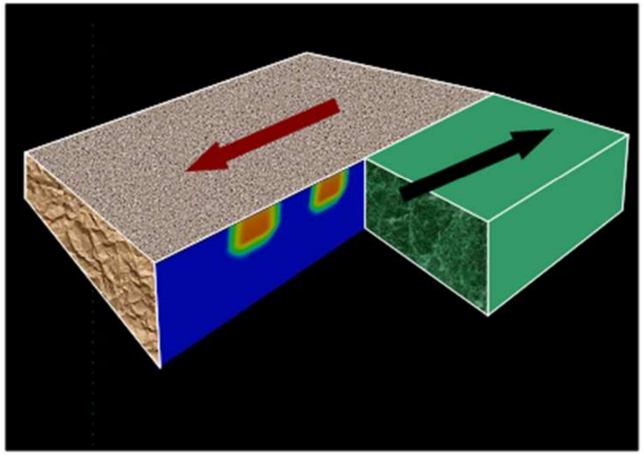
FEM
Finite Element Method



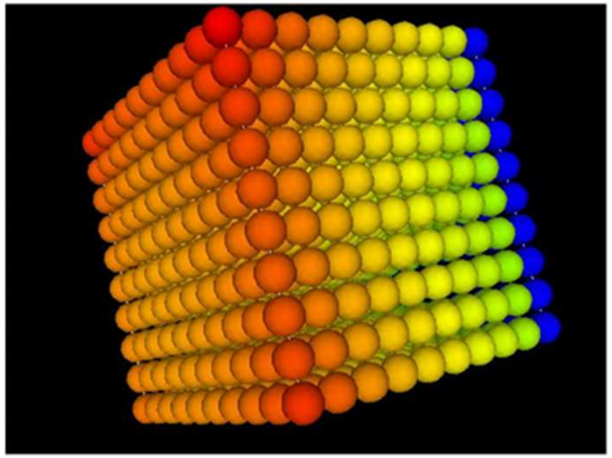
FDM
Finite Difference Method



FVM
Finite Volume Method



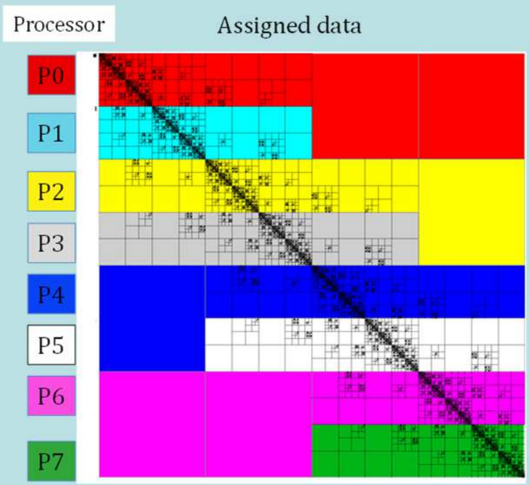
BEM
Boundary Element Method



DEM
Discrete Element Method

Featured Developments

- ppOpen-AT: AT Language for Loop Optimization
- HACApK library for H-matrix comp. in ppOpen-APPL/BEM (OpenMP/MPI Hybrid Version)
  - First Open Source Library by OpenMP/MPI Hybrid
- ppOpen-MATH/MP (Coupler for Multiphysics Simulations, Loose Coupling of FEM & FDM)
- **Sparse Linear Solvers**



Assign a^c to P_k if $S(a^c{}_{ij})$ in $R(P_k)$

A : Whole matrix

A_{IJ} : The entry in the I-th row and the J-th column of A

a^c: A small submatrix of A

$a^c{}_{ij}$: The entry in the i-th row and the j-th column of a^c

N : Number of rows of A

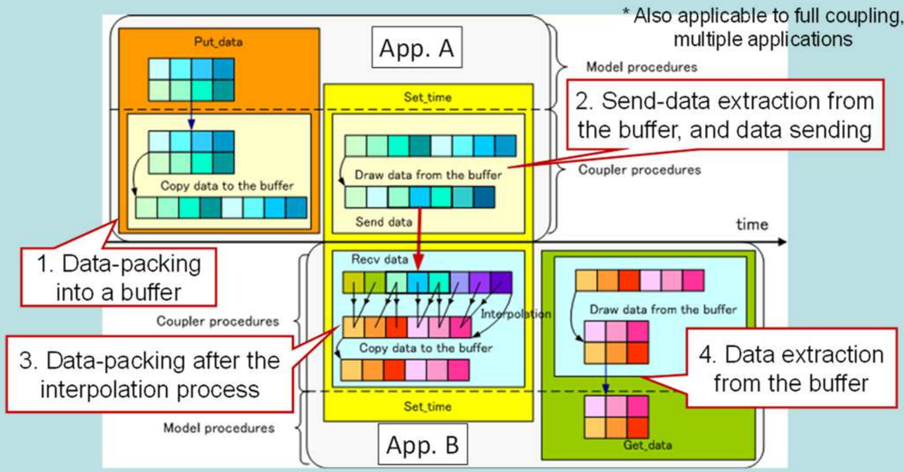
n : Number of processors

P_k : The k-th processor

$l_k : 1 = l_0 < \cdots < l_k < \cdots < l_{n+1} = N + 1$

$R : R(P_k) = \{ i \mid l_k \le i < l_{k+1} \}$

$S : S(a^c{}_{ij}) = I$ when $a^c{}_{ij} = A_{IJ}$

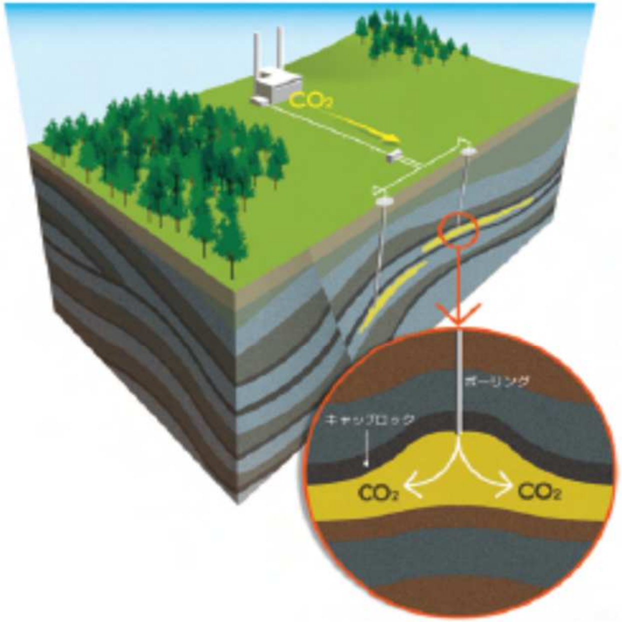


Sparse Linear Solvers in ppOpen-HPC

- **(OpenMP+MPI) Hybrid**
- **Multicoloring/RCM/CM-RCM for OpenMP**
  - **Coloring procedures are NOT parallelized yet**
    - **Distributed parallel version in HPCC 2017 [Kawai et al.]**
- ppOpen-APPL/FEM, FVM, FDM
  - ILU/BILU(p,d,t)+CG/GPBiCG/GMRES, Depth of Overlapping
  - Hierarchical Interface Decomposition (HID) [Henon & Saad 2007], Extended HID [KN 2010]
- **ppOpen-MATH/MG**
  - **Geometric Multigrid Solvers/Preconditioners**
  - **Comm./synch. avoiding/reducing based on *h*CGA**
    - **[KN 2014, Best Paper Award in IEEE/ICPADS 2014]**
- ppOpen-APPL/BEM
  - H-Matrix Solver: HACApK
  - Only Open-Source H-Matrix Solver Library by OpenMP/MPI

Example: Geologic CO_2 Storage

3D Multiphase Flow (Liquid/Gas) + 3D Mass Transfer



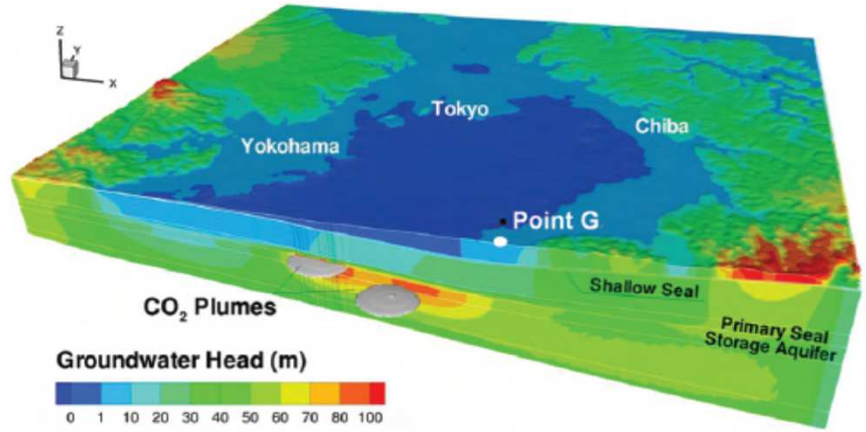
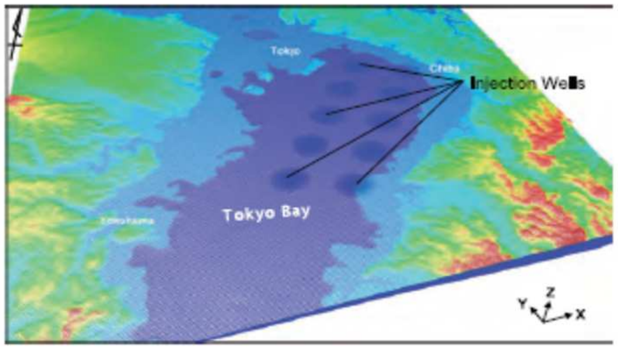
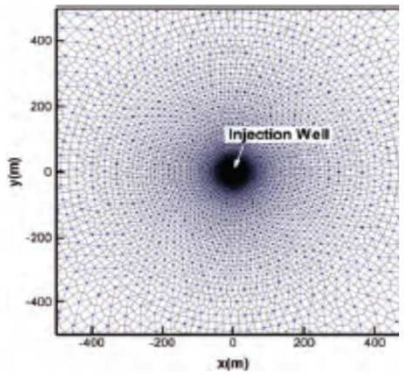
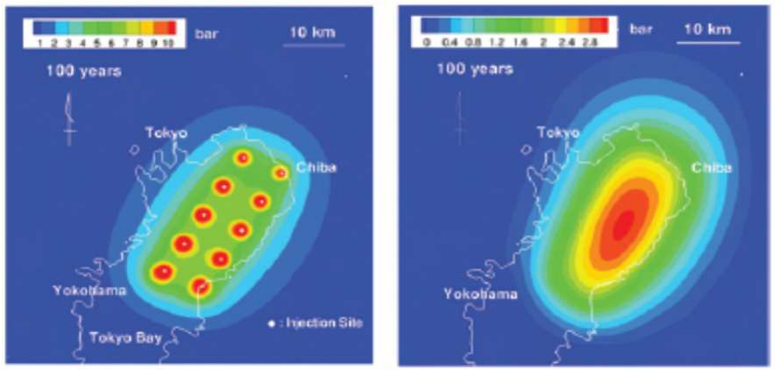


図 -4　CO_2 圧入後の地下水圧（全水頭換算）の分布（100 年後）



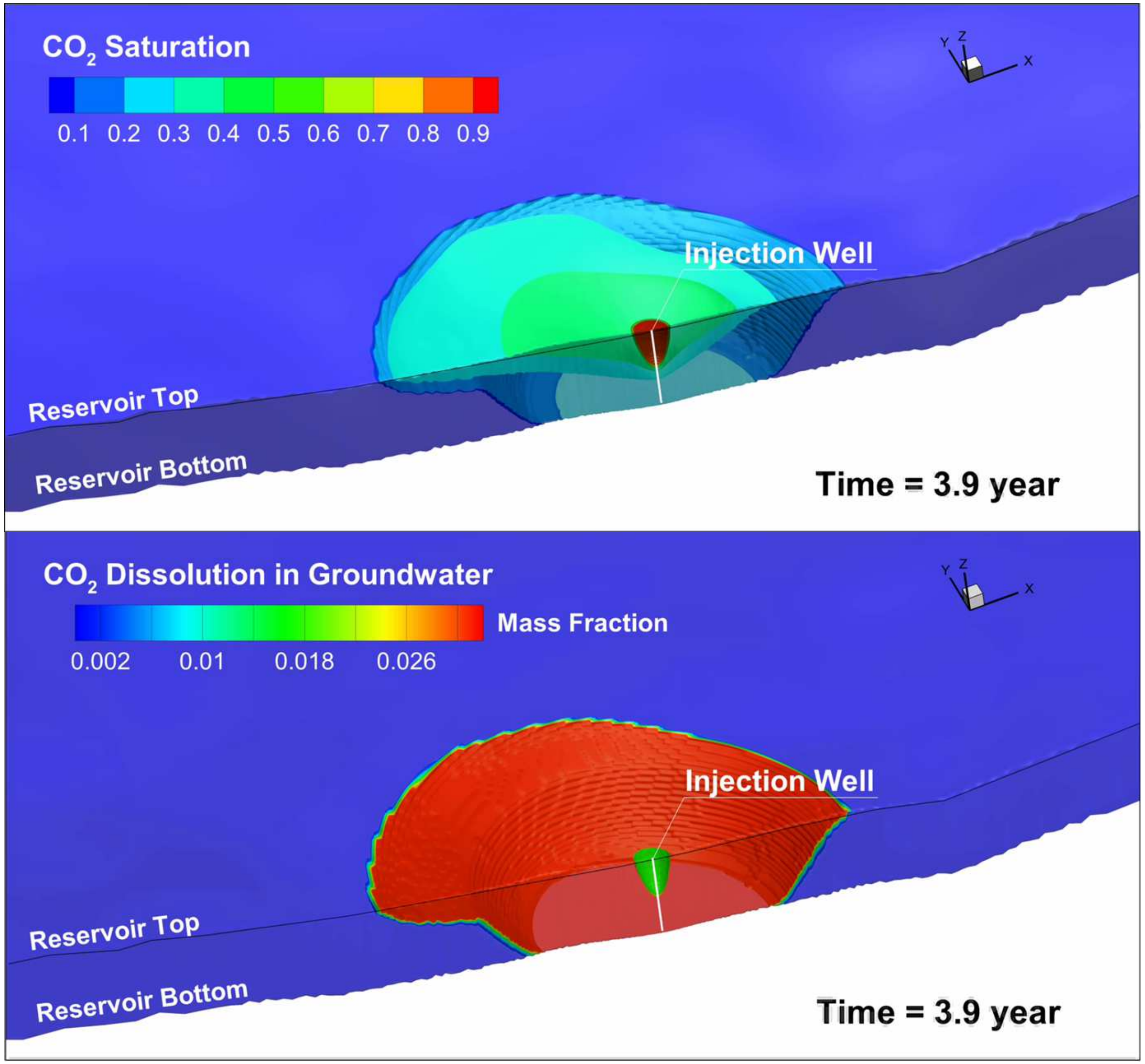




(a) 深部遮蔽層下面　　(b) 浅部遮蔽層下面

図 -5　圧力上昇量の平面分布（初期状態からの増分、圧入開始から 100 年後）

[Dr. Hajime Yamamoto, Taisei]



Density convections for 1,000 years:

Flow Model

Only the far side of the vertical cross section passing through the injection well is depicted.

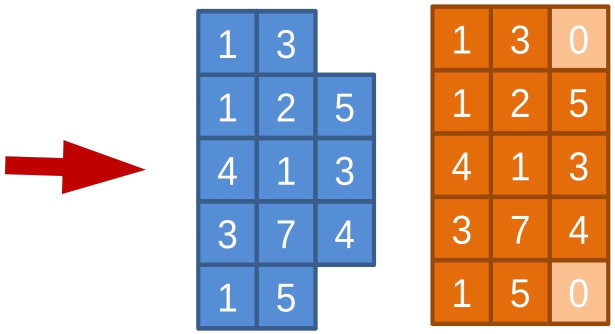
[Dr. Hajime Yamamoto, Taisei]

- The meter-scale fingers gradually developed to larger ones in the field-scale model
- Huge number of time steps (> 10^5) were required to complete the 1,000-yrs simulation
- Onset time (10-20 yrs) is comparable to theoretical (linear stability analysis, 15.5yrs)

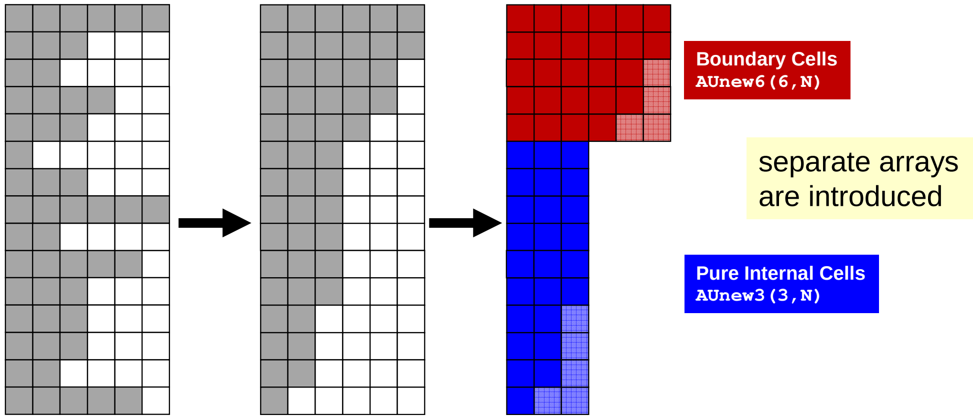
Optimization of Serial Comm.

ELL (Ellpack-Itpack), Sliced-ELL for Matrix Storage

$$\begin{bmatrix} 1 & 3 & 0 & 0 & 0 \\ 1 & 2 & 5 & 0 & 0 \\ 4 & 1 & 3 & 0 & 0 \\ 0 & 3 & 7 & 4 & 0 \\ 1 & 0 & 0 & 0 & 5 \end{bmatrix}$$



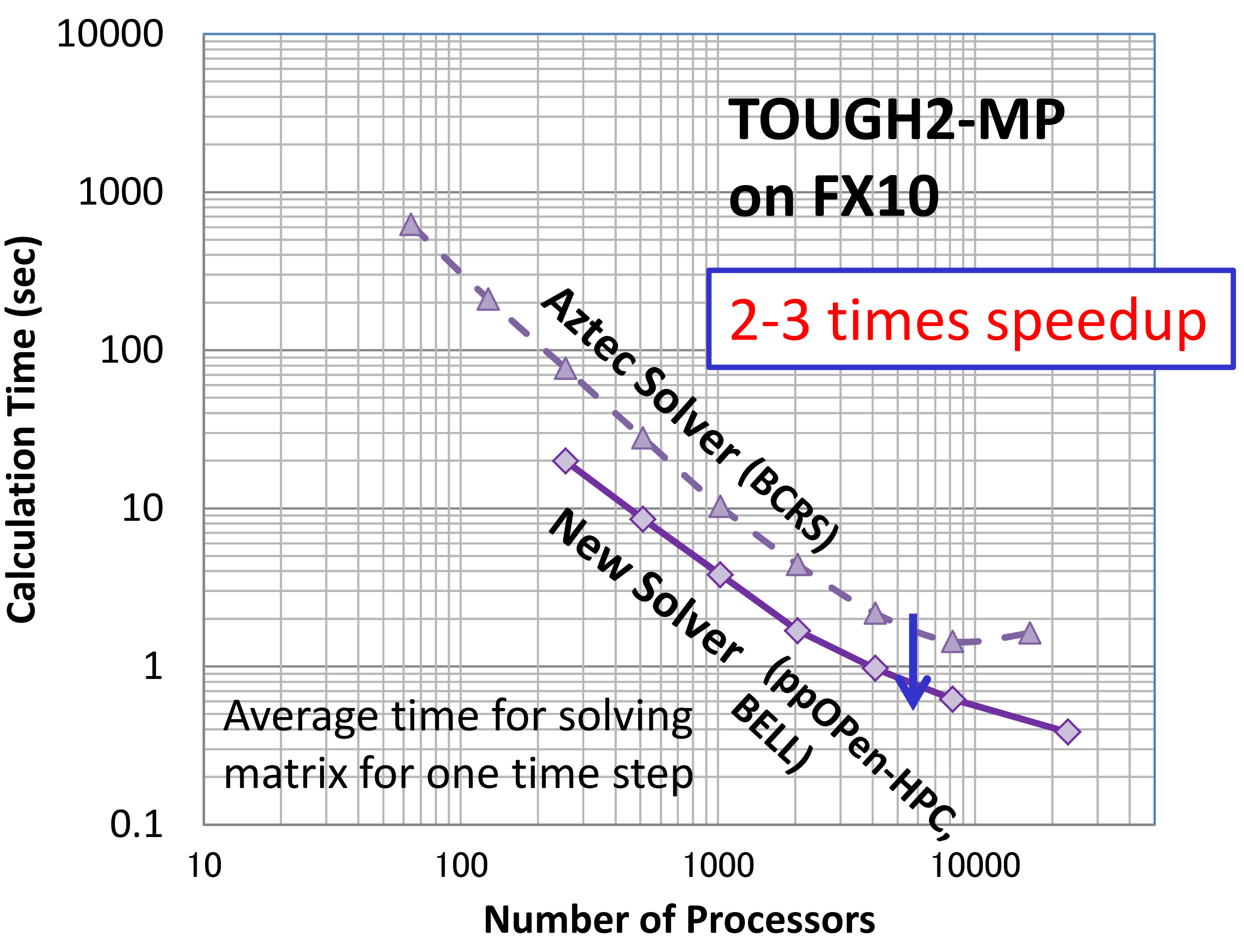
(a) CRS (b) ELL

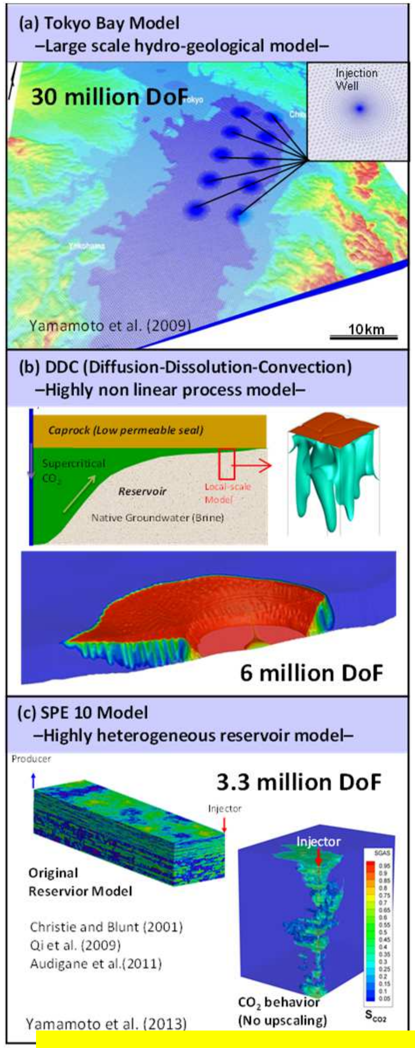


- **A lot of X-ELL-Y-Z’s !**
  - **focusing on SpMV**
- SELL-C-σ
  - M. Kreutzer et al
- Recently, X-ELL-Y-Z’s are applied to forward/backward substitutions with data dependency
  - Gauss-Seidel: Easy
- ILU
  - Much more difficult than GS

Simulation of Geologic CO_2 Storage

30 million DoF (10 million grids × 3 DoF/grid node)





[Dr. Hajime Yamamoto, Taisei]

Fujitsu FX10（Oakleaf-FX）, 30M DOF: 2x-3x improvement

3D Multiphase Flow (Liquid/Gas) + 3D Mass Transfer

- ppOpen-HPC
- **ESSEX-II: Preconditioned Iterative Solvers for Eigenvalue Problems in Quantum Physics**
  - **Dr. Masatoshi Kawai (U.Tokyo)**
  - **Most slides are based on his works.**
- Other Collaborations in ESSEX-II
  - SELL-C-σ
  - CRAFT

ESSEX-II: 2nd Phase of ppOpen-HPC

Japan-German Collaboration (FY.2016-2018)

http://blogs.fau.de/essex/

- Equipping Sparse Solvers for Exascale (FY.2013-15)
- ESSEX + Tsukuba (Prof. Sakurai) + U.Tokyo (ppOpen-HPC)
  - Leading PI: Prof. Gerhard Wellein (U. Erlangen)
- Mission of Our Group in ESSEX-II: Preconditioned Iterative Solver for Quantum Science
  - DLR (German Aerospace Research Center)

国立研究開発法人 科学技術振興機構 Japan Science and Technology Agency

CREST 戦略的創造研究推進事業 Core Research for Evolutionary Science and Technology

DFG Deutsche Forschungsgemeinschaft

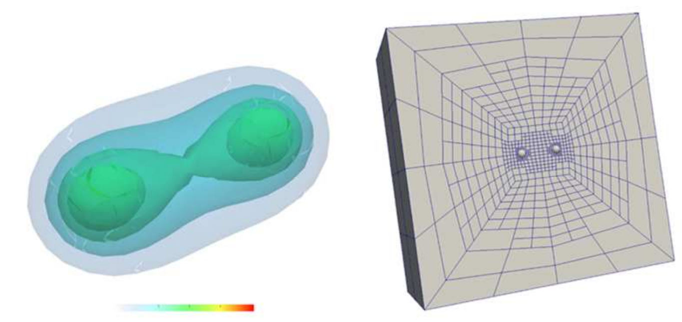
SPPEXA

Preconditioned Iterative Solvers in ESSEX-II

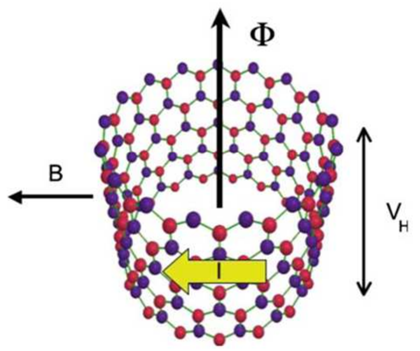
- Robust Preconditioner
  - Blocking + Diagonal Shift
- Parallel Reordering
  - Very Critical Technology for Robust Convergence of ILU/IC-type Preconditioning for Ill-Conditioned Problems on Massively Parallel Systems
    - General remedies are NOT necessarily enough: HID (Hierarchical Interface Decomposition), Extended Overlapping
    - Sequential reordering: inefficient for O(10^3+) processes
  - One of the First Example of Parallel “Global” Reordering on Massively Parallel Systems (IEEE HPCC 2017, accepted)
- Porting developments to GHOST/PHIS

ESSEX-II: General Eigenvalue Problems in Quantum Physics

- Overview [M. Kawai 2017]
  - $Ax=\lambda Bx$
    - $A,B \in \mathbb{C}$^$(n \times n)$, λ: Eigenvalue x: Eigenvectors
  - FEAST or Sakurai-Sugiura Method
    - $A_z\, x=b$, $A_z=(zB-A)$, z:
  - Complex
- Preconditioned Iterative Solver
  - Generally, ill-conditioned (small diagonal components)



Source : D. Davydov, T. D. Young and P. Steinmann *"On the adaptive finite element analysis of the Kohn–Sham equations : methods, algorithms, and implementation"*



Source : A. H. Castro Neto, F. Guinea, N. M. R. Peres, K. S. Novoselov and A. K. Geim, *"The electronic properties of graphene"*

Preconditioner for ill-conditioned matrices

Developing a regularized IC preconditioner

Target matrices

- Provided from generalized eigenvalue problems of quantum systems
  - Positive and negative diagonals
  - Small diagonals entries
  - High condition number

⇒ Challenging problem for iterative methods

Applying two regularizations to the IC preconditioner

■Blocking technique(Regularization①)

Applying the incomplete decomposition to a block matrix

→Better convergence ratio

■Diagonal transformation(Regularization②)

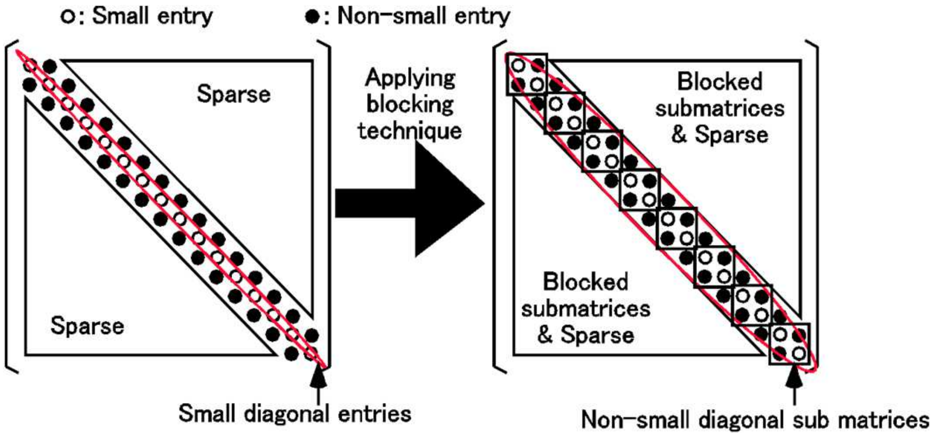
Adding a constant value to the diagonal elements

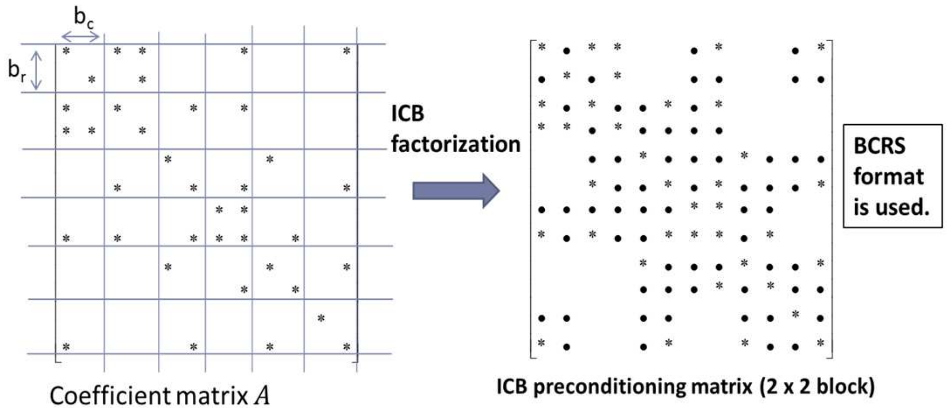
→Making the diagonally dominant matrix, directly

For robustness

Applying 2 regularization methods for ILU preconditioner
→For robustness and improving convergence

- Blocking technique(Regularization①)
  - Applying the incomplete decomposition to a block matrix
    1. More robustness because of including non-small off-diagonals
    2. Better convergence ratio because of allowing more fill-ins
- Diagonal transformation(Regularization②)
  - Adding constant value α to the diagonal elements
  - Directly method to make the diagonally dominant matrix





Target problems

Evaluated the effect of regularized preconditioner for following 15 data-sets

■Graphene

- The simulation of the electrical properties.
- Number of DOF is 128～1,000,000
- 9 data sets
- Number of non-zero elements per row : 13 or 4

■Kohn-Sham

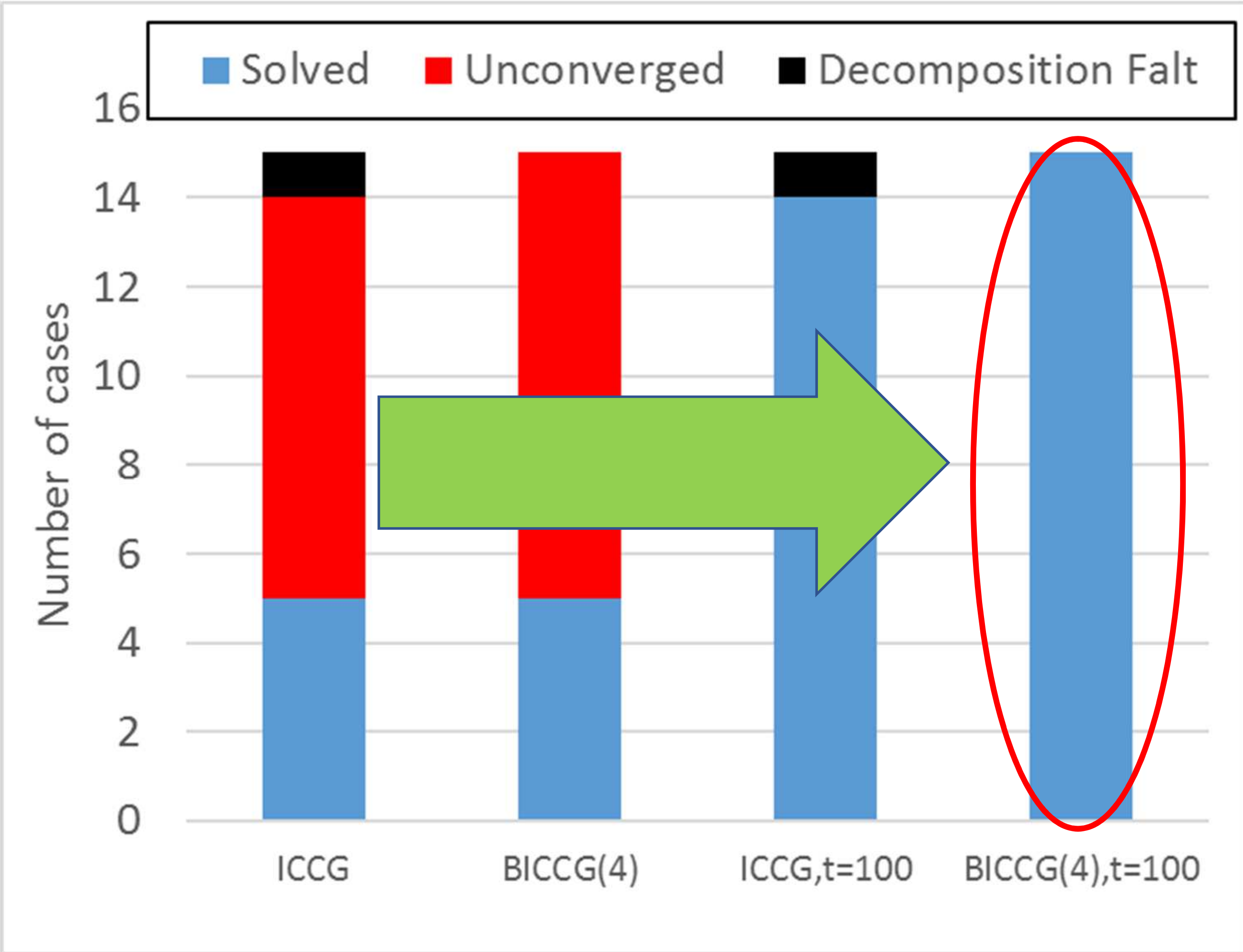
- Simulation of the status of any molecules.
- Number of DOF is 57,575～76,163
- 6 data sets
- Number of non-zero elements per row : 20~24 on average

All problems have a symmetric coefficient matrix

→We used a block IC preconditioned CG method

Result

Solved all problems by applying both regularizations



The size of block for Block IC (Regularization1) is 4
The constant value for the diagonal shifting (Regularization2) is 100.0

Parallelization method for the preconditioner

Proposing a hierarchical parallelization method for multi-coloring

■ Multi-coloring algorithms

- Greedy
- Cuthill-Mckee
- Block multi-coloring
- Cyclic multi-coloring

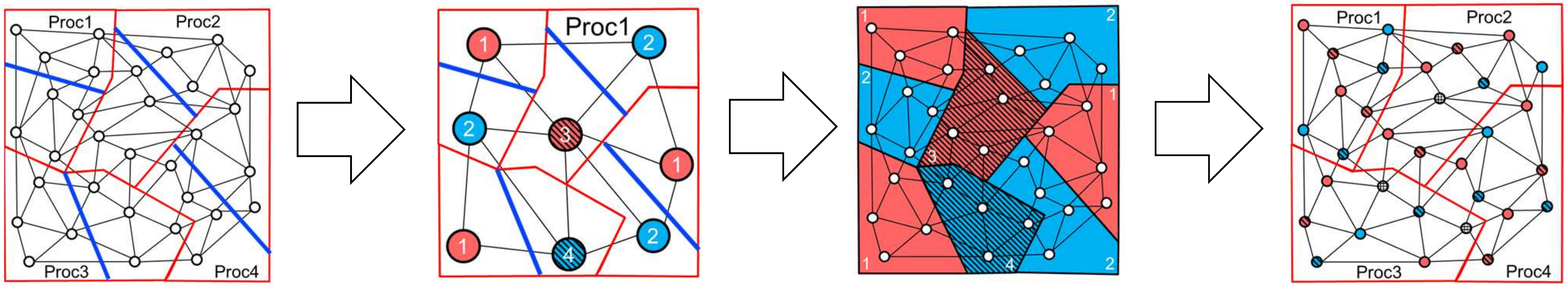
Sequential algorithms

To parallelize the algorithms, proposing a hierarchical method

Hierarchical algorithm

Supplied a colored “area” by hierarchical method

→ Applying any multi-coloring algorithm in parallel



[M. Kawai 2017]

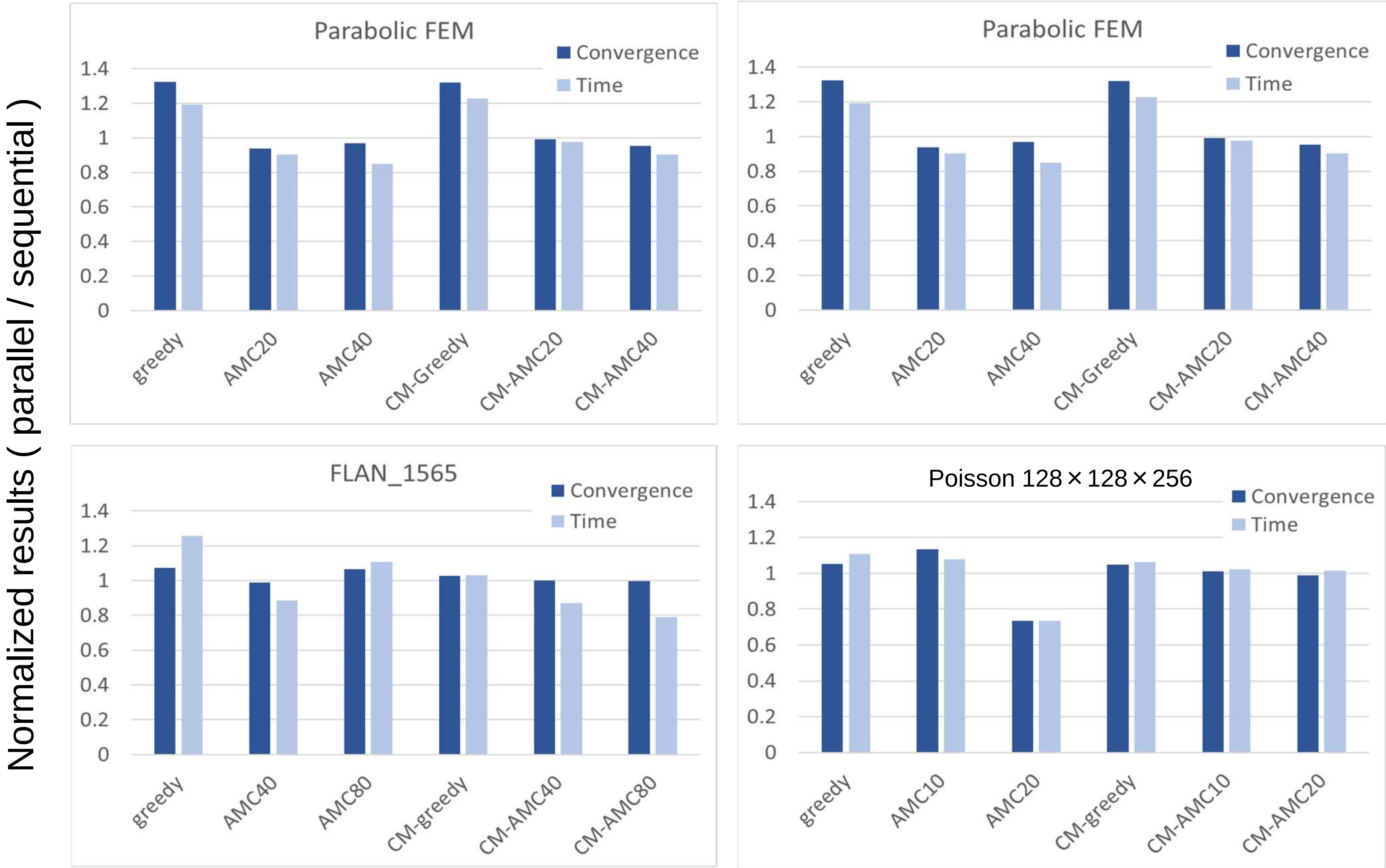
Creating graph & coloring

Scattering colored area

Coloring parallely

Comparing convergence and calculation time

Convergence and calculation time are similar to sequential coloring.

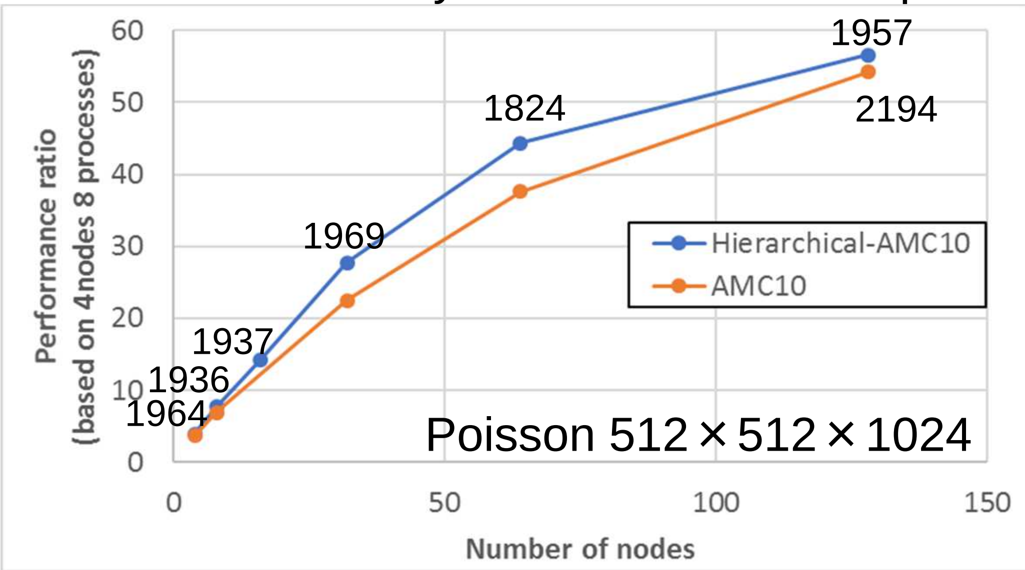


The results with 32 nodes : 2 processes/node : 18 threads/process on Reedbush-U

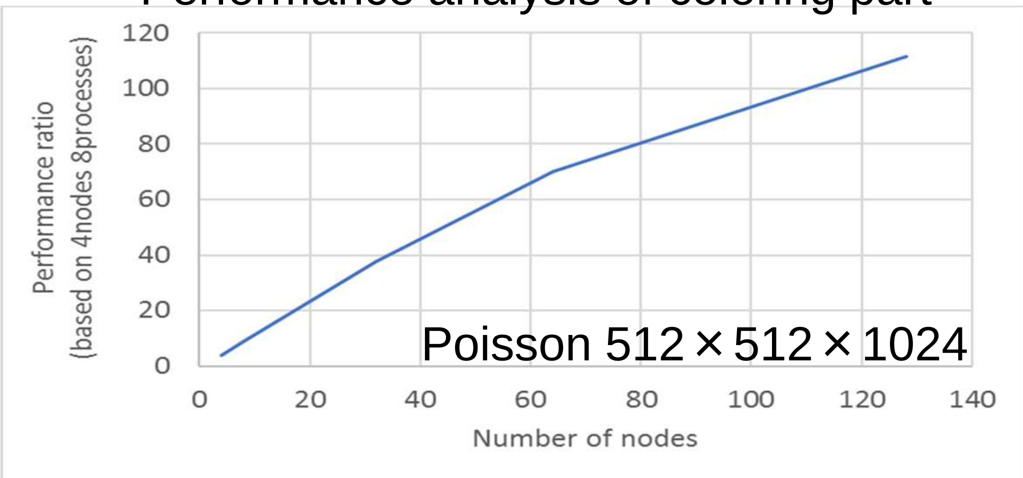
Result of hierarchical coloring on Poisson problem

Showed the similar convergence and calculation time on the larger problem.

Performance analysis of ICCG iteration part



Performance analysis of coloring part



Performance of iteration and coloring part is good.

Computational time of hierarchical coloring is 2.3% of the total.

[M. Kawai 2017]

Performance evaluation with the graphene models

Evaluated performance with the large graphene model

■Oakleaf-FX 4～1024 nodes

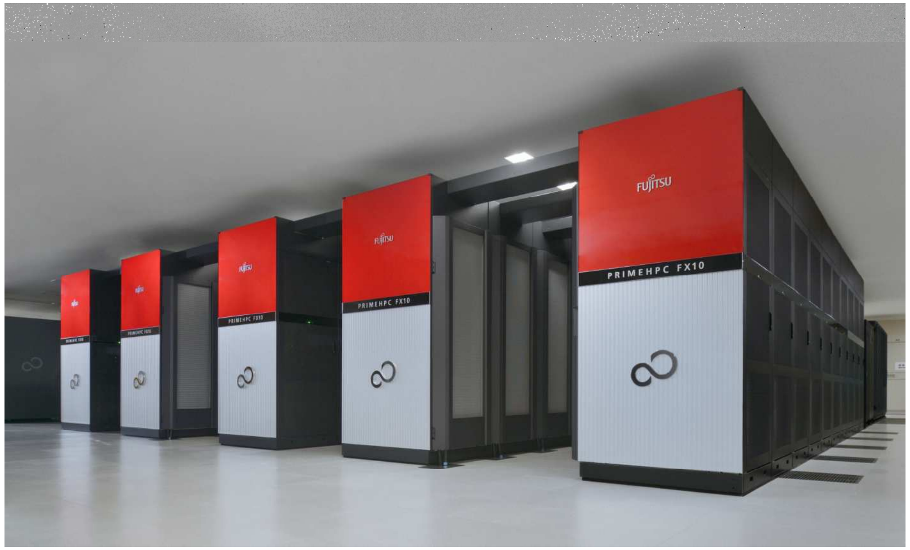
Node specifications

✓SPARC64™ IXfx 16cores

✓32GiB

Network specifications

✓Tofu



■Hybrid parallelization

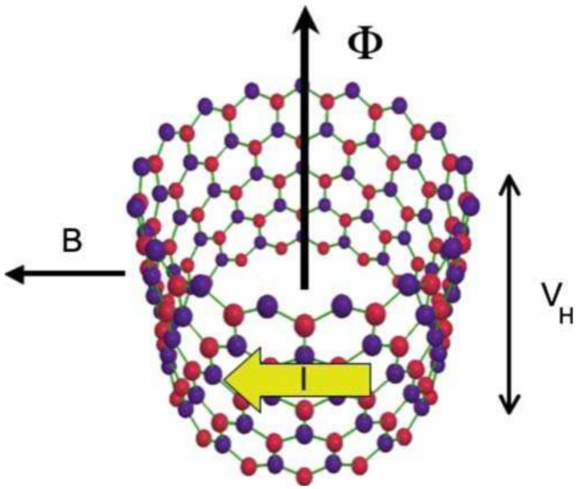
1process-16 threads (1process per node)

■ Coloring algorithm for test

Hierarchical parallelized AMC(10)

■ Target problem

Graphen8194 × 4096 ≒ 33M DoF

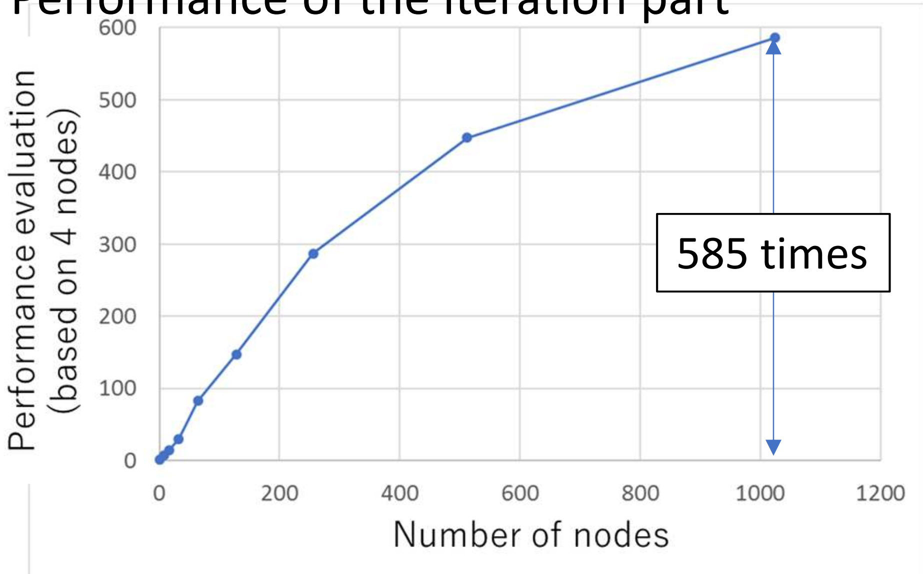


Source : The electronic properties of graphene

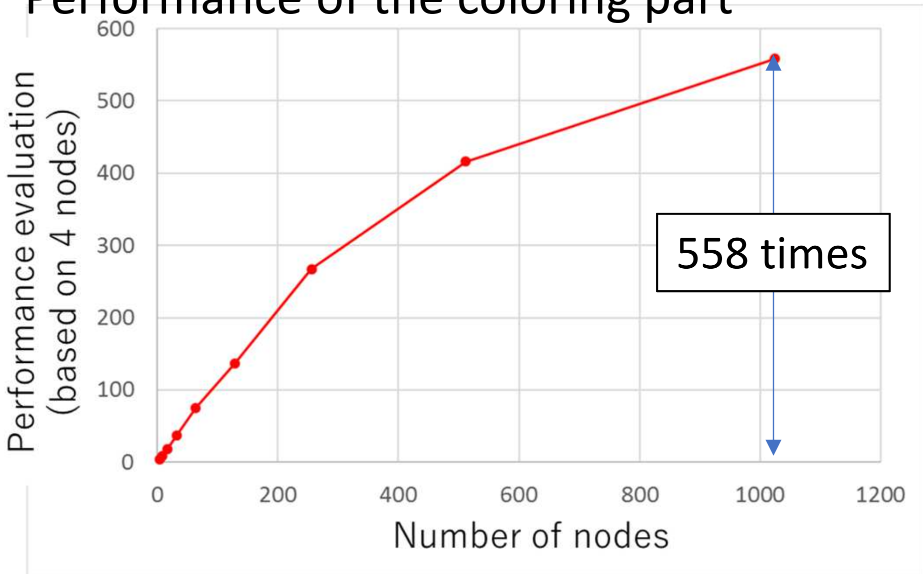
Results on the large graphene model

Showed good performance both iteration and coloring part

Performance of the iteration part



Performance of the coloring part



Maximum difference of the number of iterations among each processes was 0.08%.

Next Challenge:
4,800 nodes of Oakleaf-FX
Oakforest-PACS

GHOST, PHIST

https://blogs.fau.de/essex/

- C++ Libraries developed in ESSEX project
  - Multicore/Manycore/GPU Clusters
- **Libraries developed in ESSEX-II will be based on GHOST/PHIST**
- Fortran interface is also under development

■ GHOST
- General, Hybrid and Optimized Sparse Toolkit
- Dense/Sparse Matrices

■ PHIST (Pipelined Hybrid Parallel Iterative Solver Toolkit)
- Pipelined Hybrid Parallel Iterative Solver Toolkit
- Krylov Iterative Solvers for Sparse Matrices
- GHOST functions are called

[M. Kawai 2017]

How to publish the code

The codes will be published with the *Phist.*

The *Phist* is the framework which is developed by the ESSEX-II project.

→ We are adding our code to the *PHIST*

Pro of the *PHIST* is that we can choose any kernels easily.

→ We can use optimized code to each systems.

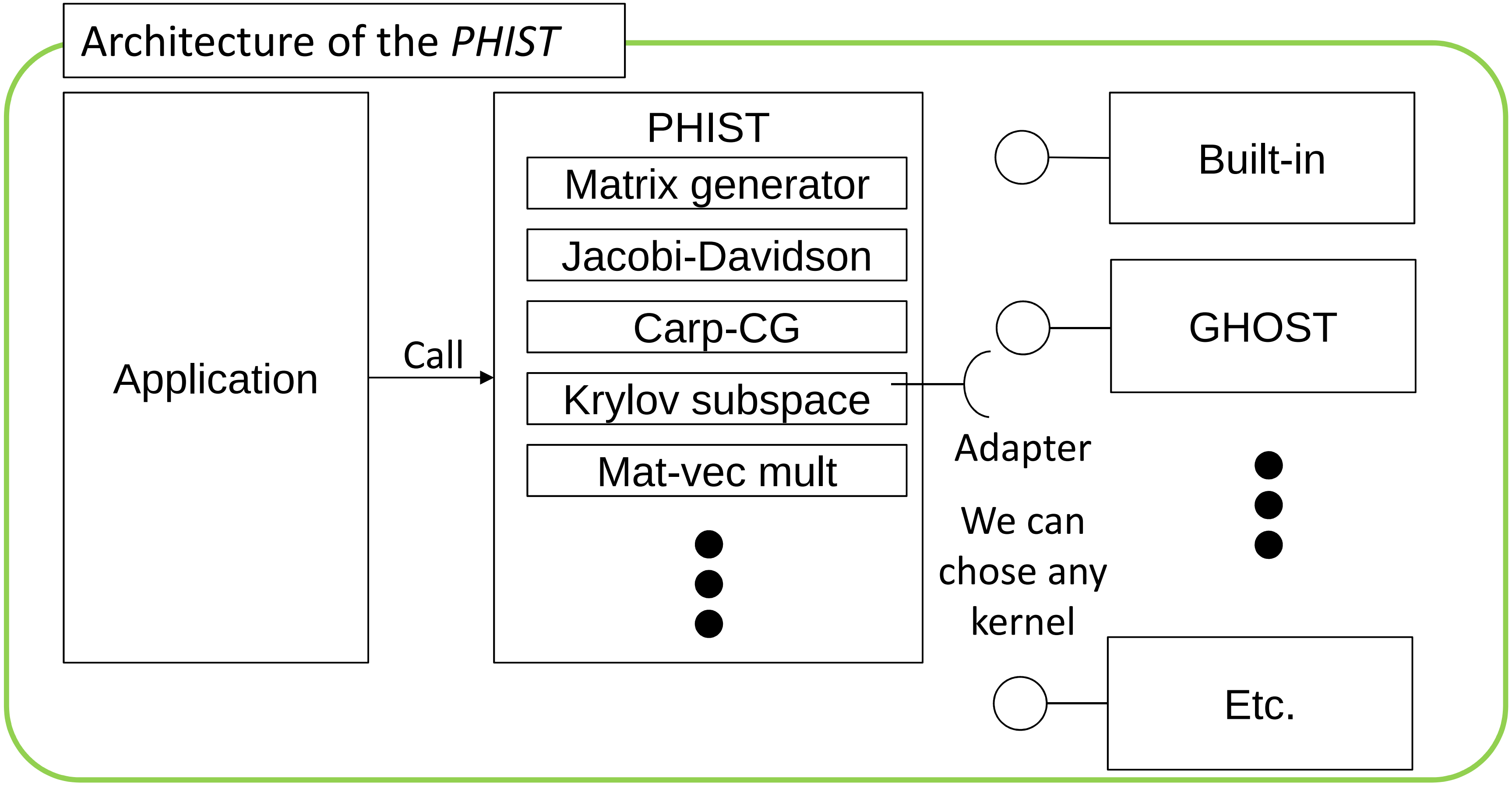
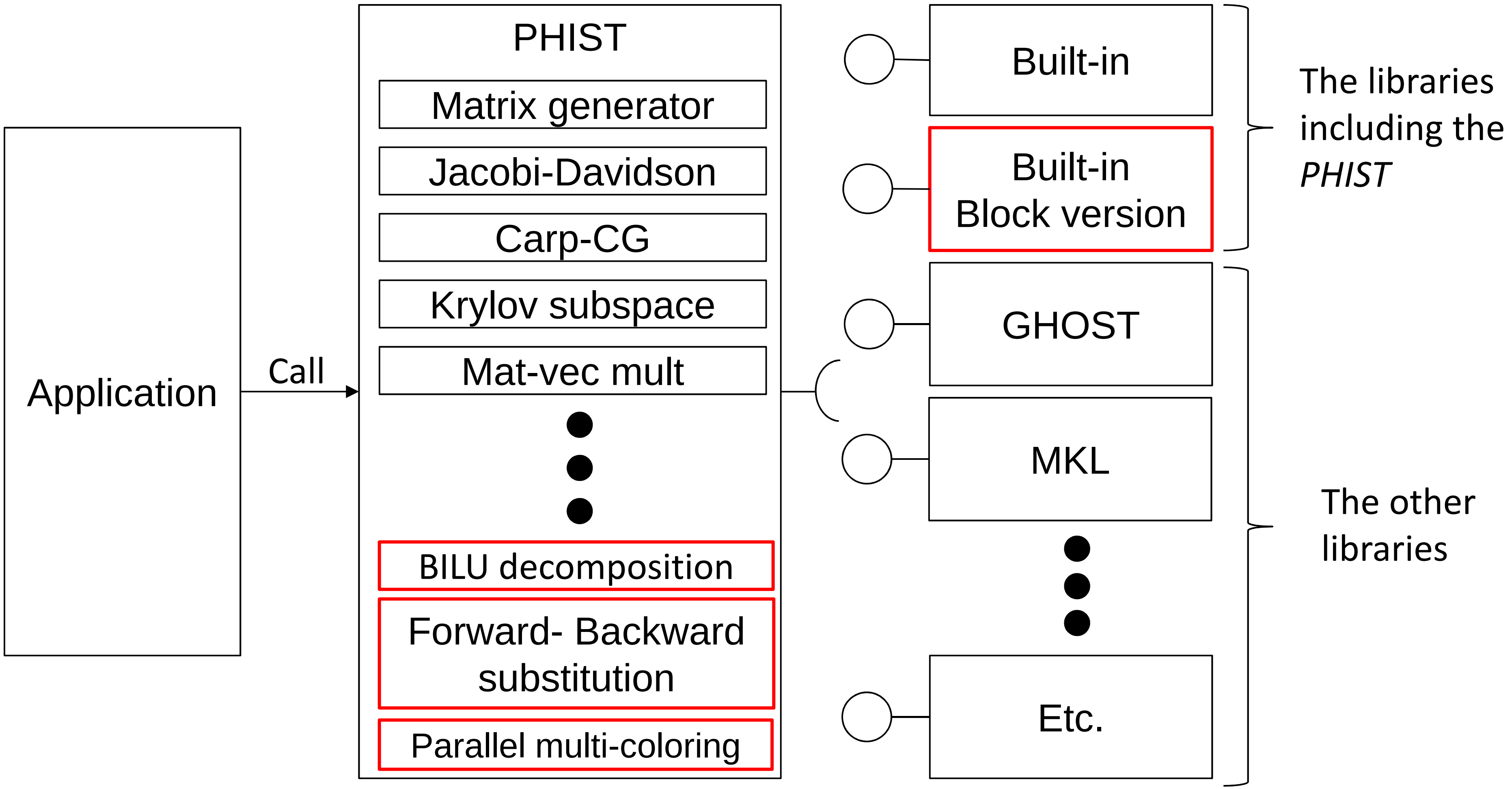


Image of the *PHIST* including our codes

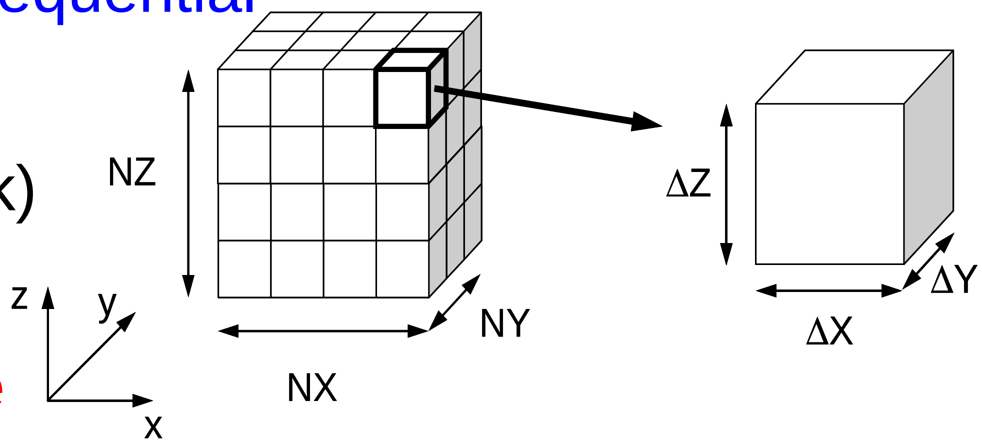
Red parts are our implementation.



- ppOpen-HPC
- ESSEX-II: Preconditioned Iterative Solvers for Eigenvalue Problems in Quantum Physics
- **Other Collaborations in ESSEX-II: Lucky Experiences**
  - **SELL-C-σ**
  - CRAFT

Poisson3D-OMP

- Finite Volume Method, Poisson Equations (128^3 cells)
  - FDM-type mesh (7-pt. Stencil), Unstructured data structure
  - SPD matrix
  - ICCG: Data Dependency for Incomplete Cholesky Fact.
- Fortran 90 + OpenMP
- Thread Parallelization by OpenMP: Reordering needed
  - CM-RCM + Coalesced/Sequential
- Storage of Matrix
  - CRS, ELL (Ellpack-Itpack)
- Outer Loops
  - Row-Wise, Column-Wise

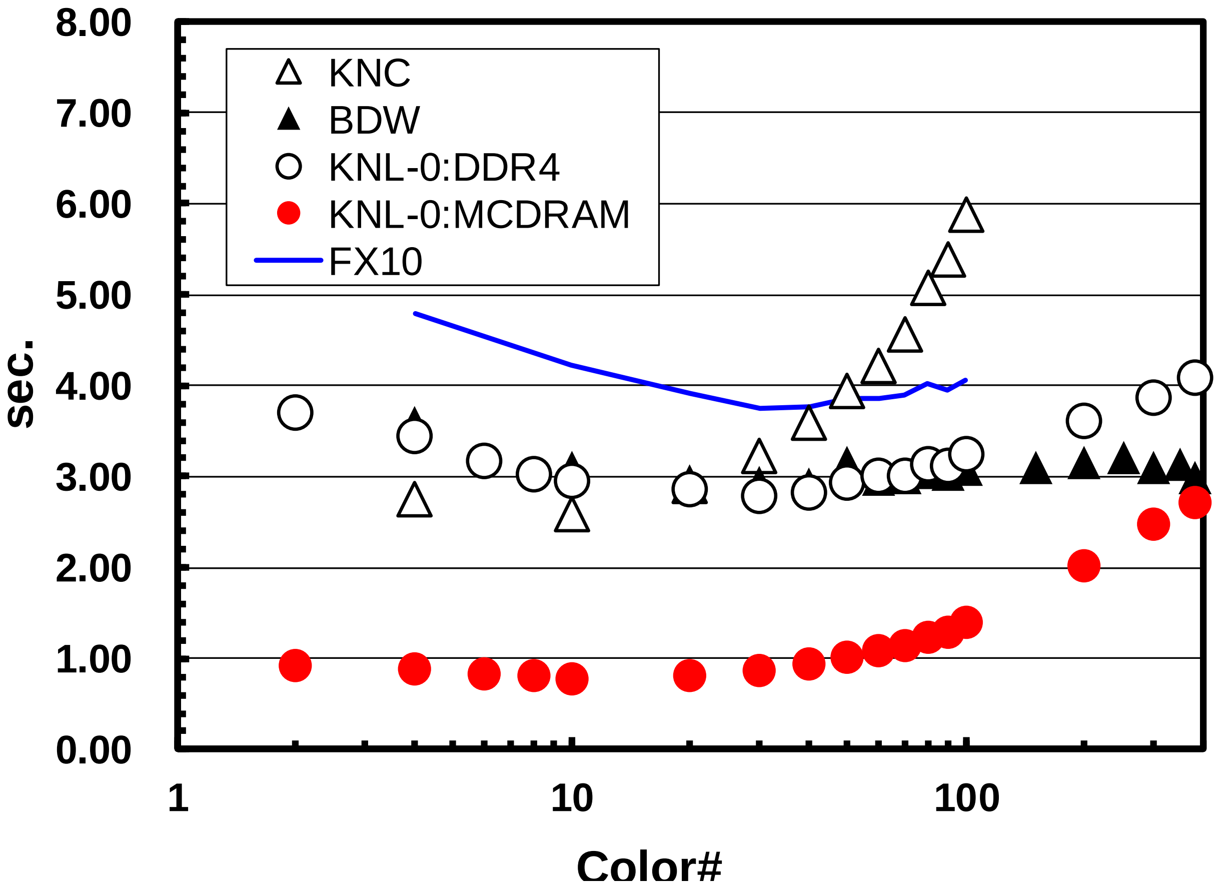


Comp. Time for ICCG (Best ELL)

Effects of synchronization overhead are significant on KNL & KNC, if number of colors is larger

Generally, optimum number of color is 10 for KNL/KNC

Down is Good !!

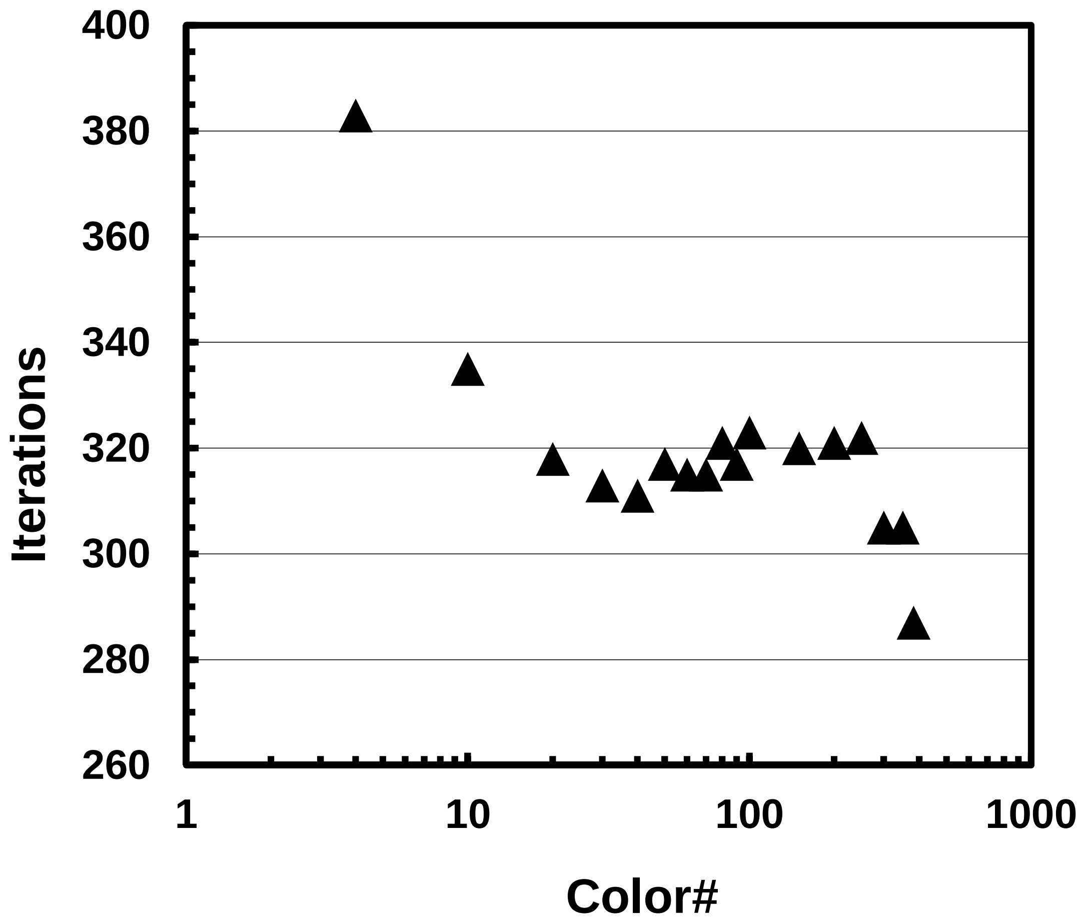


- FX10: 60+GB/sec memory throughput at 20 colors (equivalent to STREAM)
- KNL-0: MCDRAM: 300 GB/sec (70% of STREAM)

Code Name	KNC	KNL-0	BDW	FX10
Architecture	Intel Xeon Phi 5110P (Knights Corner)	Intel Xeon Phi 7210 (Knights Landing)	Intel Xeon E5-2695 v4 (Broadwell-EP)	SPARC IX fx
Frequency (GHz)	1.053	1.30	2.10	1.848
Core # (Max Thread #)	60 (240)	64 (256)	18 (18)	16 (16)
Peak Performance (GFLOPS)	1,010.9	2,662.4	604.8	236.5
Memory (GB)	8	MCDRAM: 16 DDR4: 96	128	32
Memory Bandwidth(GB/sec., Stream Triad)	159	MCDRAM: 454 DDR4: 72.5	65.5	64.7
Out-of-Order	N	Y	Y	N
System		OFP-mini	Reedbush-U	Oakleaf-FX

Number of Iter's until Convergence

Better convergence for larger number of colors
Synchronization overhead is significant for larger number of colors

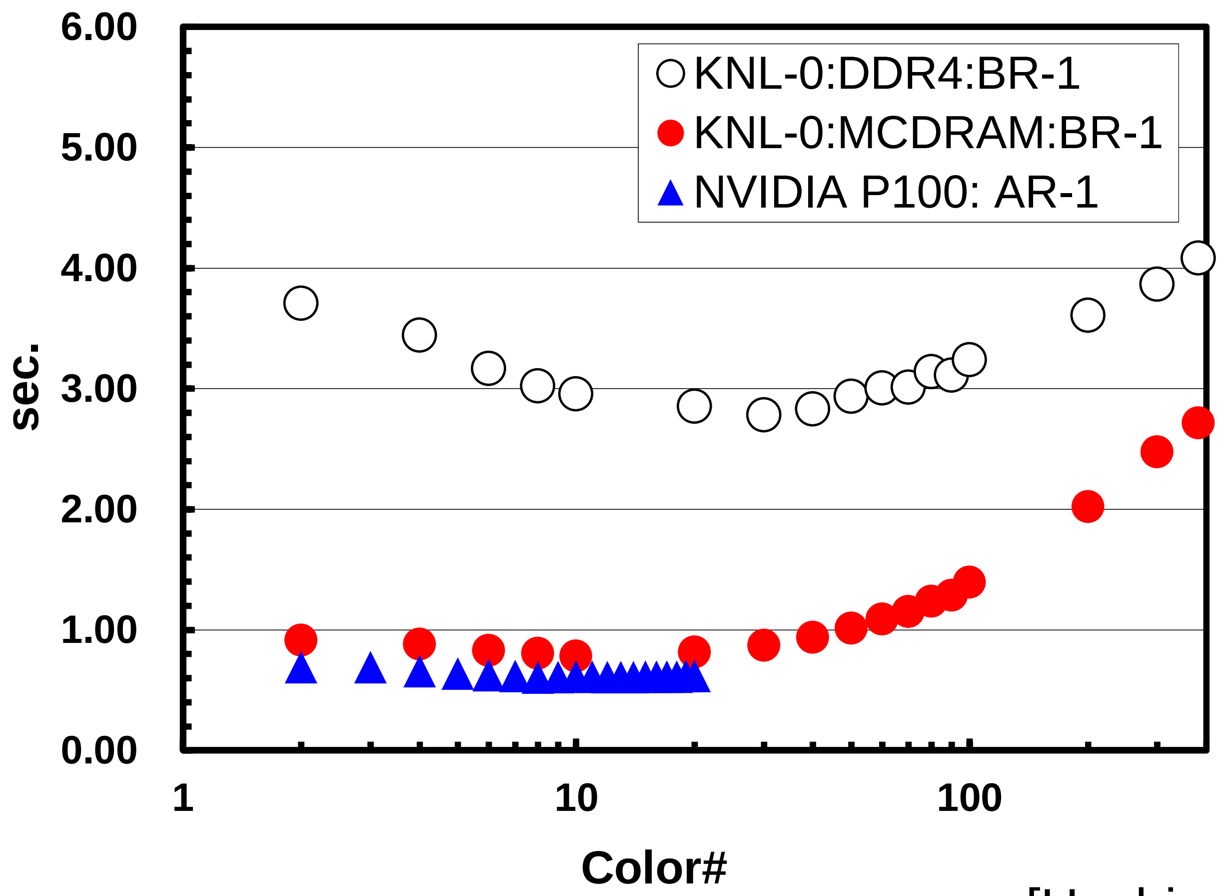


Code Name	KNC	KNL-0	P100
Architecture	Intel Xeon Phi 5110P (Knights Corner)	Intel Xeon Phi 7210 (Knights Landing)	NVIDIA Tesla P100 (Pascal)
Frequency (GHz)	1.053	1.30	1.328
Core # (Max Thread #)	60 (240)	64 (256)	1,792
Peak Performance (GFLOPS)	1,010.9	2,662.4	4,759
Memory (GB)	8	MCDRAM: 16 DDR4: 96	16
Memory Bandwidth(GB /sec., Stream Triad)	159	MCDRAM: 454 DDR4: 72.5	530

ICCG for FVM (CM-RCM reordering)

Xeon Phi (KNL) and P100 are competitive

Down is Good !!



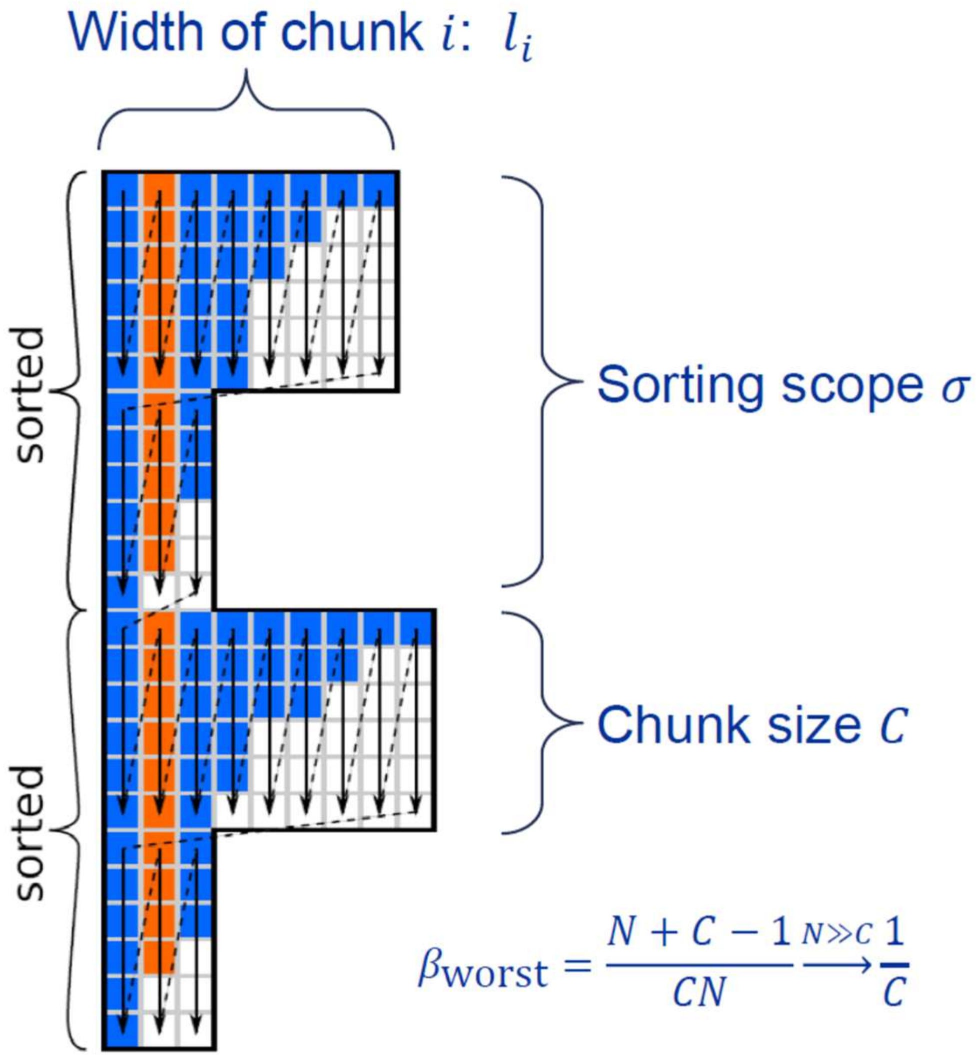
[Hoshino et al. 2017]

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

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



$$\beta_{\text{worst}} = \frac{N + C - 1}{CN} \xrightarrow{N \gg C} \frac{1}{C}$$

SELL-6-12
β=0.66

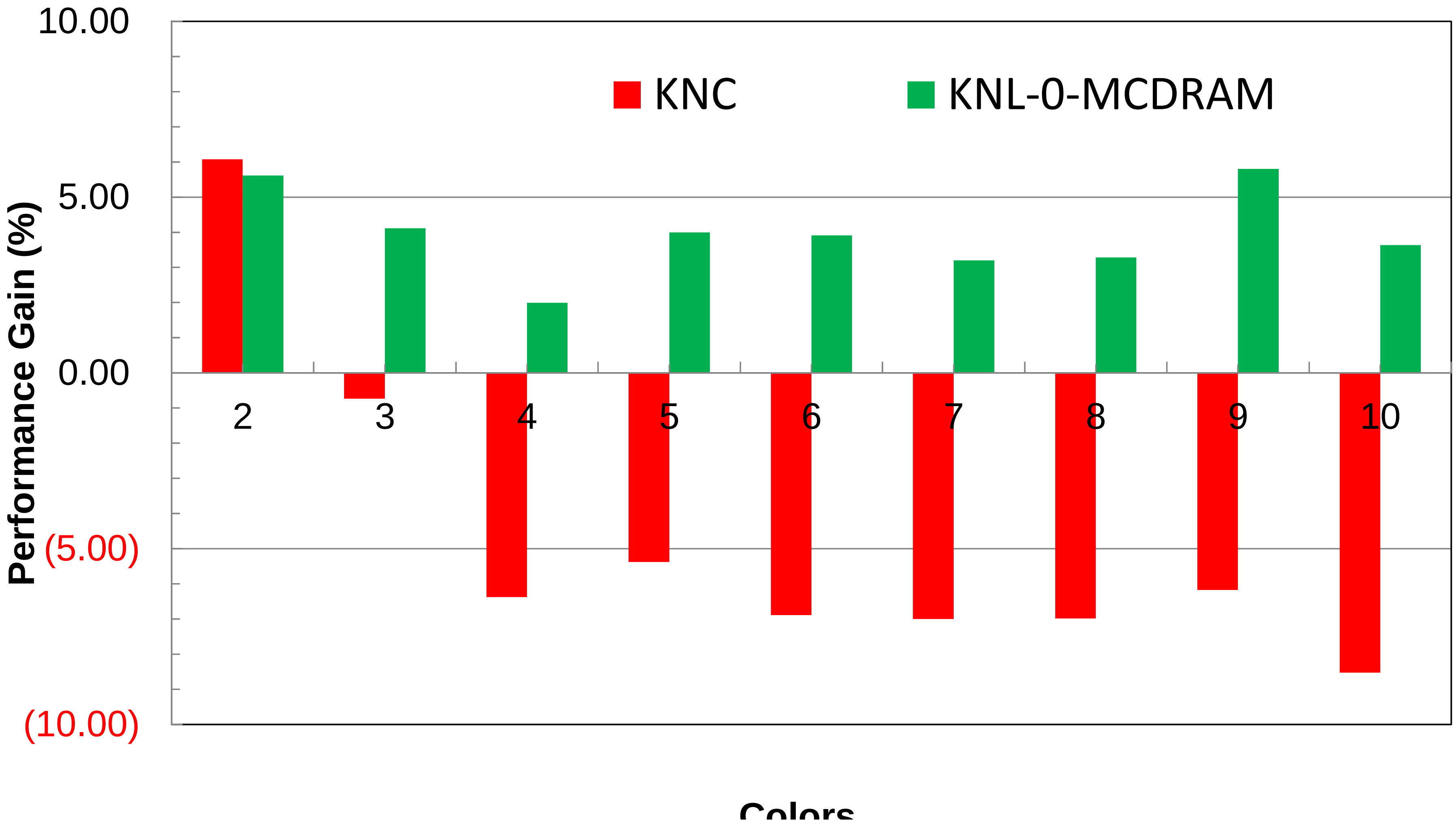
[Kreutzer, Hager, Wellein 2014]

Effects of SELL-C-σ on KNL-0-MC/D

First Example of SELL-C-σ in F/B Substitution

Improvement over Original Best ELL Cases (2-10 colors)

SELL-C-σ is rather slower on KNC



Further Reduction of OpenMP Overhead on KNL

[Hoshino et al. 2017]

- NO Wait for SpMV
- Call $OMP PARALLEL only once at Each Iteration
- Remove $OMP DO
- Remove REDUCTION: Dot products are calculated at each thread
- Performance Improvement on SELL-8-1
  - KNL: 16.7 %, KNC: 25.0 %
- Finally, KNL-MCDRAM is 3.6x faster than KNL-DDR4
  - Room for further performance engineering
  - 1% of peak (e.g. HPCG (27-pt stencil): 1.5% of peak)
  - Throughput: 70+% of STREAM Triad

- ppOpen-HPC
- ESSEX-II: Preconditioned Iterative Solvers for Eigenvalue Problems in Quantum Physics
- **Other Collaborations in ESSEX-II: Lucky Experiences**
  - SELL-C-σ
  - **CRAFT**

Data Migration for Fault-Resilient Scientific Computations

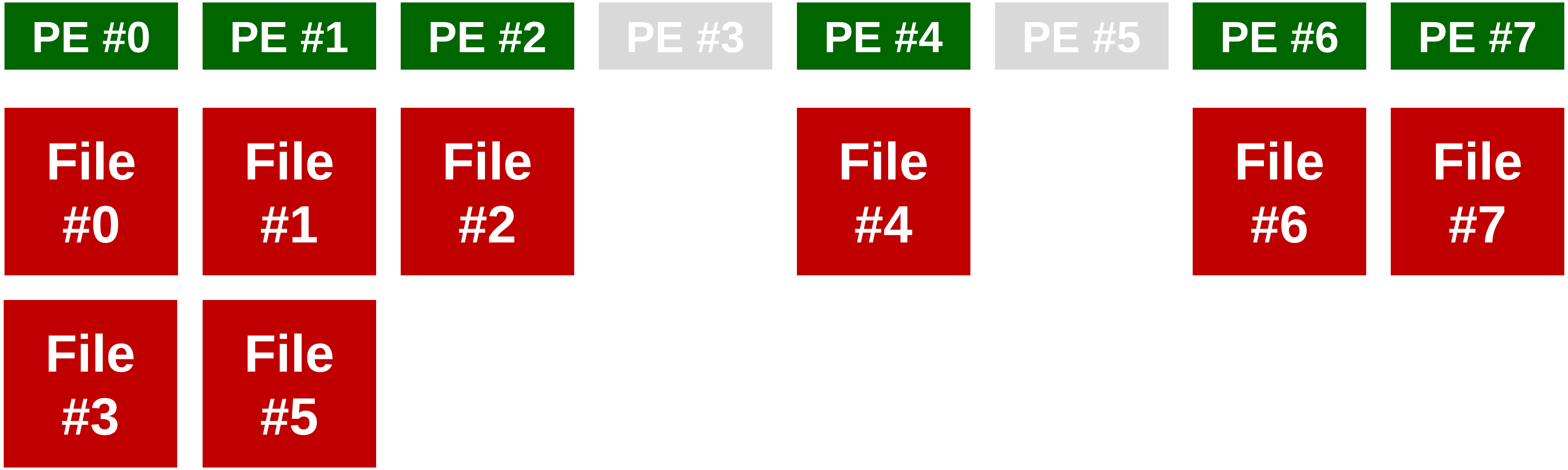
- Target Applications
  - FEM, FVM, FDM etc. with Domain Decomposition
- **CRAFT Library [Shazad, Wellein]**
  - **A library for easier application-level Checkpoint/Restart and Automatic Fault Tolerance**
  - **Based on ULFM MPI (User Level Failure Mitigation)**
- Distributed Check-Point Files
  - Geometric Mesh Files, Result Vectors
  - Could be by MPI-IO
- Starting with M processes, m processes failed
  - shrink to (M-m) processes, without “spare” nodes

Data Migration for Fault-Resilient Scientific Computations Starting with 8 Processes

PE #0	PE #1	PE #2	PE #3	PE #4	PE #5	PE #6	PE #7
File #0	File #1	File #2	File #3	File #4	File #5	File #6	File #7

Data Migration for Fault-Resilient Scientific Computations
#3 and #5 failed

Data Migration for Fault-Resilient Scientific Computations
Files are read by #0 and #1



Data Migration for Fault-Resilient Scientific Computations Repartitioning & Data Migration

PE #0	PE #1	PE #2	PE #3	PE #4	PE #5
File #0	File #1	File #2	File #3	File #4	File #5

Repartitioning/data-migration is supported by AMR and Load Balancing Tool of ppOpen-HPC and ParMETIS

Data Migration for Fault-Resilient Scientific Computations

- This topic was not considered at all in the beginning of ESSEX-II at January 2016
  - While discussing with Faisal Shazad (Ph.D student of Gerhard, developer of CRAFT), I tried to introduce CRAFT library to my FEM code with AMR/Load Balancing.
- My MS student is working for this now.
  - He is staying at FAU, Erlangen this week, and working with Faisal.
  - I visited FAU yesterday (Tuesday), and will be back on Friday.
  - Current application is 3D Steady-State Heat Transfer.
    - Next Target is Time-Dependent Application
- Joint Poster Presentation: SIAM PP18

Summary/Future Works

- ppOpen-HPC
- ESSEX-II
  - pK-Open-SOL
  - Preconditioning
  - Parallel Reordering
- Other Collaborations in ESSEX-II
  - SELL-C-σ
  - CRAFT
- Further Optimization of SELL-C-σ on OFP (Original Target of ppOpen-HPC)

SIAM Conference on
Parallel Processing for Scientific Computing (PP18)
March 7-10, 2018
Waseda University, Tokyo, Japan

Record Breaking 126 MS Proposals, 100+ CP
700+ Participants ? <u>Please book your Hotel ASAP !</u>

- Organizing Committee Co-Chairs's
  - Satoshi Matsuoka (Tokyo Institute of Technology, Japan)
  - Kengo Nakajima (The University of Tokyo, Japan)
  - Olaf Schenk (Universita della Svizzera italiana, Switzerland)

Thanks for Your Contributions !!
Many MS's related to SPPEXA

.jp

  - http://siampp18.jsiam.org/