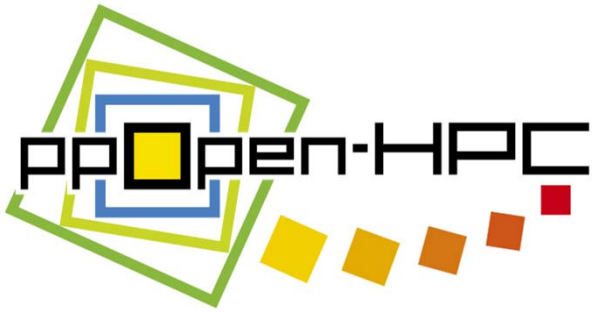
本講習会の概要(1/2)

- ppOpen-HPCで学ぶ並列プログラミングと並列前処理付き反復法
  - 共催:東京大学情報基盤センター・PCクラスターコンソーシアム
  - http://nkl.cc.u-tokyo.ac.jp/seminars/ppOpen-APPL-FVM/
- ppOpen-HPC(http://ppopenhpc.cc.u-tokyo.ac.jp/)
  - JST戦略的創造研究推進事業CREST研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」の一環として実施されている「自動チューニング機構を有するアプリケーション開発・実行環境(研究代表:中島研吾(東京大学情報基盤センター))」において開発
  - Fortranベース，C言語へのインタフェースは現在開発中

本講習会の概要(2/2)

- 本講習会
  - ppOpen-HPC の概要
  - ppOpen-HPC の機能の一つである ppOpen-APPL/FVM の使用法と内部データ構造
  - ppOpen-APPL/FVM を使用して開発された並列有限要素法プログラムの詳細
  - 大規模科学技術計算に必須である「並列前処理付き反復法
  - FX10 スーパーコンピュータシステム(Oakleaf-FX)(Fujitsu PRIMEHPC FX10)による実習
- 有限要素法等による科学技術アプリケーションの並列化をお考えの方、大規模並列シミュレーション手法全般について勉強したい方には是非受講をお勧めいたします。



自動チューニング機構を有するアプリケーション開発・実行環境 ppOpen-HPC

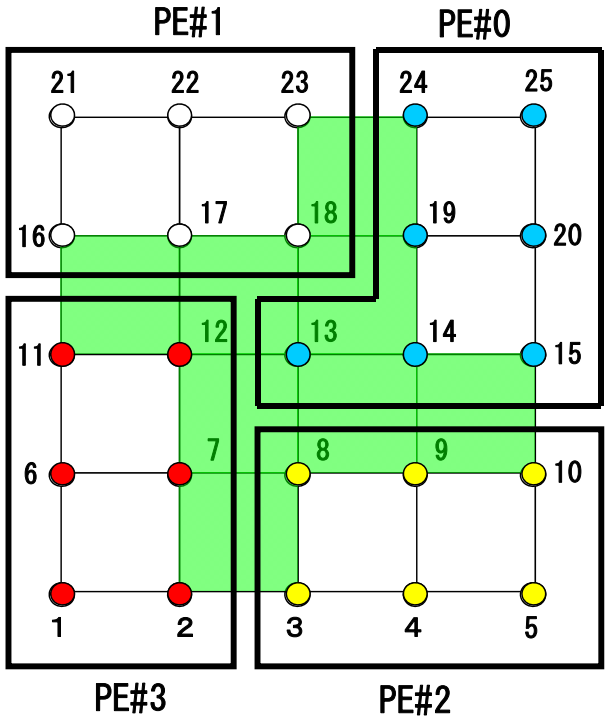
中島 研吾

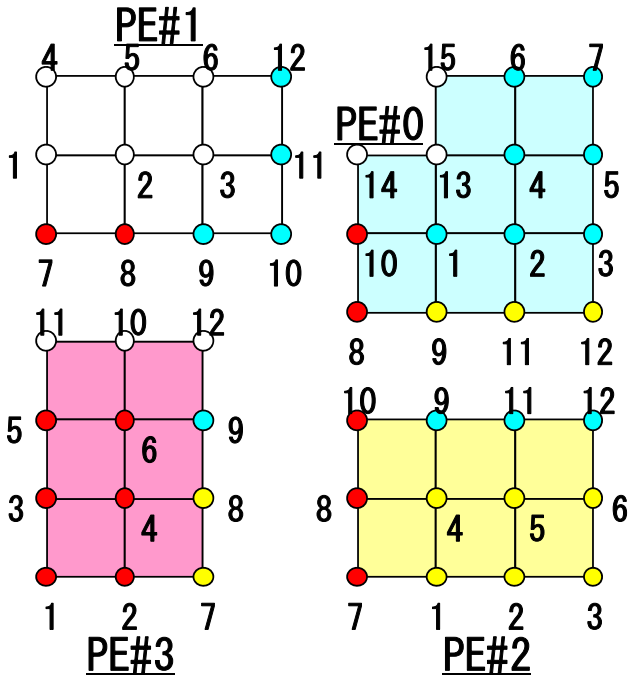
東京大学情報基盤センター

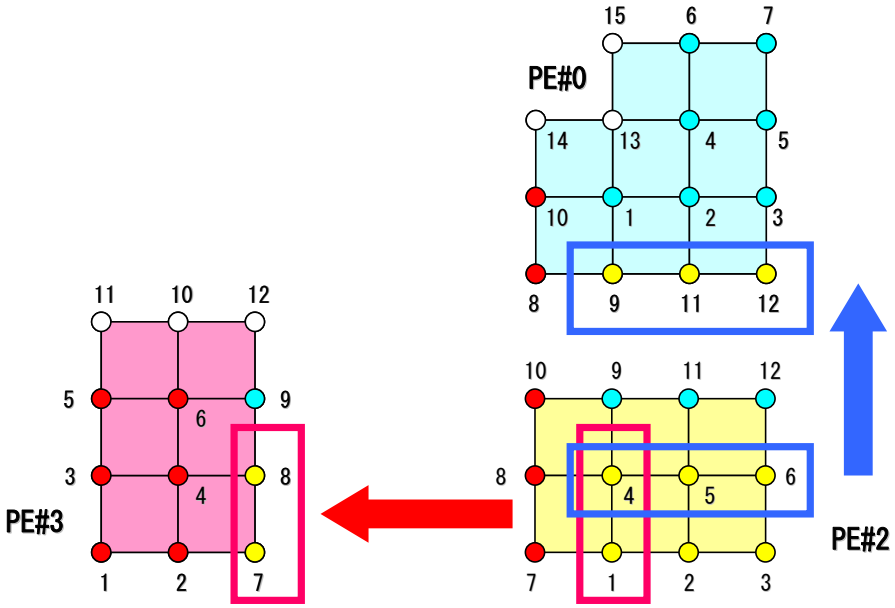
佐藤正樹（東大・大気海洋研究所），奥田洋司（東大・新領域創成科学研究科），
古村孝志（東大・情報学環/地震研），岩下武史（北大・情報基盤センター），
阪口秀（海洋研究開発機構）

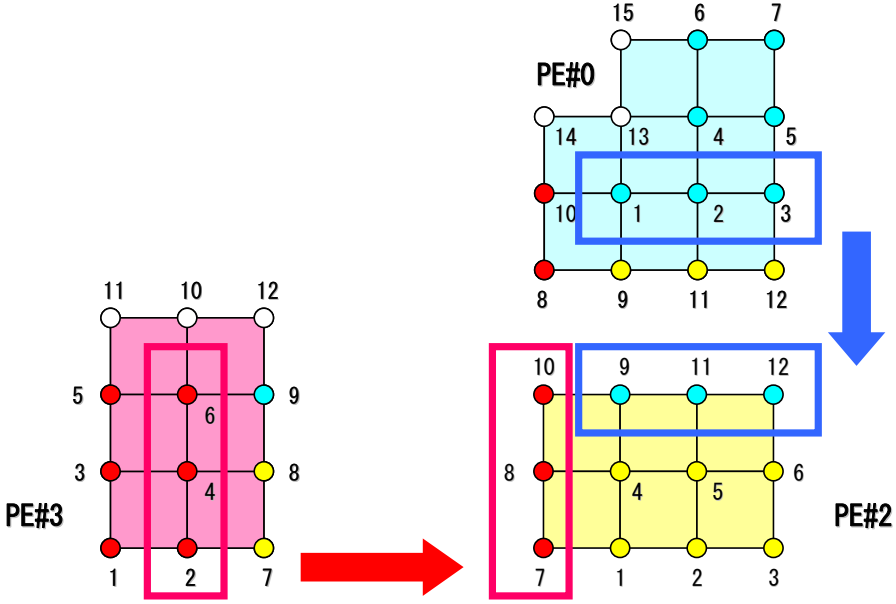
背　景(1/2)

- 大規模化，複雑化，多様化するハイエンド計算機環境の能力を充分に引き出し，効率的なアプリケーションプログラムを開発することは困難
- 有限要素法等の科学技術計算手法：
  - プリ・ポスト処理，行列生成，線形方程式求解等の一連の共通プロセスから構成される。
  - 共通プロセスを抽出し，ハードウェアに応じた最適化を施したライブラリとして整備することで，アプリケーション開発者から共通プロセスに関わるプログラミング作業，並列化も含むチューニング作業を隠蔽可能。
    - 例：並列有限要素法における通信：局所分散データ構造
  - アプリケーションMW，HPC-MW，フレームワーク


PE#1
PE#0
21 22 23 24 25
16 17 18 19 20
11 12 13 14 15
6 7 8 9 10
1 2 3 4 5
PE#3
PE#2

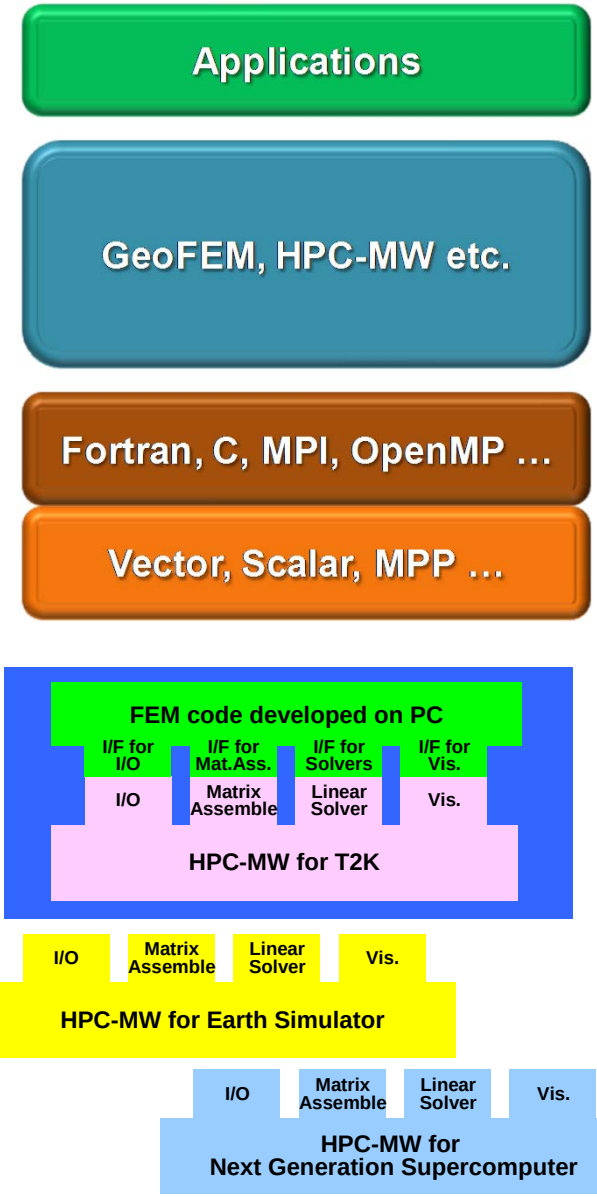

PE#1
4 5 6 12
1 2 3 11
7 8 9 10
PE#0
15 6 7
14 13 4 5
10 1 2 3
8 9 11 12
11 10 12
5 6 9
3 4 8
1 2 7
PE#3
10 9 11 12
8 4 5 6
7 1 2 3
PE#2


PE#0
15 6 7
14 13 4 5
10 1 2 3
8 9 11 12
11 10 12
5 6 9
3 4 8
1 2 7
PE#3
10 9 11 12
8 4 5 6
7 1 2 3
PE#2


PE#0
15 6 7
14 13 4 5
10 1 2 3
8 9 11 12
11 10 12
5 6 9
3 4 8
1 2 7
PE#3
10 9 11 12
8 4 5 6
7 1 2 3
PE#2

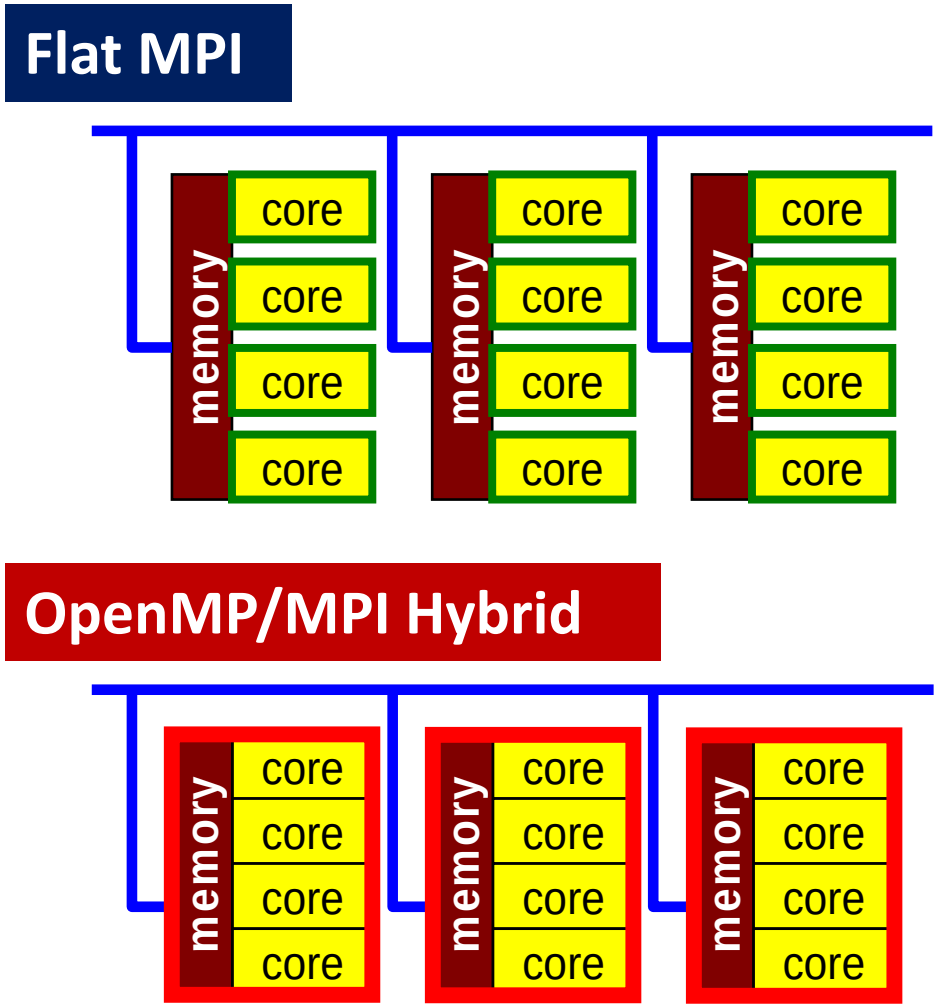
背　景(2/2)

- A.D.2000年前後
  - GeoFEM，HPC-MW
  - 地球シミュレータ，Flat MPI，FEM
- 現在：より多様，複雑な環境
  - マルチコア，GPU
  - ハイブリッド並列
    - MPIまでは何とかたどり着いたが・・・
    - 「京」でも重要
  - CUDA，OpenCL，OpenACC
  - ポストペタスケールからエクサスケールへ
    - より一層の複雑化



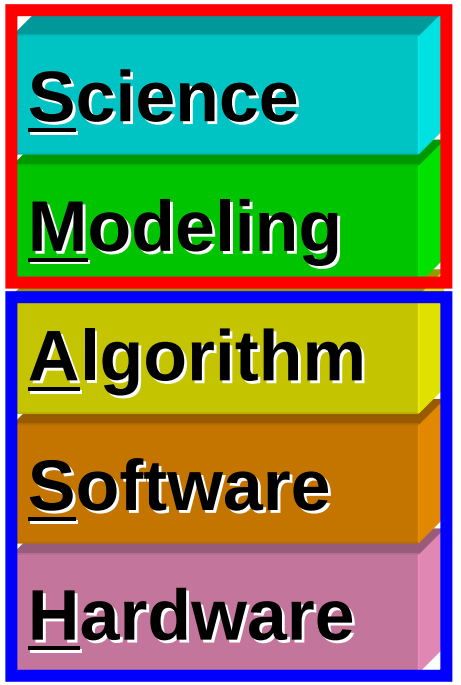
Hybrid並列プログラミングモデル MPI+"X"

- Message Passing
  - MPI
- Multi Threading
  - OpenMP
  - CUDA, OpenCL



HPCミドルウェア：何がうれしいか

- アプリケーション開発者のチューニング（並列，単体）からの解放
  - **SM**ASHの探求に専念
    - 一生SMASHと付き合うのはきつい
  - SM**ASH**をカバー
- コーディングの量が減る
- 教育にも適している
- 問題点
  - ハードウェア，環境が変わるたびに最適化が必要となる



Key-Issues for Appl's/Algorithms towards Post-Peta & Exa Computing

Jack Dongarra (ORNL/U. Tennessee) at ISC 2013

- Heterogeneous/Hybrid Architecture
- Communication/Synchronization Reducing Algorithms
- Mixed Precision Computation
- Auto-Tuning/Self-Adapting
- Fault Resilient Algorithms
- Reproducibility of Results

東大情報基盤センターのスパコン

1システム～6年，3年周期でリプレース

Oakleaf-FX (Fujitsu PRIMEHPC FX10)		T2K-Todai （Hitachi HA8000-tc/RS425 ）		Yayoi (Hitachi SR16000/M1)	
Total Peak performance	: 1.13 PFLOPS	Total Peak performance	: 140 TFLOPS	Total Peak performance	: 54.9 TFLOPS
Total number of nodes	: 4800	Total number of nodes	: 952	Total number of nodes	: 56
Total memory	: 150 TB	Total memory	: 32000 GB	Total memory	: 11200 GB
Peak performance / node	: 236.5 GFLOPS	Peak performance / node	: 147.2 GFLOPS	Peak performance / node	: 980.48 GFLOPS
Main memory per node	: 32 GB	Main memory per node	: 32 GB, 128 GB	Main memory per node	: 200 GB
Disk capacity	: 1.1 PB + 2.1 PB	Disk capacity	: 1 PB	Disk capacity	: 556 TB
SPARC64 Ixfx 1.84GHz		AMD Quad Core Opteron 2.3GHz		IBM POWER 7 3.83GHz	

(retired, March 2014)

“Oakbridge-fx” with 576 nodes installed in April 2014 (separated) (136TF)

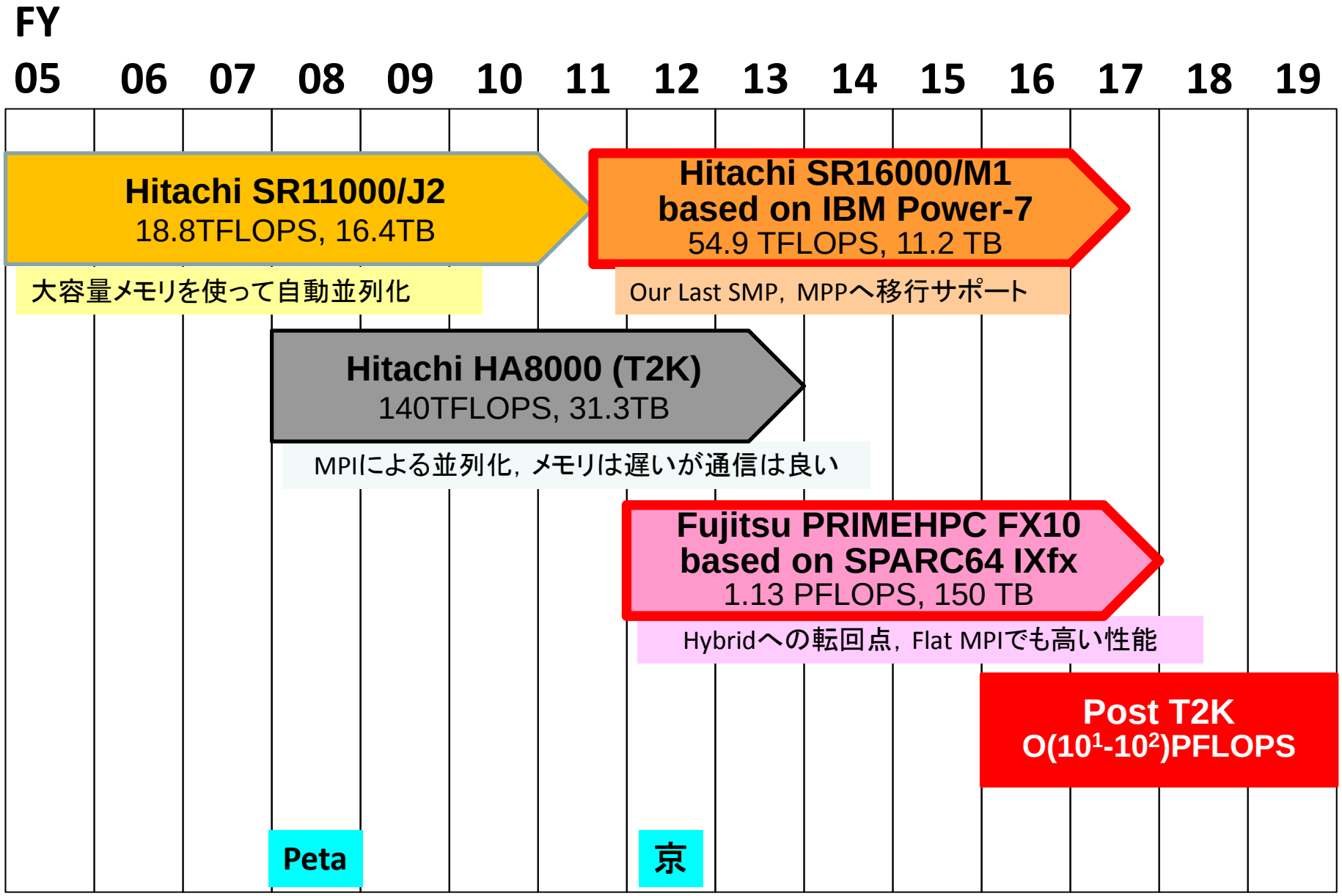
Total Users > 2,000

	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	National Supercomputing Center in Tianjin, China	**Tianhe-2** Intel Xeon E5-2692, TH Express-2, IXeon Phi2013 NUDT	3120000	33863 (= 33.9 PF)	54902	17808
2	Oak Ridge National Laboratory, USA	**Titan** Cray XK7/NVIDIA K20x, 2012 Cray	560640	17590	27113	8209
3	Lawrence Livermore National Laboratory, USA	**Sequoia** BlueGene/Q, 2011 IBM	1572864	17173	20133	7890
4	**RIKEN AICS, Japan**	**K computer, SPARC64 VIIIfx , 2011 Fujitsu**	**705024**	**10510**	**11280**	**12660**
5	Argonne National Laboratory, USA	**Mira** BlueGene/Q, 2012 IBM	786432	8587	10066	3945
6	Swiss Natl. Supercomputer Center, Switzerland	**Piz Daint** Cray XC30/NVIDIA K20x, 2013, Cray	115984	6271	7789	2325
7	TACC, USA	**Stampede** Xeon E5-2680/Xeon Phi, 2012 Dell	462462	5168	8520	4510
8	Forschungszentrum Juelich (FZJ), Germany	**JuQUEEN** BlueGene/Q, 2012 IBM	458752	5009	5872	2301
9	DOE/NNSA/LLNL, USA	**Vulcan** BlueGene/Q, 2012 IBM	393216	4293	5033	1972
10	Government, USA	Cray CS-Storm/Xeon E5-2670/2680/NVIDIA K40, 2014 Cray	72800	3577	6132	1499
48	**ITC/U. Tokyo Japan**	**Oakleaf-FX SPARC64 IXfx, 2012 Fujitsu**	**76800**	**1043**	**1135**	**1177**

R_{max}：実効性能（TFLOPS）
R_{peak}：ピーク性能（TFLOPS），Power: kW

http://www.top500.org/

東大情報基盤センターのスパコン



Post T2K System

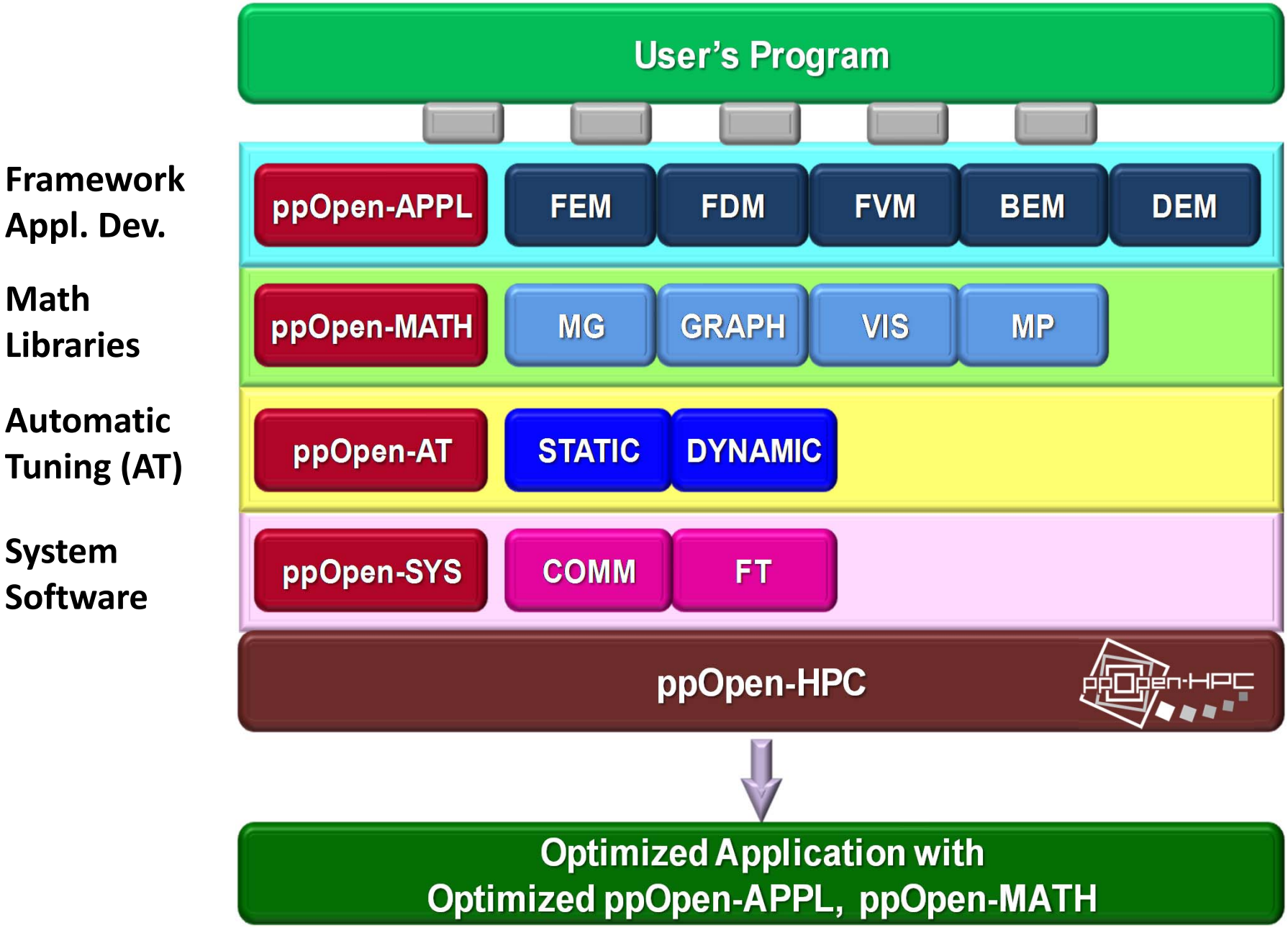
- 20-30 PFLOPS, FY.2016
- Many-core based (e.g. (only) Intel MIC/Xeon Phi)
- Joint Center for Advanced High Performance Computing (JCAHPC, http://jcahpc.jp/)
  - University of Tsukuba
  - University of Tokyo
- Programming is still difficult, although Intel compiler works.
  - (MPI + OpenMP)
  - Tuning for performance (e.g. prefetching) is essential
  - Some framework for helping users needed

ppOpen-HPC

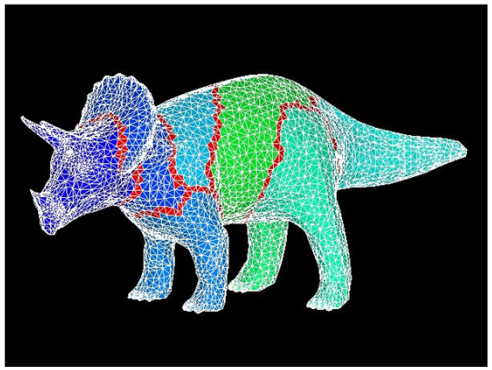
- 東京大学情報基盤センターでは，メニィコアに基づく計算ノードを有するポストペタスケールシステムの処理能力を充分に引き出す科学技術アプリケーションの効率的な開発，安定な実行に資する「自動チューニング機構を有するアプリケーション開発・実行環境：ppOpen-HPC」を開発している。
  - 科学技術振興機構戦略的創造研究推進事業（CREST）研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出（Post-Peta CREST）」（2011～2015年度）
  - PI：中島研吾（東京大学情報基盤センター）
  - 東大（情報基盤センター，大気海洋研究所，地震研究所，大学院新領域創成科学研究科），京都大学術情報メディアセンター，北海道大学情報基盤センター，海洋研究開発機構
  - 様々な分野の専門家によるCo-Design

概 要(1/3)

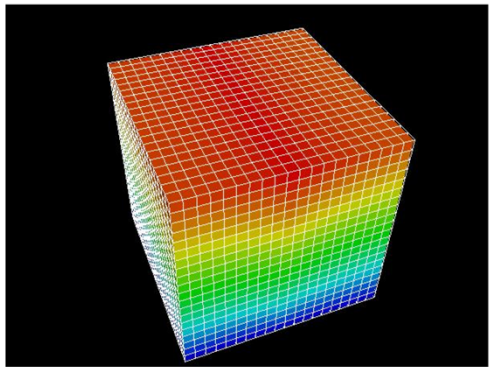
- メニーコアクラスタによるポストペタスケールシステム上での科学技術アプリケーションの効率的開発，安定な実行に資するppOpen-HPCの研究開発を計算科学，計算機科学，数理科学各分野の緊密な協力のもとに実施している。
  - 6 Issues in Post-Peta/Exascale Computingを考慮
  - “pp”: Post Peta
- 東大情報基盤センターに平成27年度導入予定のO(10)PFLOPS級システム(ポストT2K，Intel MIC/Xeon-Phiベース)をターゲット:
  - スパコンユーザーの円滑な移行支援
- 大規模シミュレーションに適した5種の離散化手法に限定し，各手法の特性に基づいたアプリケーション開発用ライブラリ群，耐故障機能を含む実行環境を実現する。
  - ppOpen-APPL:各手法に対応した並列プログラム開発のためのライブラリ群
  - ppOpen-MATH:各離散化手法に共通の数値演算ライブラリ群
  - ppOpen-AT:科学技術計算のための自動チューニング(AT)機構
  - ppOpen-SYS:ノード間通信，耐故障機能に関連するライブラリ群


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

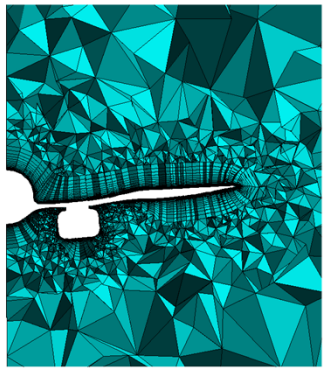
対象とする離散化手法
局所的，隣接通信中心，疎行列



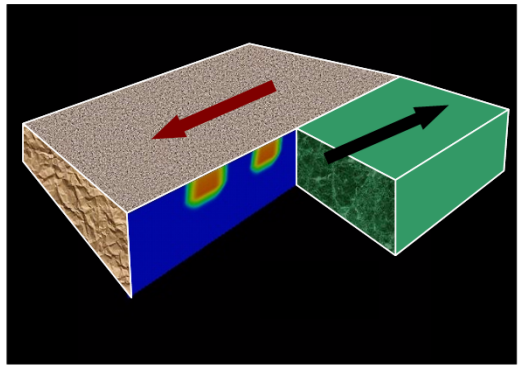
有限要素法
Finite Element Method
FEM



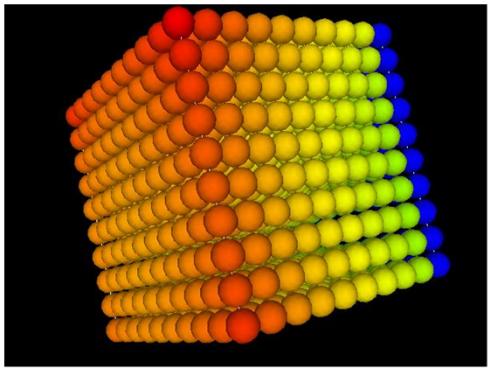
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



境界要素法
Boundary Element Method
BEM



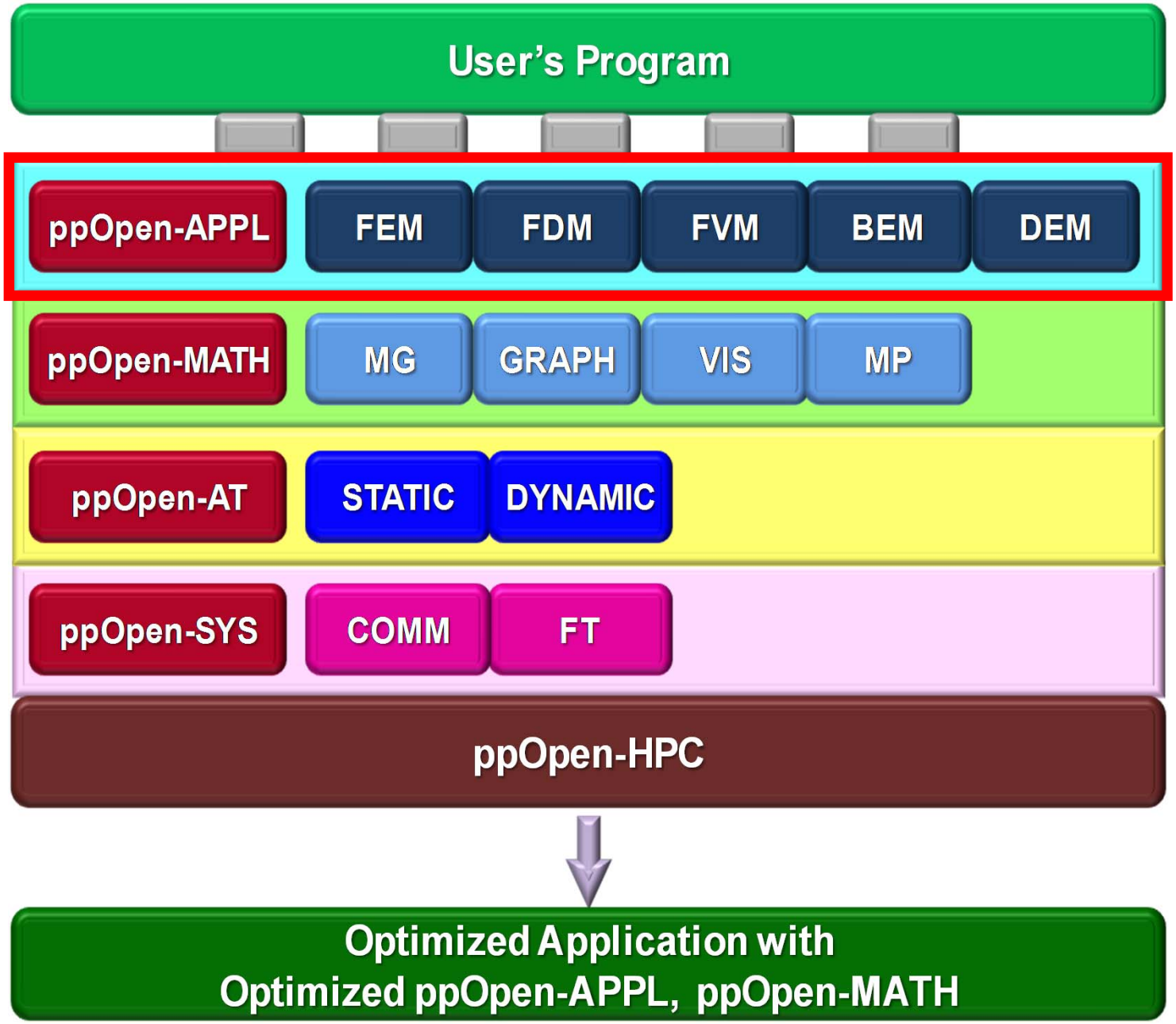
個別要素法
Discrete Element Method
DEM

概 要(2/3)

- 先行研究において各メンバーが開発した大規模アプリケーションに基づきppOpen-APPLの各機能を開発，実装
  - 各離散化手法の特性に基づき開発・最適化
    - 共通データ入出力インタフェース，領域間通信，係数マトリクス生成
    - 離散化手法の特性を考慮した前処理付き反復法
    - 適応格子，動的負荷分散
  - 実際に動いているアプリケーションから機能を切り出す
  - 各メンバー開発による既存ソフトウェア資産の効率的利用
    - GeoFEM，HEC-MW，HPC-MW，DEMIGLACE，ABCLibScript
- ppOpen-ATはppOpen-APPLの原型コードを対象として研究開発を実施し，その知見を各ppOpen-APPLの開発，最適化に適用
  - 自動チューニング技術により，様々な環境下における最適化ライブラリ・アプリケーション自動生成を目指す

概 要(3/3)

- 平成24年11月にマルチコアクラスタ向けに各グループの開発したppOpen-APPL，ppOpen-AT，ppOpen-MATHの各機能を公開(Ver.0.1.0)
  - http://ppopenhpc.cc.u-tokyo.ac.jp/
  - 平成25年11月：Ver.0.2.0
  - 平成26年11月：Ver.0.3.0
- 現在は各機能の最適化，機能追加，ppOpen-APPLによるアプリケーション開発とともに，Intel Xeon/Phi等メニーコア向けバージョンを開発中


User's Program
ppOpen-APPL
FEM
FDM
FVM
BEM
DEM
ppOpen-MATH
MG
GRAPH
VIS
MP
ppOpen-AT
STATIC
DYNAMIC
ppOpen-SYS
COMM
FT
ppOpen-HPC
Optimized Application with
Optimized ppOpen-APPL, ppOpen-MATH

ppOpen-APPL

- A set of libraries corresponding to each of the five methods noted above (FEM, FDM, FVM, BEM, DEM), providing:
  - I/O
    - netCDF-based Interface
  - Domain-to-Domain Communications
  - Optimized Linear Solvers (Preconditioned Iterative Solvers)
    - Optimized for each discretization method
  - H-Matrix Solvers in ppOpen-APPL/BEM
  - Matrix Assembling
  - AMR and Dynamic Load Balancing
- Most of components are extracted from existing codes developed by members

FEM Code on ppOpen-HPC

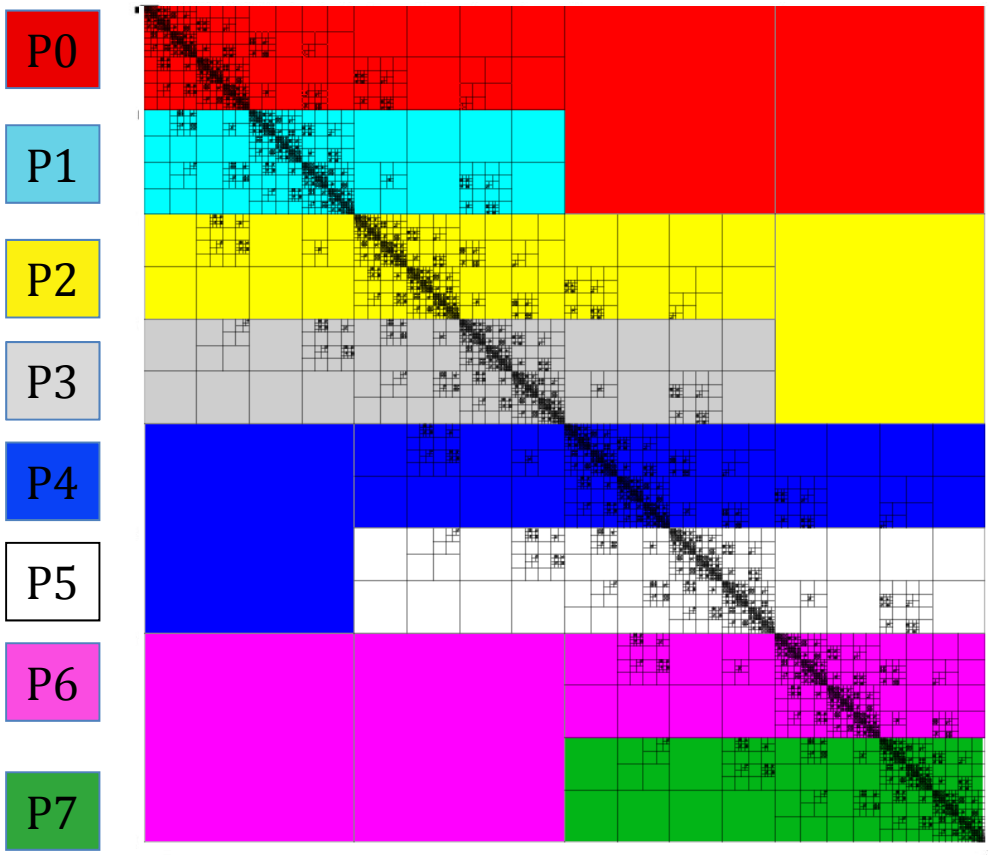
Optimization/parallelization could be hidden from application developers

```
Program My_pFEM
use ppOpenFEM_util
use ppOpenFEM_solver

call ppOpenFEM_init
call ppOpenFEM_cntl
call ppOpenFEM_mesh
call ppOpenFEM_mat_init

do
  call Users_FEM_mat_ass
  call Users_FEM_mat_bc
  call ppOpenFEM_solve
  call ppOPenFEM_vis
  Time= Time + DT
enddo

call ppOpenFEM_finalize
stop
end
```

ppOpen-APPL

- A set of libraries corresponding to each of the five methods noted above (FEM, FDM, FVM, BEM, DEM), providing:
  - I/O
    - netCDF-based Interface
  - Domain-to-Domain Communications
  - Optimized Linear Solvers (Preconditioned Iterative Solvers)
    - Optimized for each discretization method
  - **H-Matrix Solvers in ppOpen-APPL/BEM**
  - Matrix Assembling
  - AMR and Dynamic Load Balancing
- Most of components are extracted from existing codes developed by members

How to distribute sub-matrices to each processor



Assign a^c to P_k if $S(a^c{}_{1j})$ 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

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

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

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

ppOpen-HPC

c/o T. Iwashita & A. Ida

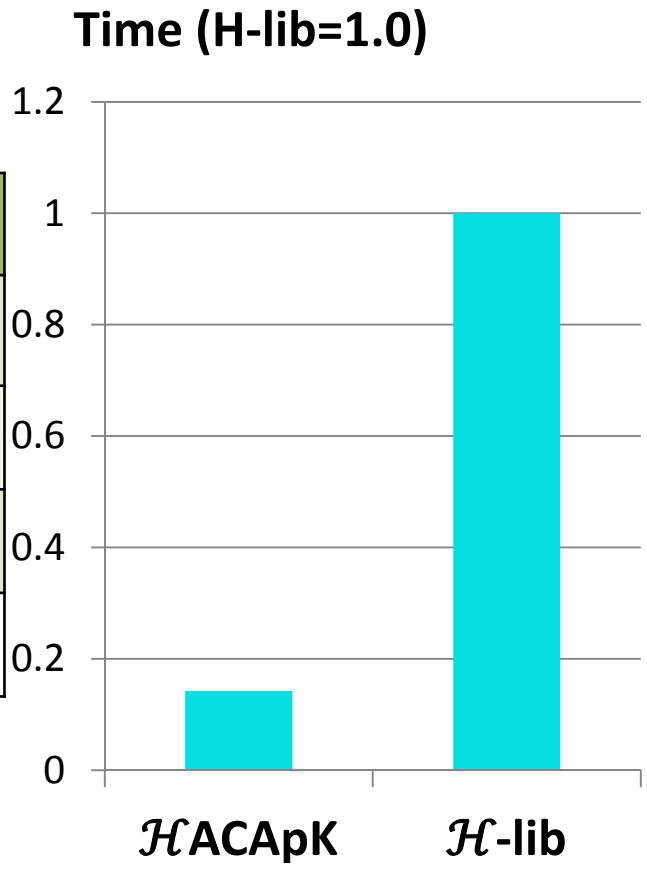
Calculation time of $\mathcal{H}$ACApK

c/o T. Iwashita & A. Ida

■ Comparison between $\mathcal{H}$ACApK and $\mathcal{H}$-lib (conventional library)

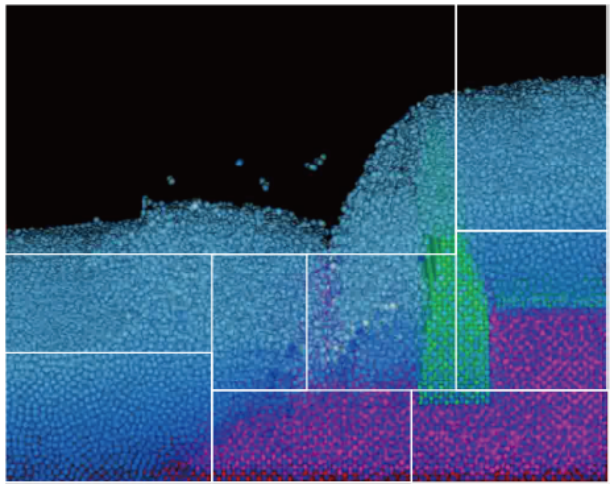
Calculation time of Mat−Vec Multiplication

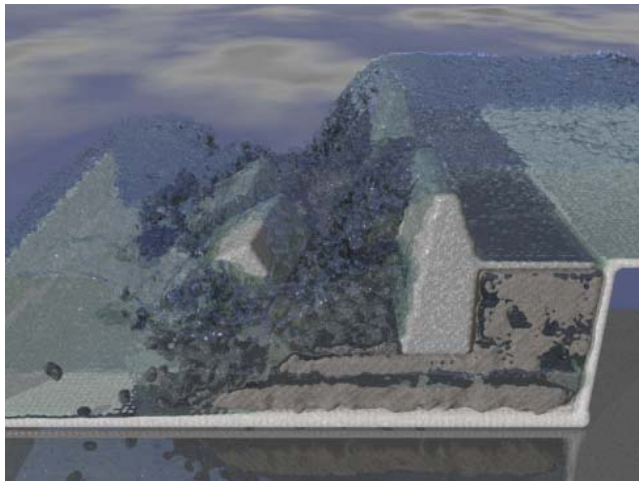
ε_{ACA}	N=2,400		N=2,1600	
	Hacapk	H-lib	Hacapk	H-lib
1.0E-3	2.29e-3	6.64e-3	3.21e-2	2.90e-1
1.0E-4	2.91e-3	7.94e-3	4.09e-2	3.26e-1
1.0E-5	3.42e-3	8.60e-3	4.92e-2	3.45e-1

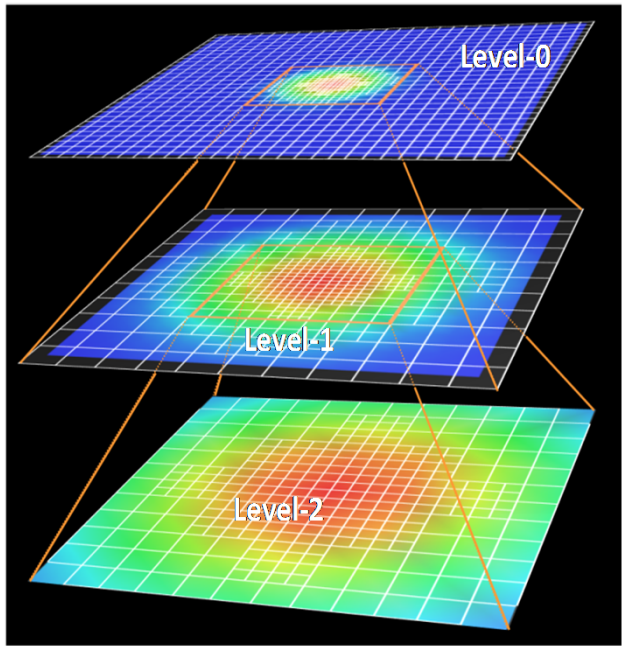


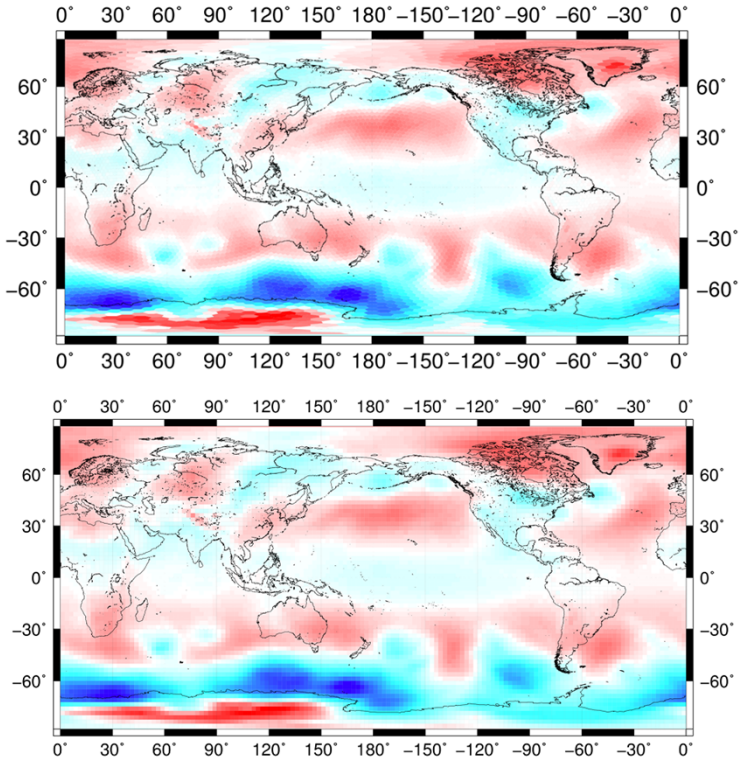
Target Applications

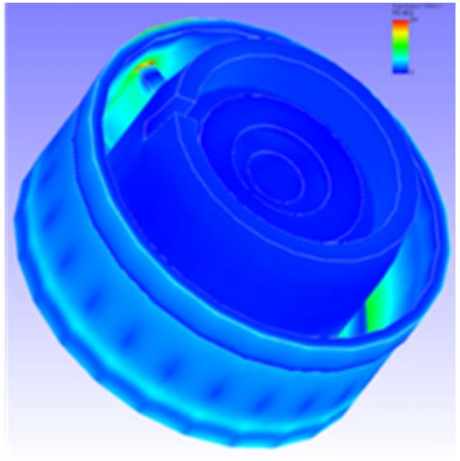
- Our goal is not development of applications, but we need some target appl. for evaluation of ppOpen-HPC.
- ppOpen-APPL/FEM
  - Incompressible Navier-Stokes
  - Heat Transfer, Solid Mechanics (Static, Dynamic)
- ppOpen-APPL/FDM
  - Incompressible Navier-Stokes
  - Transient Heat Transfer, Solid Mechanics (Dynamic)
- ppOpen-APPL/FVM
  - Compressible Navier-Stokes, Heat Transfer
- ppOpen-APPL/BEM
  - Electromagnetics, Solid Mechanics (Quasi Static) (Earthquake Generation Cycle)
- ppOpen-APPL/DEM
  - Incompressible Navier-Stokes, Solid Mechanics (Dynamic)

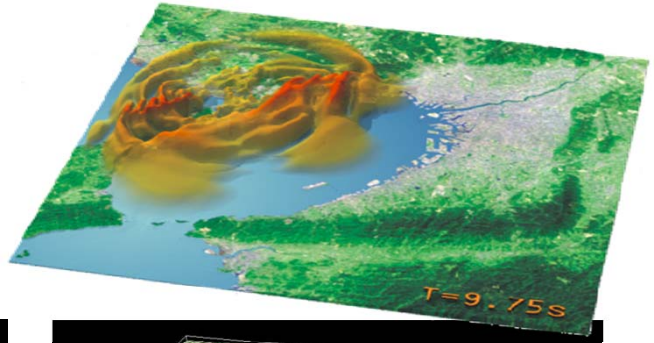


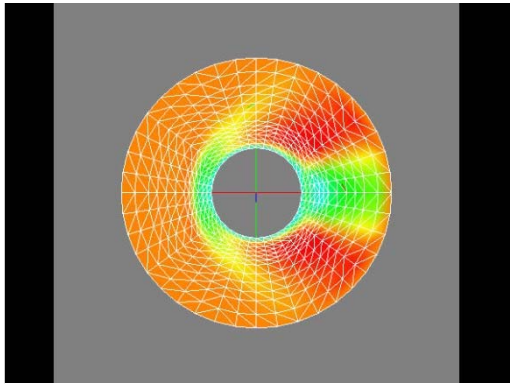












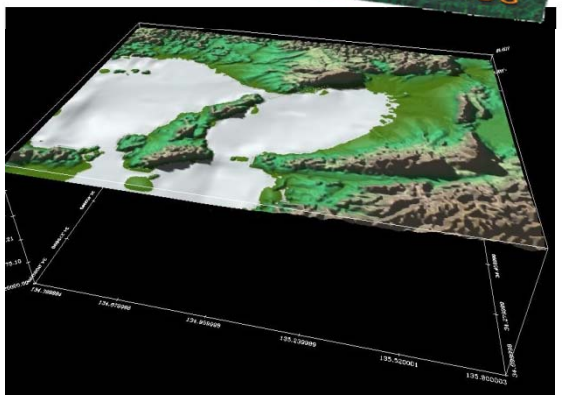
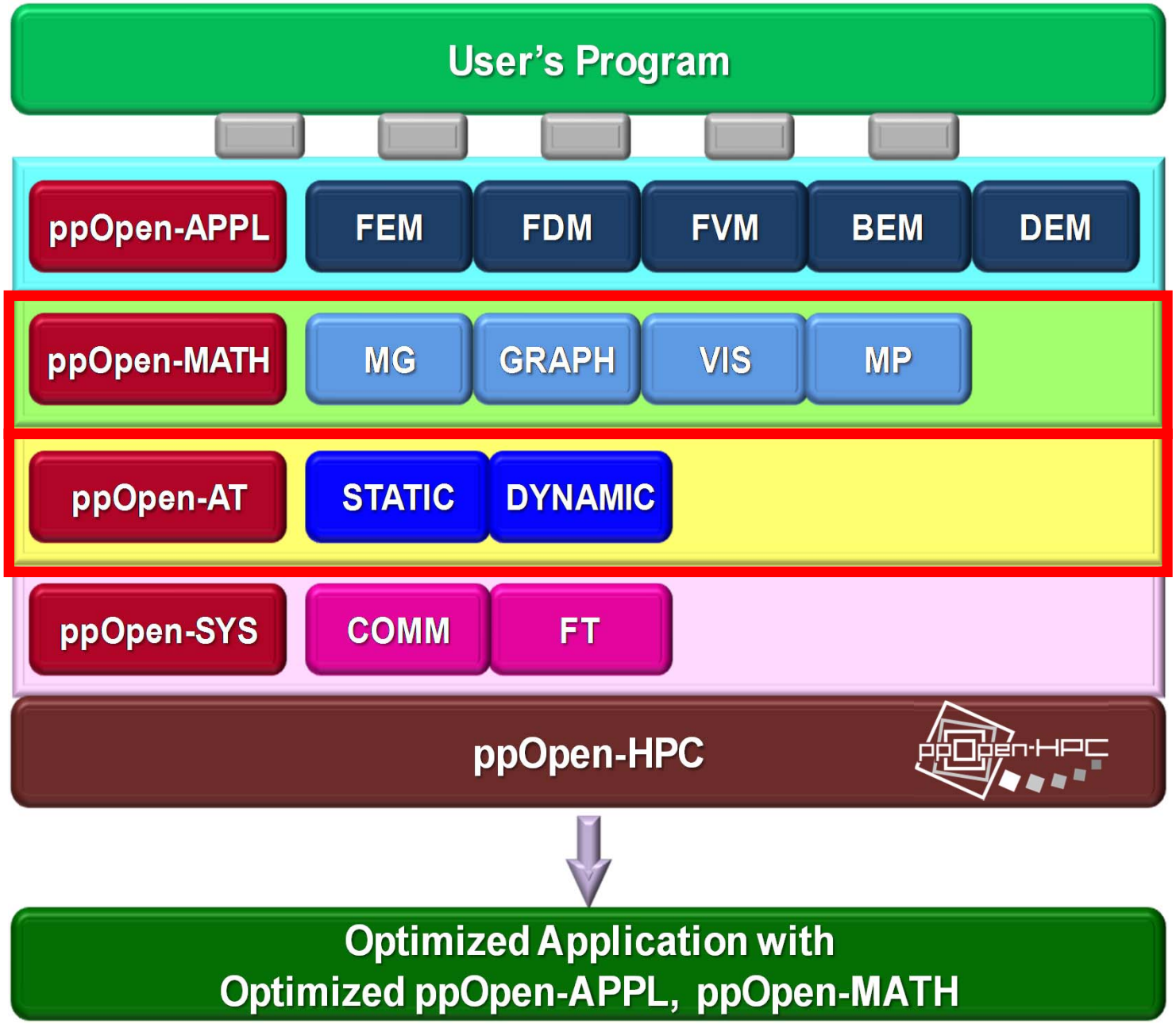


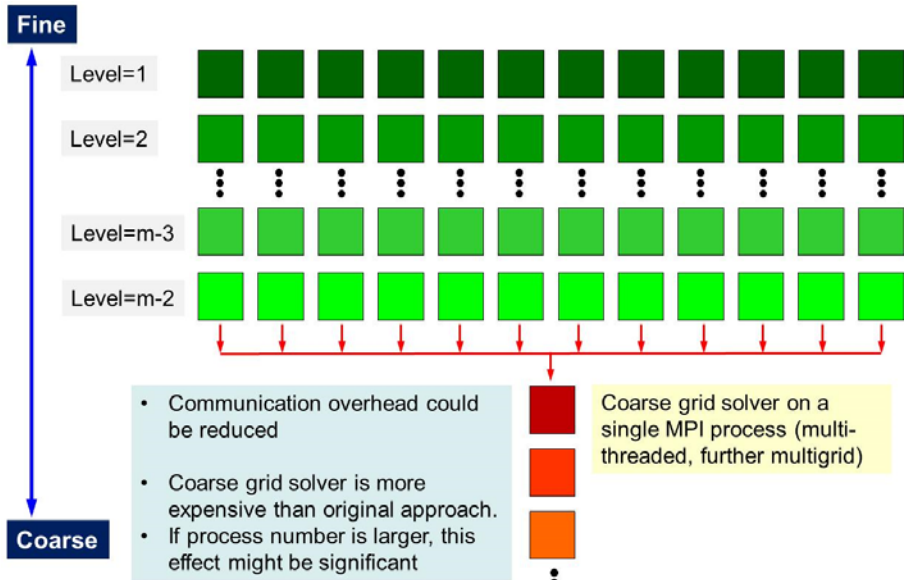
図 1　NICAM と IO モジュールの海面気圧


User's Program
ppOpen-APPL
FEM
FDM
FVM
BEM
DEM
ppOpen-MATH
MG
GRAPH
VIS
MP
ppOpen-AT
STATIC
DYNAMIC
ppOpen-SYS
COMM
FT
ppOpen-HPC
ppOpen-HPC
Optimized Application with
Optimized ppOpen-APPL, ppOpen-MATH

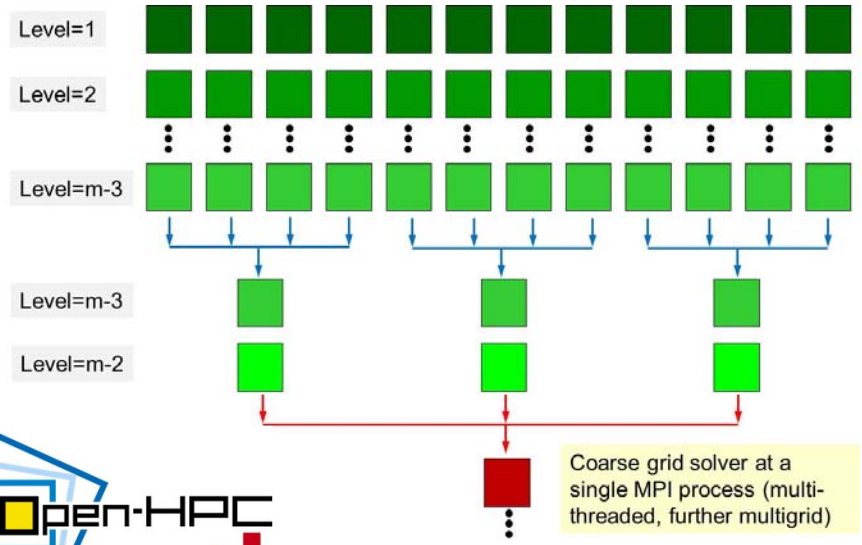
ppOpen-MATH

- A set of common numerical libraries
  - Multigrid solvers (ppOpen-MATH/MG) (Later)
  - Parallel graph libraries (ppOpen-MATH/GRAPH)
    - Multithreaded RCM for reordering (under development)
  - Parallel visualization (ppOpen-MATH/VIS)
  - Library for coupled multi-physics simulations (loose-coupling) (ppOpen-MATH/MP)
    - Originally developed as a coupler for NICAM (atmosphere, unstructured), and COCO (ocean, structured) in global climate simulations using K computer
      - Both codes are major codes on the K computer.
        - » Prof. Masaki Satoh (AORI/U.Tokyo): NICAM
        - » Prof. Hiroyasu Hasumi (AORI/U.Tokyo): COCO
    - Developed coupler is extended to more general use.
      - Coupled seismic simulations

ppOpen-MATH/MG (with CR/SR)



CGA (Coarse Grid Aggregation)

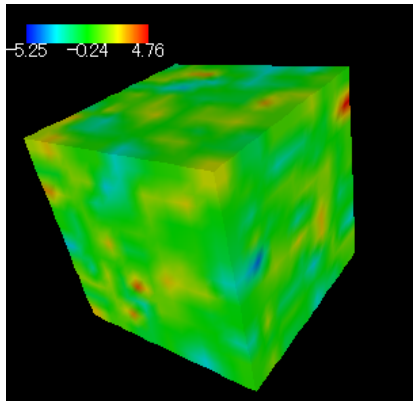


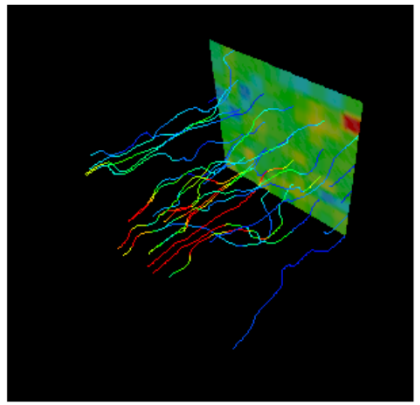
*h*CGA (Hierarchical CGA)

MGCG Solver with CGA/*h*CGA on 4,096 nodes (65,536 cores) of Fujitsu FX10 (Oakleaf-FX)

3D Groundwater Flow through Heterogeneous Porous Media

Nakajima, K. "Optimization of Serial and Parallel Communications for Parallel Geometric Multigrid Method" (Best Paper Award, ICPADS 2014)



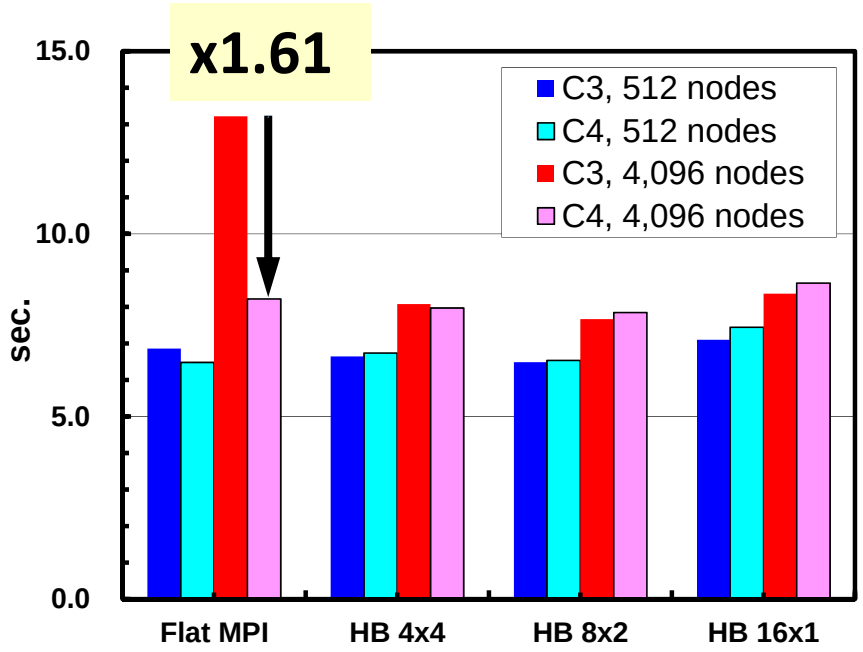


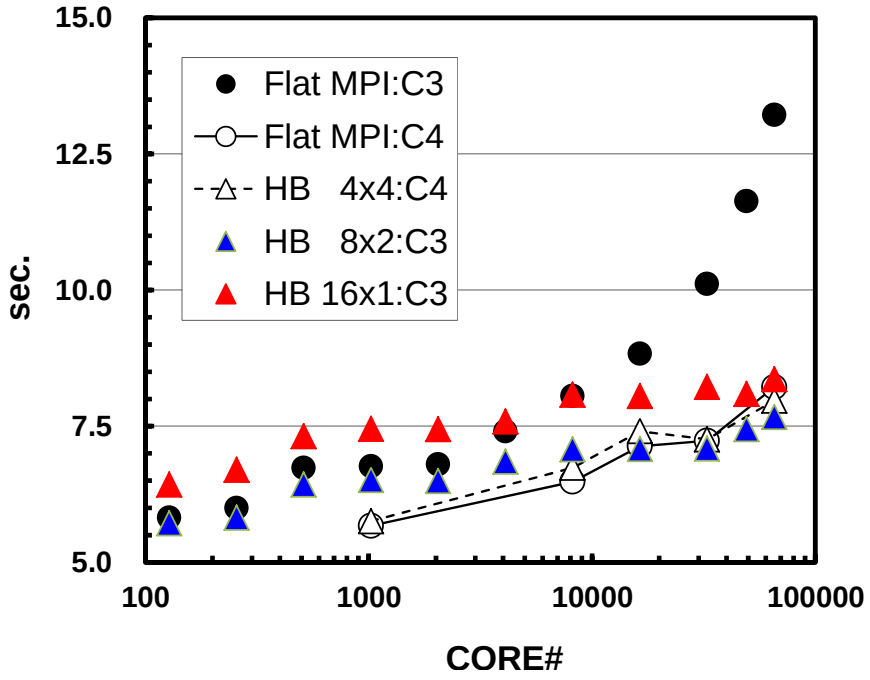
Weak Scaling up to 4,096 nodes

max. 17,179,869,184 meshes (64^3 meshes/core)

DOWN is GOOD

	Matrix	Coarse Grid
C0	CRS	Single Core
C1	ELL (org)	Single Core
C2	ELL (org)	CGA
C3	**ELL (sliced)**	**CGA**
C4	**ELL (sliced)**	***h*CGA**





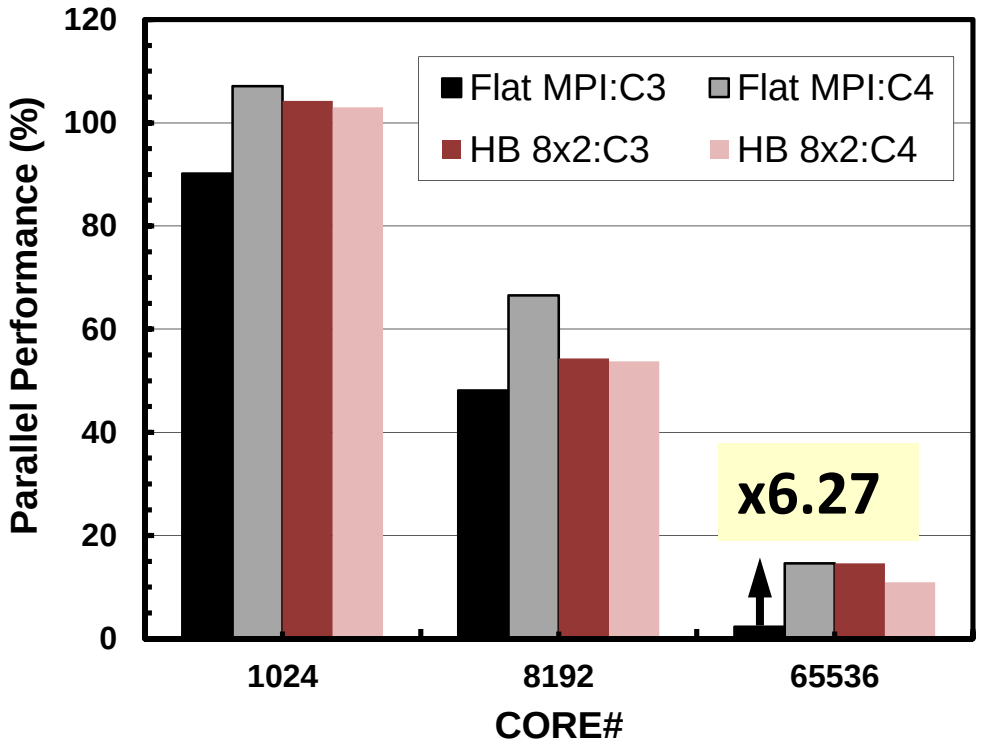
Strong Scaling up to 4,096 nodes

268,435,456 meshes, 16^3 meshes/core at 4,096 nodes

UP is GOOD

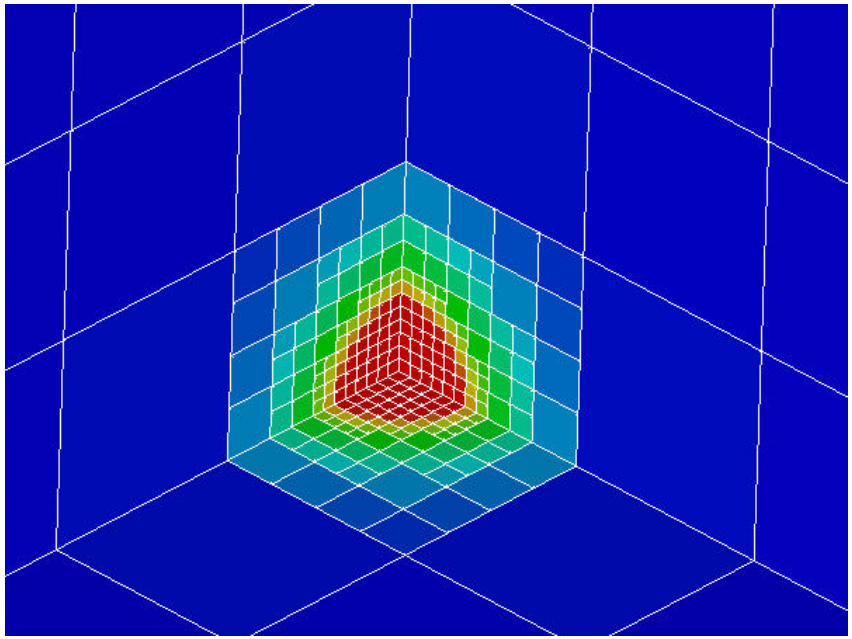
Flat MPI/ELL (C3),
8 nodes (128 cores) : 100%

	Matrix	Coarse Grid
C0	CRS	Single Core
C1	ELL (org)	Single Core
C2	ELL (org)	CGA
C3	**ELL (sliced)**	**CGA**
C4	**ELL (sliced)**	***h*CGA**



ppOpen-MATH/VIS

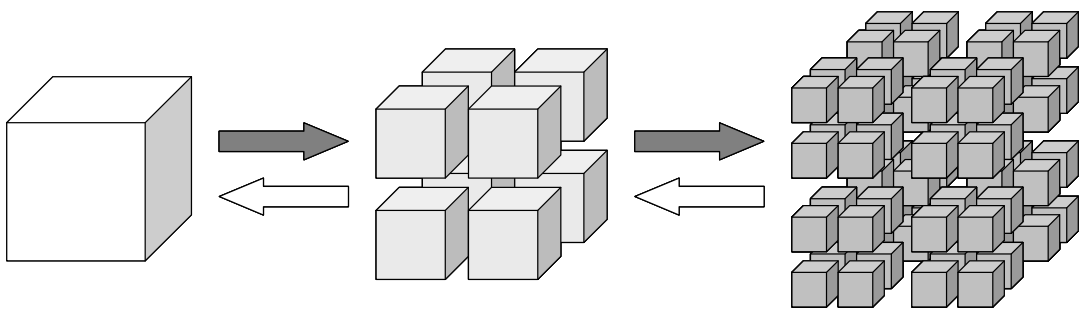
- ボクセル型背景格子を使用した大規模並列可視化手法〔Nakajima & Chen 2006〕に基づく
  - 差分格子用バージョン公開：ppOpen-MATH/VIS-FDM3D
- UCD single file
- プラットフォーム
  - T2K，Cray
  - FX10
  - Flat MPI
    - Hybrid，非構造格子：今年度実施

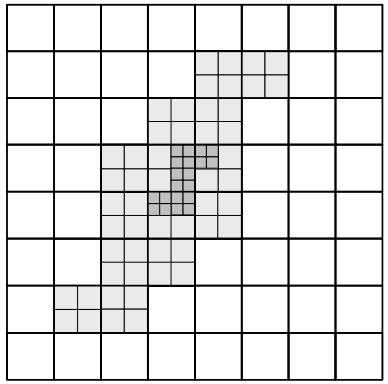


```
[Refine]
AvailableMemory = 2.0    Available memory size (GB), not available in this version.
MaxVoxelCount   = 500    Maximum number of voxels
MaxRefineLevel  = 20     Maximum number of refinement levels
```

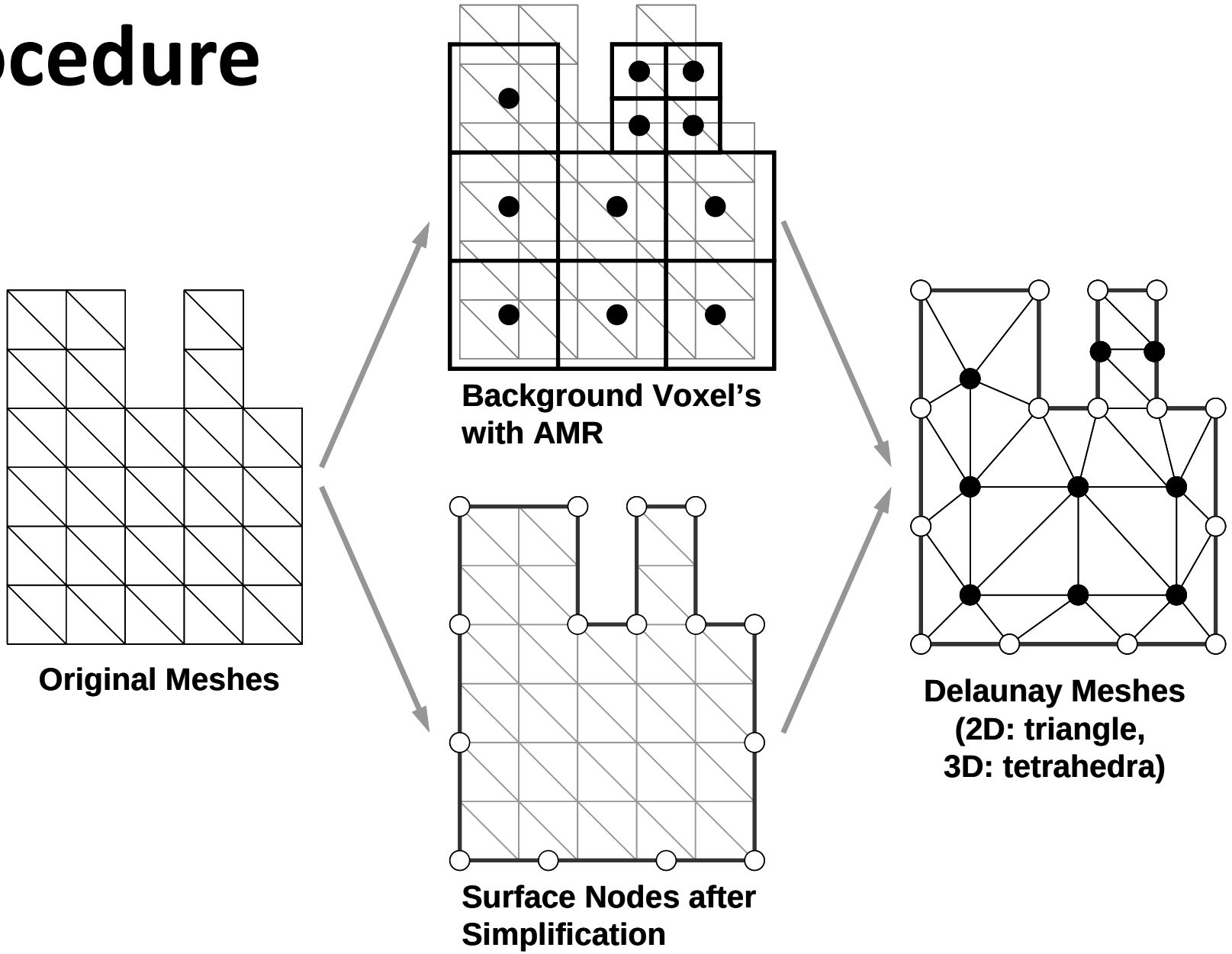
Simplified Parallel Visualization using Background Voxels

- Octree-based AMR
- AMR applied to the region where gradient of field values are large
  - stress concentration, shock wave, separation etc.
- If the number of voxels are controled, a single file with 10^5 meshes is possible, even though entire problem size is 10^9 with distributed data sets.

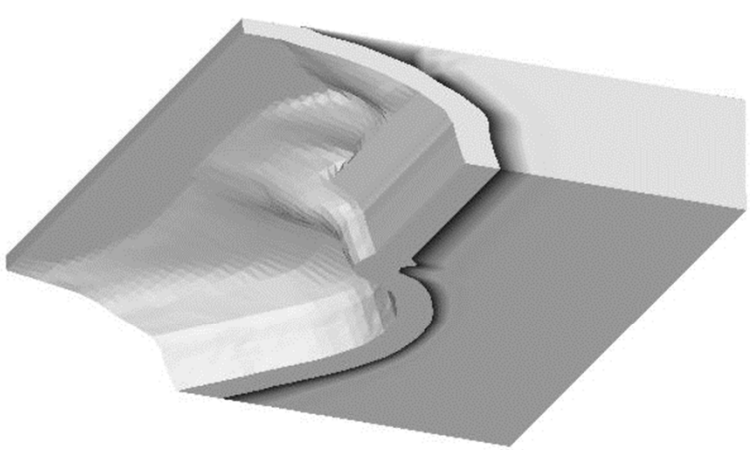


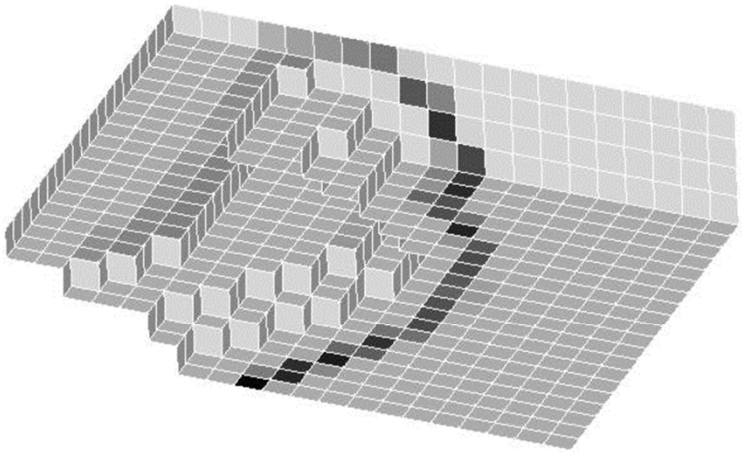


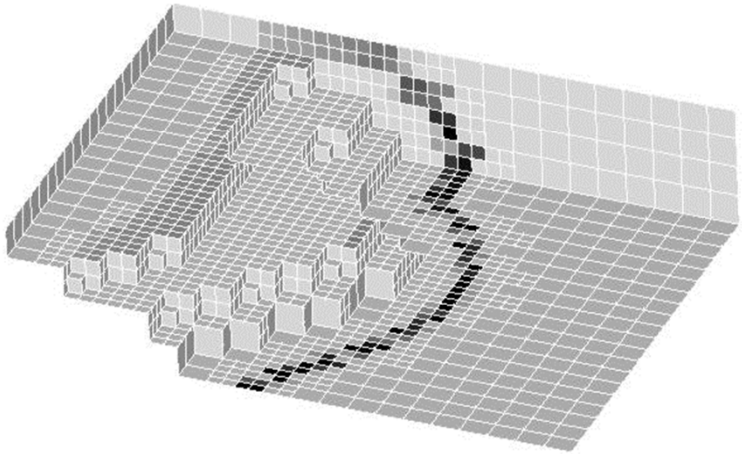
Procedure

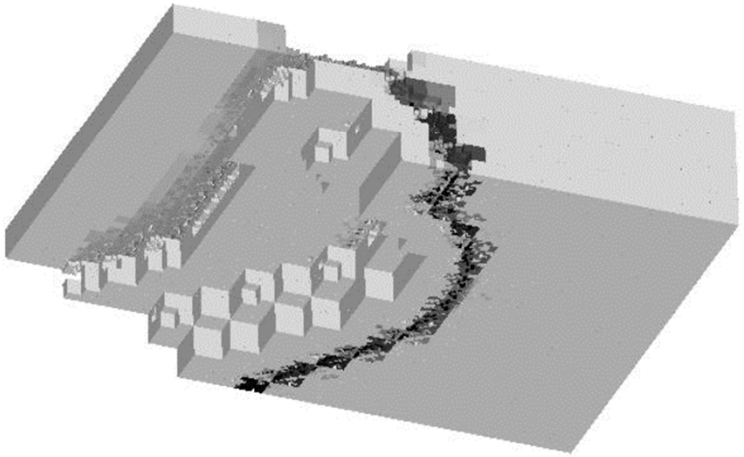

Original Meshes
Background Voxel's with AMR
Surface Nodes after Simplification
Delaunay Meshes (2D: triangle, 3D: tetrahedra)

Voxel Mesh (adapted)

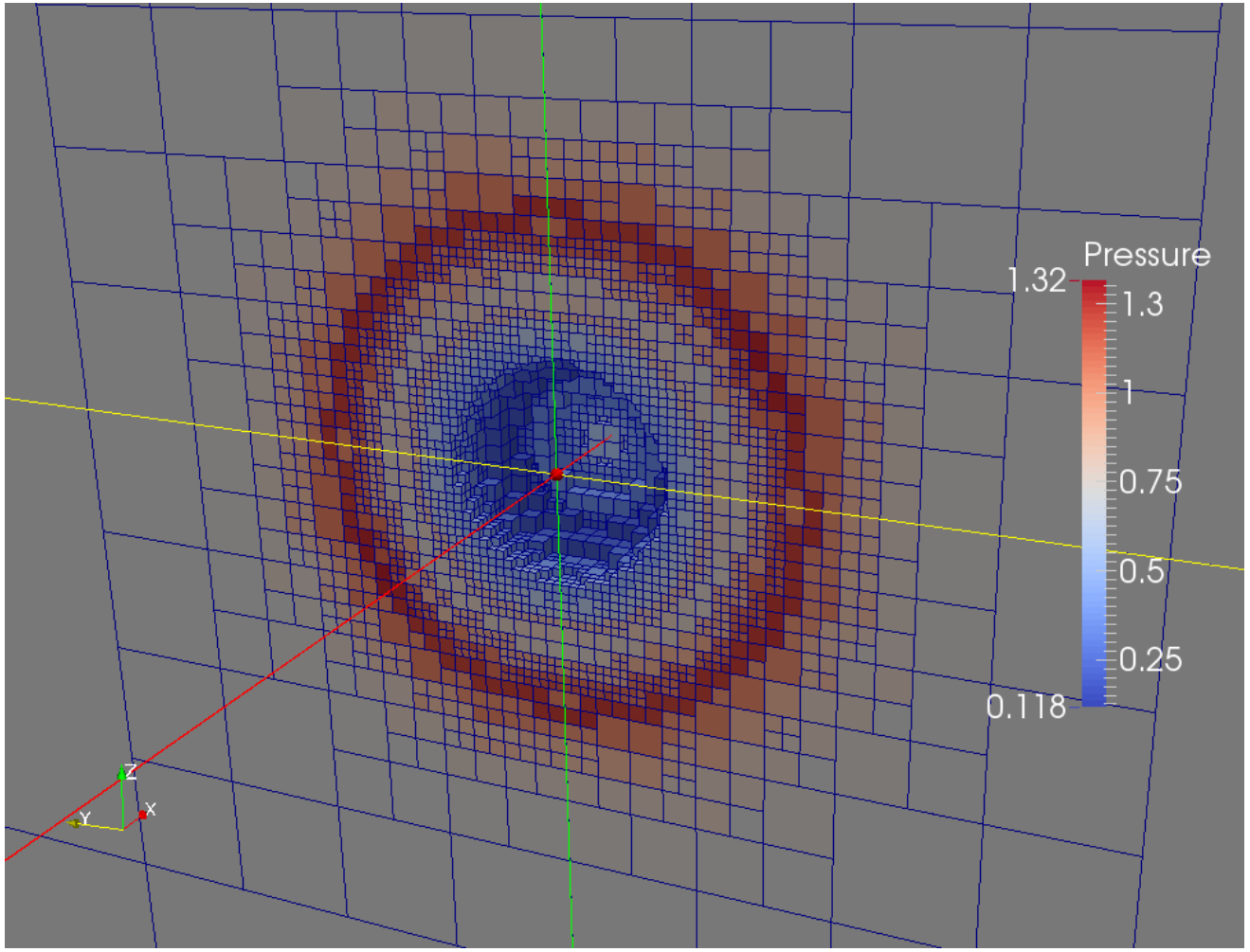








Flow around a sphere



ppOpen-MATH

- A set of common numerical libraries
  - Multigrid solvers (ppOpen-MATH/MG)
  - Parallel graph libraries (ppOpen-MATH/GRAPH)
    - Multithreaded RCM for reordering (under development)
  - Parallel visualization (ppOpen-MATH/VIS)
  - Library for coupled multi-physics simulations (loose-coupling) (ppOpen-MATH/MP)
    - Originally developed as a coupler for NICAM (atmosphere, unstructured), and COCO (ocean, structured) in global climate simulations using K computer
      - Both codes are major codes on the K computer.
        - » Prof. Masaki Satoh (AORI/U.Tokyo): NICAM
        - » Prof. Hiroyasu Hasumi (AORI/U.Tokyo): COCO
    - Developed coupler is extended to more general use.
      - Coupled seismic simulations

ppOpen-MATH/MP

- FVMやFEMなど異なる離散化手法を持つ複数のモデル結合，大規模データ転送，データ変換のための弱連成カップリングツールppOpen-MATH/MPを開発
  - 補間高速化アルゴリズム開発（H25公開）
  - モデルの出力をリアルタイムで受信し格子変換の後ファイルに出力するIOコンポーネント
- ケーススタディとして正二十面体格子大気モデルNICAMと海洋モデルCOCOを結合（H24）
  - H25以降は一般手法へ拡張

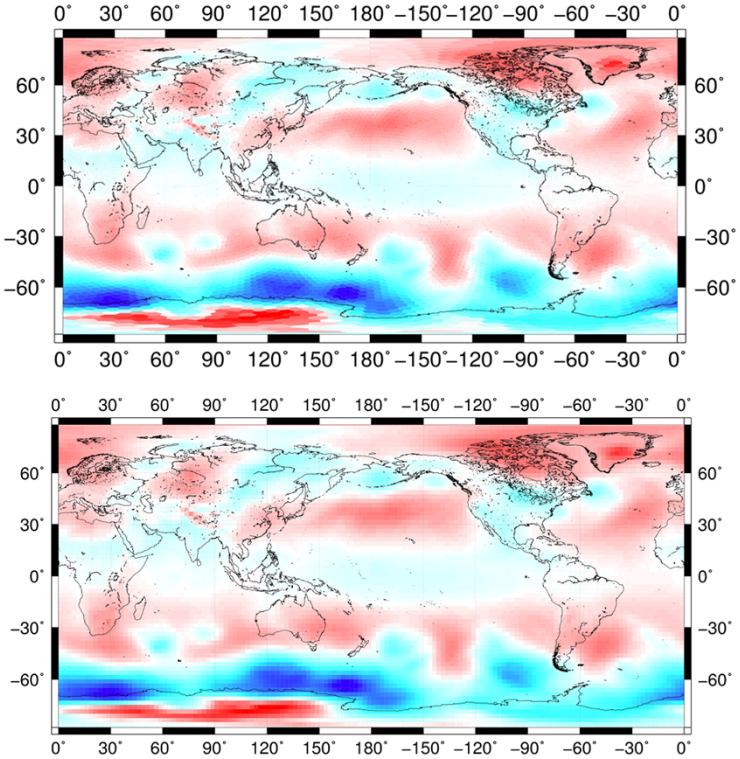
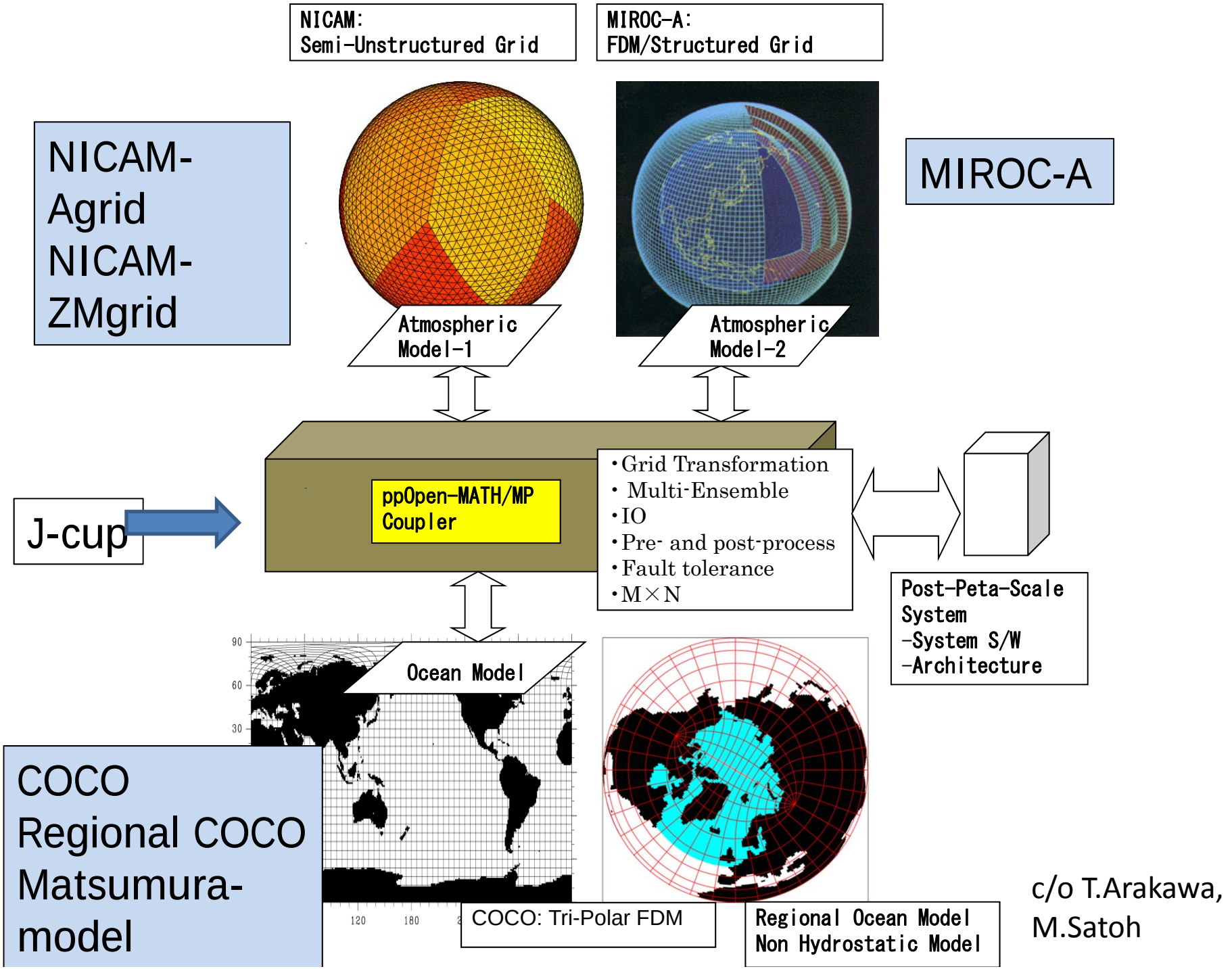


図 1　NICAM と IO モジュールの海面気圧

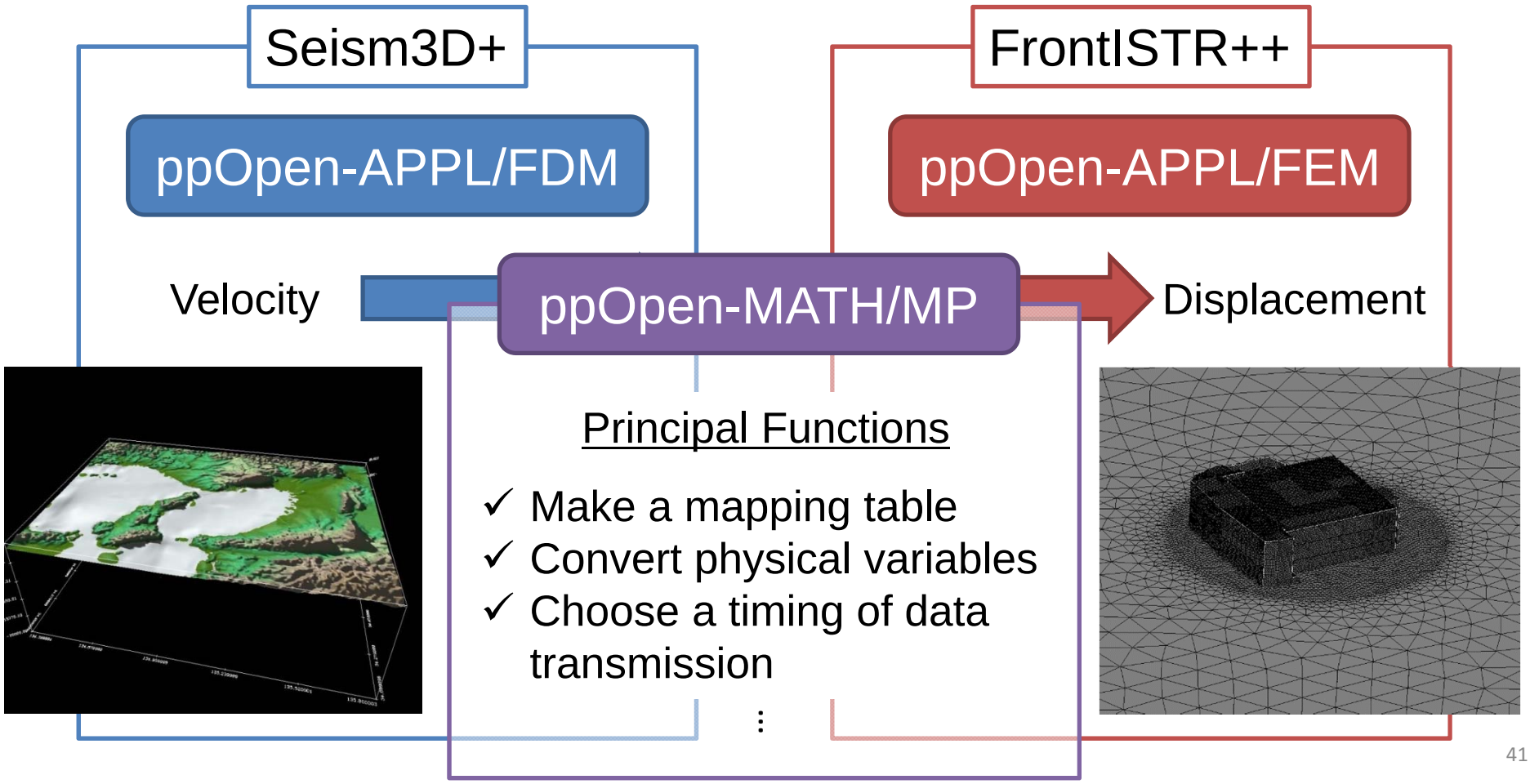
- 構造＋準構造格子を結合した先駆的事例，従来のスペクトル法では不可であった高解像度気候シミュレーションの可能性を開く。



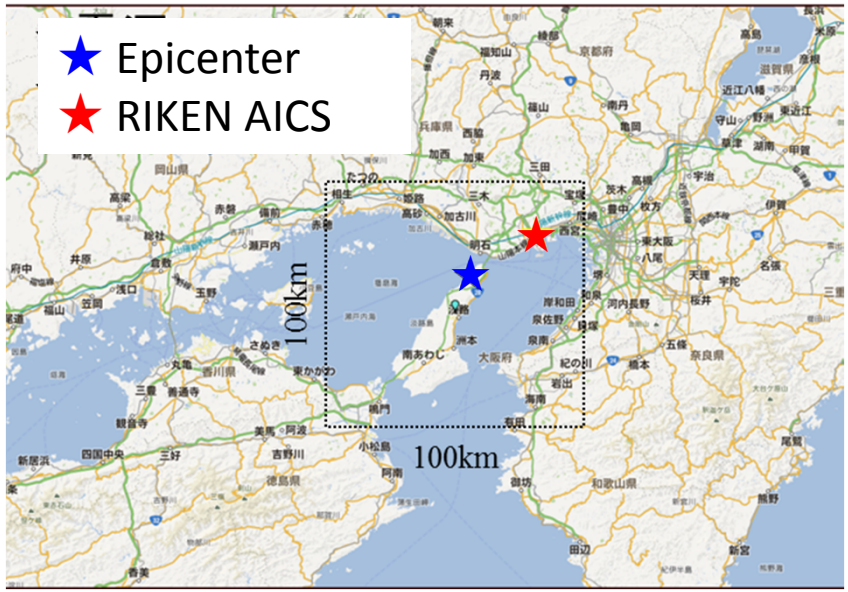
c/o T.Arakawa,
M.Satoh

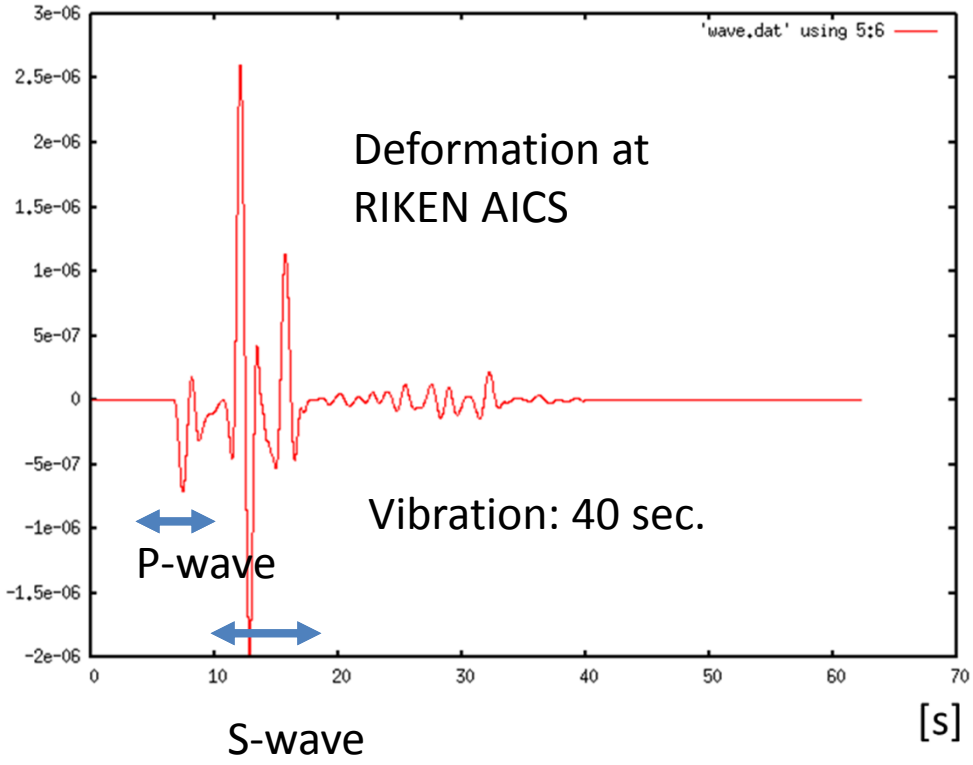
Weak-Coupled Simulation by the ppOpen-HPC Libraries

Two kinds of applications (Seism3D+ based on FDM, and FrontISTR++ based on FEM) are connected by the ppOpen-MATH/MP coupler.



1995 Kobe Earthquake M.7.3

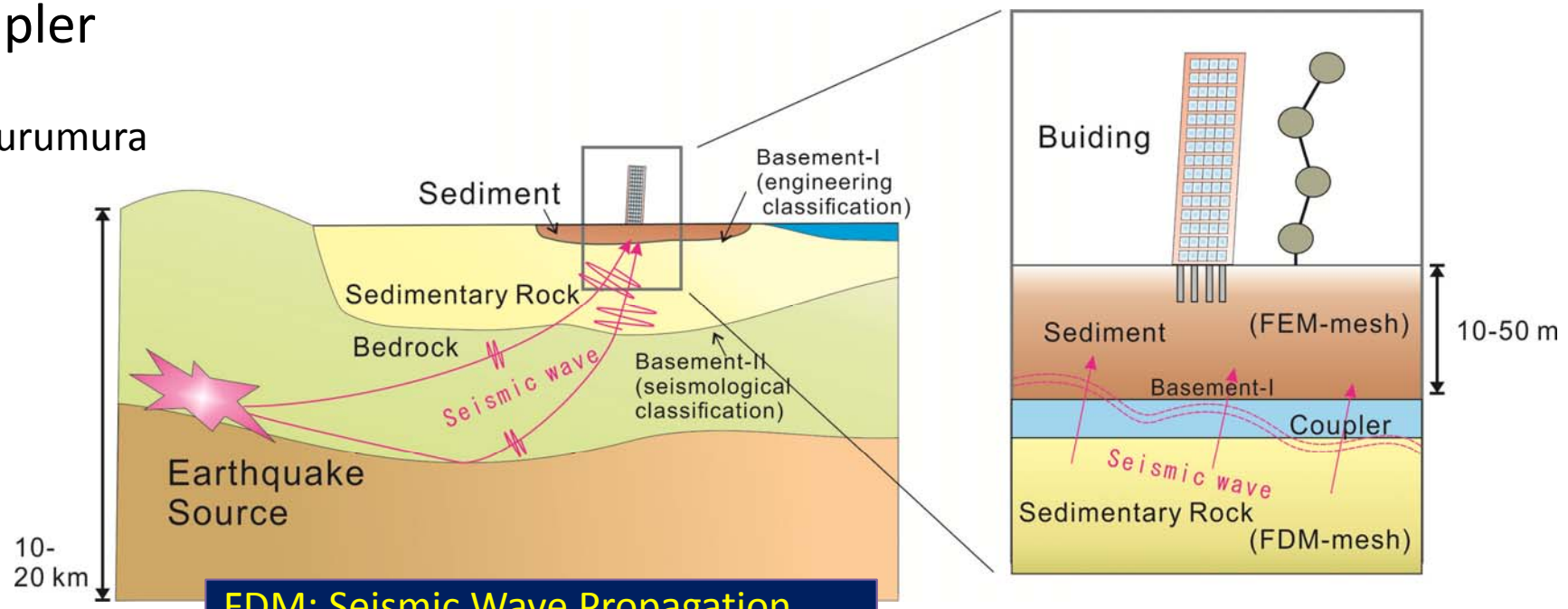


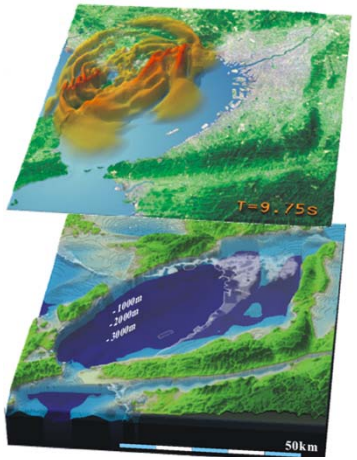


Large-Scale Coupled Simulations in FY.2013-2014

A test of a coupling simulation of FDM (regular grid) and FEM (unconstructed grid) using newly developed ppOpen-MATH/MP Coupler

c/o T.Furumura





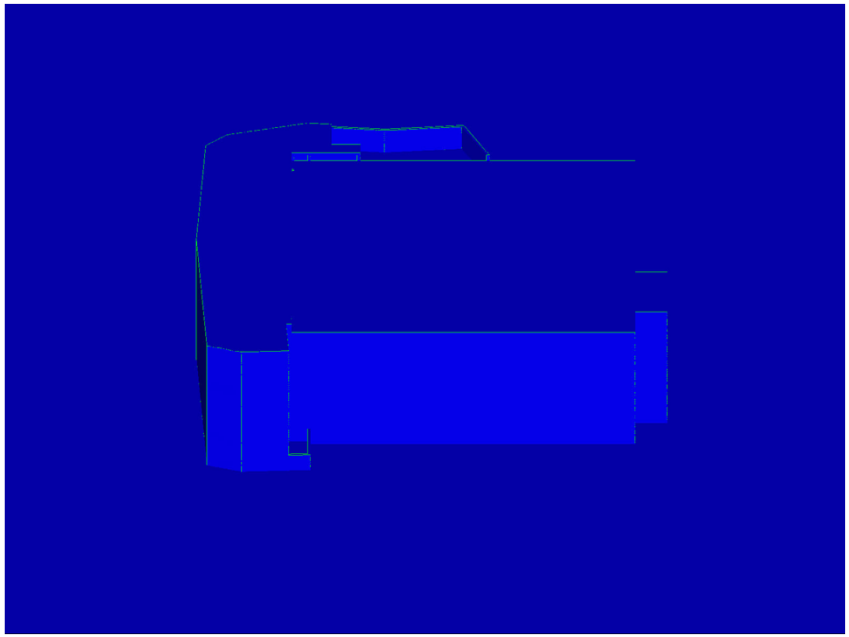
FDM: Seismic Wave Propagation
Model size: 60x60x32 km
Time : 90 s
Resolution (space) : 40x40x20 m
Resolution (time) : 1.00 ms

FEM: Building Response
Model size : 400x400x200 m
Time : 90 s
Resolution (space) : 1 m, 6M nodes
Resolution (time) : 0.2 ms

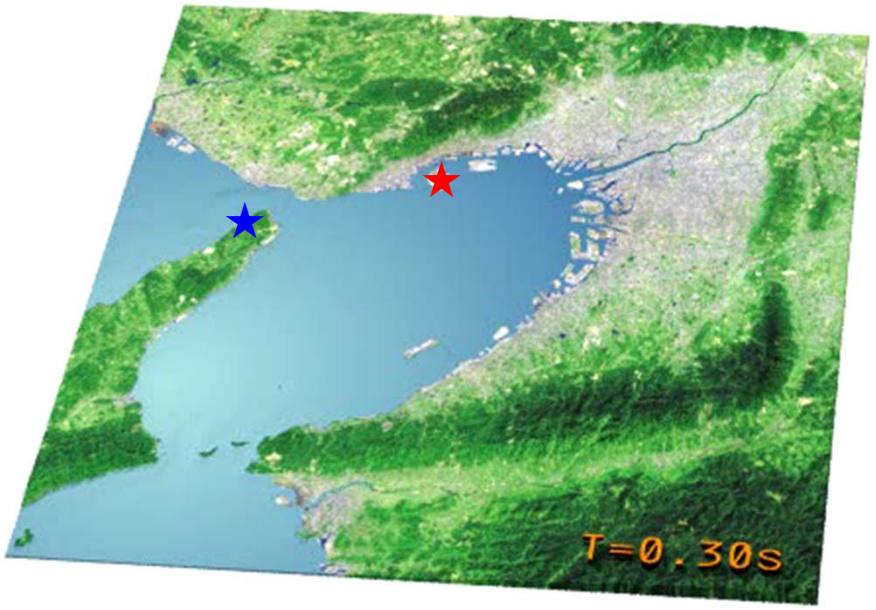
ppOpen-MATH/MP: Space-temporal interpolation, Mapping between FDM and FEM mesh, etc.

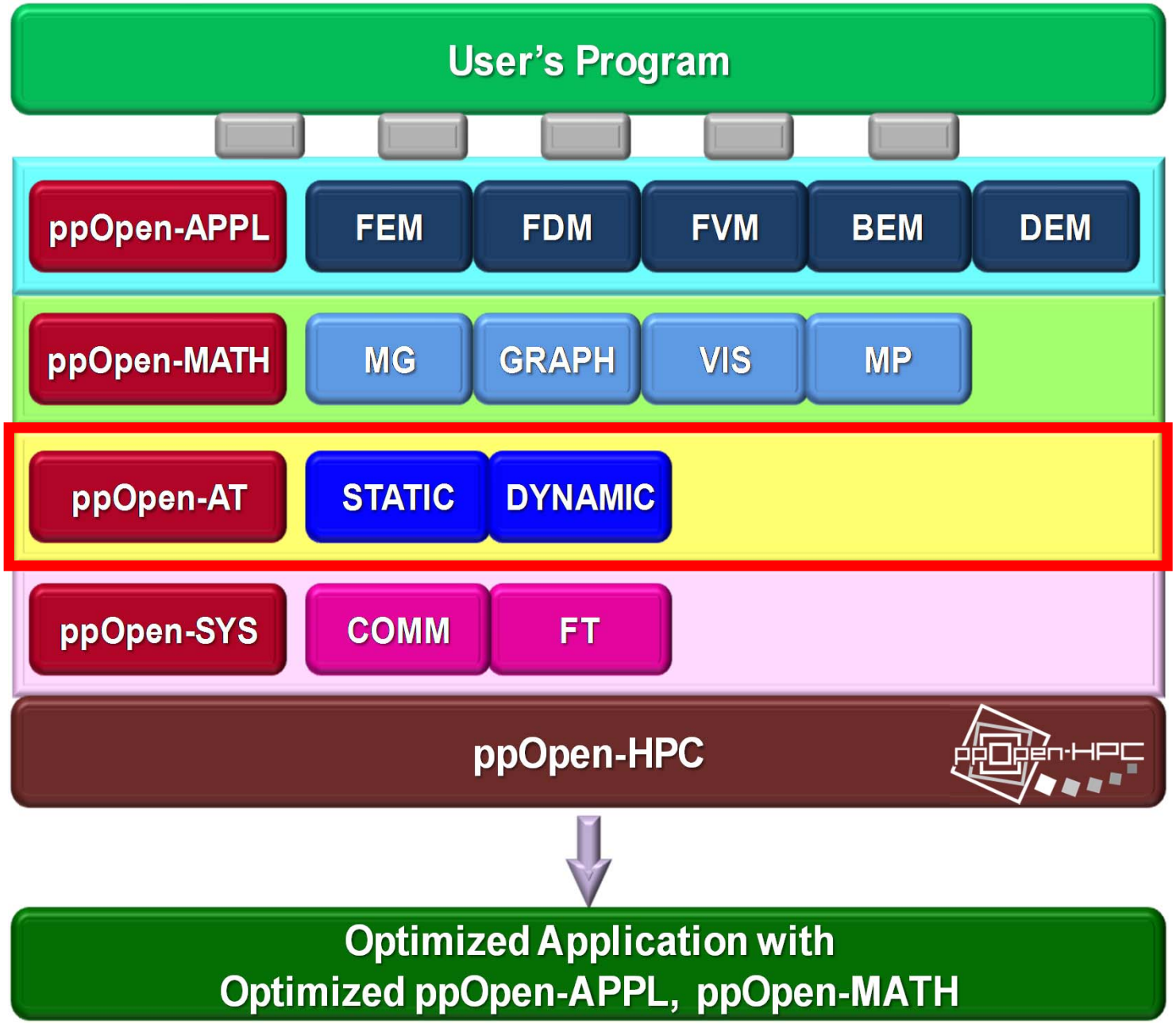
2,560 nodes for FDM, 2,000 nodes for FEM = 4,560 nodes of FX10

RIKEN AICS Building



★ Epicenter
★ RIKEN AICS



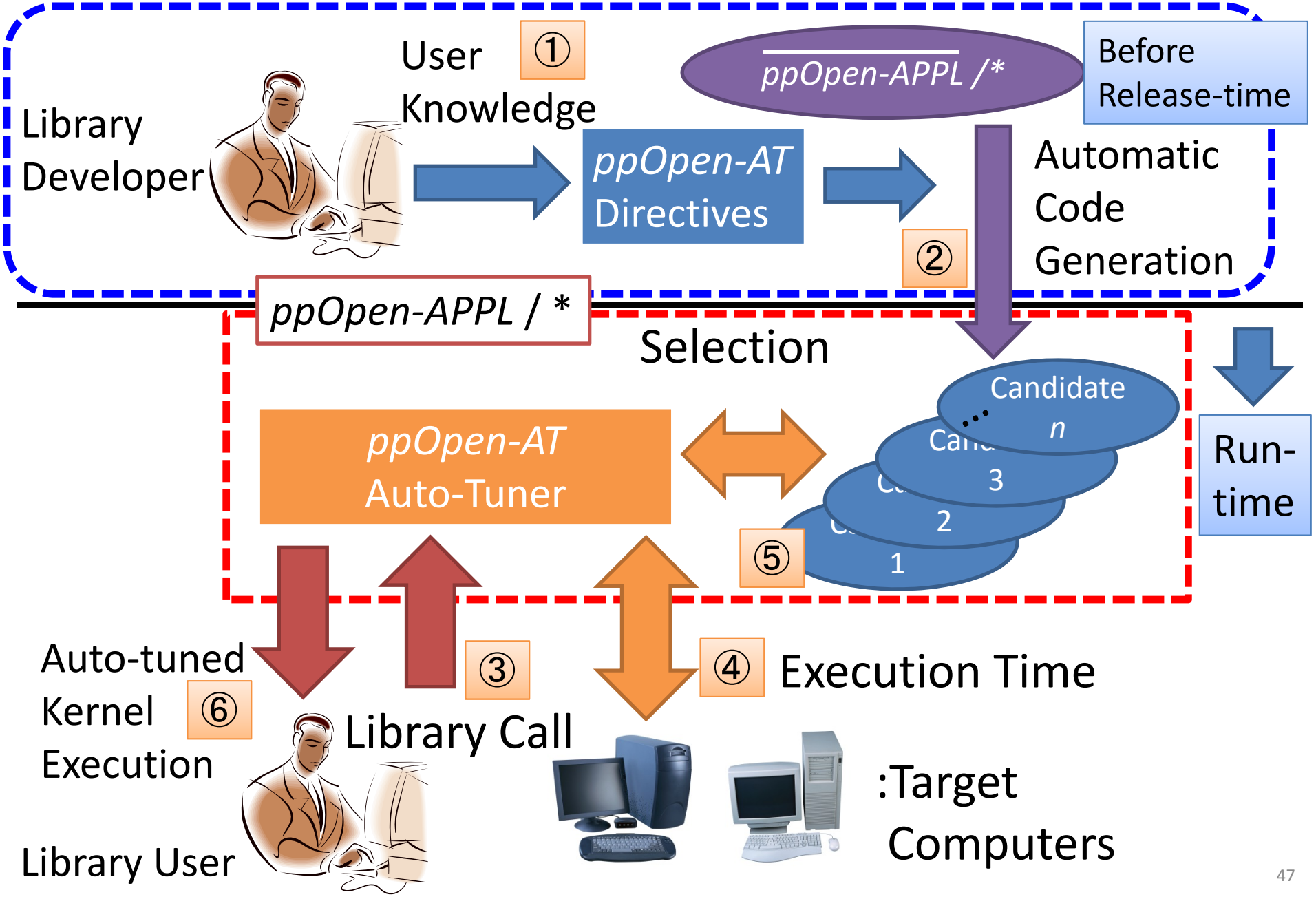

User's Program
ppOpen-APPL
FEM
FDM
FVM
BEM
DEM
ppOpen-MATH
MG
GRAPH
VIS
MP
ppOpen-AT
STATIC
DYNAMIC
ppOpen-SYS
COMM
FT
ppOpen-HPC
ppOpen-HPC
Optimized Application with
Optimized ppOpen-APPL, ppOpen-MATH

ppOpen-AT

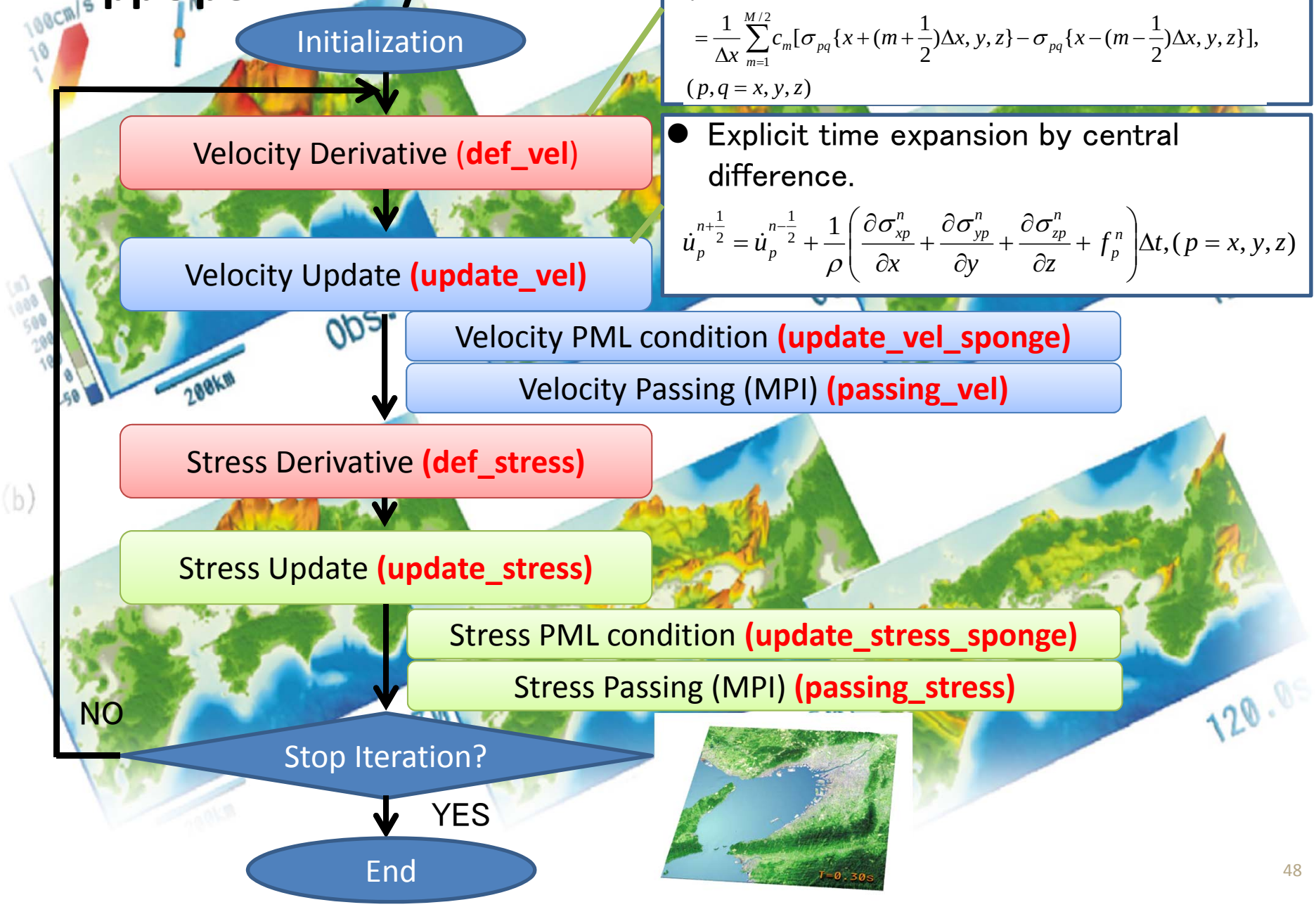
- Automatic Tuning (AT) enables development of optimized codes and libraries on emerging architectures
- ppOpen-AT automatically and adaptively generates optimum implementation for efficient memory accesses in procedures of scientific computing in each component of ppOpen-APPL
- A directive-based special AT language is developed
  - Well-known loop transformation techniques are utilized
- ppOpen-APPL/FDM, ppOpen-APPL/BEM
- AT applied to 3D FDM code for seismic simulations developed on ppOpen-APPL/FDM for Intel Xeon/Phi

c/o T.Katagiri

ppOpen-AT System



Seism3D on ppOpen-APPL/FDM



- Space difference by FDM.

$$\frac{d}{dx}\sigma_{pq}(x,y,z) = \frac{1}{\Delta x}\sum_{m=1}^{M/2} c_m[\sigma_{pq}\{x+(m+\frac{1}{2})\Delta x, y, z\} - \sigma_{pq}\{x-(m-\frac{1}{2})\Delta x, y, z\}], \quad (p,q = x,y,z)$$

- Explicit time expansion by central difference.

$$\dot{u}_p^{n+\frac{1}{2}} = \dot{u}_p^{n-\frac{1}{2}} + \frac{1}{\rho}\left(\frac{\partial \sigma_{xp}^n}{\partial x} + \frac{\partial \sigma_{yp}^n}{\partial y} + \frac{\partial \sigma_{zp}^n}{\partial z} + f_p^n\right)\Delta t, (p = x,y,z)$$

Seism3D: Code for Seismic Wave Sim. Triple-nested loops, Most Expensive

```
DO K = 1, NZ
 DO J = 1, NY
  DO I = 1, NX
     RL  = LAM (I,J,K)
     RM  = RIG (I,J,K)
     RM2 = RM + RM
     RMAXY = 4.0/(1.0/RIG(I,J,K) + 1.0/RIG(I+1,J,K)  + 1.0/RIG(I,J+1,K) + 1.0/RIG(I+1,J+1,K))
     RMAXZ = 4.0/(1.0/RIG(I,J,K) + 1.0/RIG(I+1,J,K)  + 1.0/RIG(I,J,K+1) + 1.0/RIG(I+1,J,K+1))
     RMAYZ = 4.0/(1.0/RIG(I,J,K) + 1.0/RIG(I,J+1,K)  + 1.0/RIG(I,J,K+1) + 1.0/RIG(I,J+1,K+1))
     RLTHETA = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL
     QG  = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
     SXX (I,J,K) = ( SXX (I,J,K)  + (RLTHETA + RM2*DXVX(I,J,K))*DT )*QG
     SYY (I,J,K) = ( SYY (I,J,K)  + (RLTHETA + RM2*DYVY(I,J,K))*DT )*QG
     SZZ (I,J,K) = ( SZZ (I,J,K)  + (RLTHETA + RM2*DZVZ(I,J,K))*DT )*QG
     SXY (I,J,K) = ( SXY (I,J,K)  + (RMAXY*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG
     SXZ (I,J,K) = ( SXZ (I,J,K)  + (RMAXZ*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG
     SYZ (I,J,K) = ( SYZ (I,J,K)  + (RMAYZ*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG
   END DO
  END DO
END DO
```

c/o T.Katagiri

Loop Splitting

```
      DO K = 1, NZ
      DO J = 1, NY
      DO I = 1, NX
           RL  = LAM (I,J,K)
           RM  = RIG (I,J,K)
           RM2 = RM + RM
           RLTHETA  = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL
           QG  = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
           SXX (I,J,K) = ( SXX (I,J,K)  + (RLTHETA + RM2*DXVX(I,J,K))*DT )*QG
           SYY (I,J,K) = ( SYY (I,J,K)  + (RLTHETA + RM2*DYVY(I,J,K))*DT )*QG
           SZZ (I,J,K) = ( SZZ (I,J,K)   + (RLTHETA + RM2*DZVZ(I,J,K))*DT )*QG
      ENDDO;  ENDDO; ENDDO

      DO K = 1, NZ
      DO J = 1, NY
      DO I = 1, NX
           STMP1 = 1.0/RIG(I,J,K)
           STMP2 = 1.0/RIG(I+1,J,K)
           STMP4 = 1.0/RIG(I,J,K+1)
           STMP3 = STMP1 + STMP2
           RMAXY = 4.0/(STMP3  + 1.0/RIG(I,J+1,K) + 1.0/RIG(I+1,J+1,K))
           RMAXZ = 4.0/(STMP3  + STMP4 + 1.0/RIG(I+1,J,K+1))
           RMAYZ = 4.0/(STMP3  + STMP4 + 1.0/RIG(I,J+1,K+1))
           QG  = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
           SXY (I,J,K) = ( SXY (I,J,K) + (RMAXY*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG
           SXZ (I,J,K) = ( SXZ (I,J,K) + (RMAXZ*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG
           SYZ (I,J,K) = ( SYZ (I,J,K)  + (RMAYZ*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG
      END DO;  END DO;  END DO;
```

c/o T.Katagiri

Loop Fusion: Double-nested

```
DO KK = 1, NZ * NY
  K = (KK-1)/NY +  1
  J = mod(KK-1,NY) + 1
  DO I = 1, NX
    RL  = LAM (I,J,K)
    RM  = RIG (I,J,K)
    RM2 = RM + RM
    RMAXY = 4.0/(1.0/RIG(I,J,K) + 1.0/RIG(I+1,J,K)  + 1.0/RIG(I,J+1,K) +
                 1.0/RIG(I+1,J+1,K))
    RMAXZ = 4.0/(1.0/RIG(I,J,K) + 1.0/RIG(I+1,J,K)  + 1.0/RIG(I,J,K+1) +
                 1.0/RIG(I+1,J,K+1))
    RMAYZ = 4.0/(1.0/RIG(I,J,K) + 1.0/RIG(I,J+1,K)  + 1.0/RIG(I,J,K+1) +
                 1.0/RIG(I,J+1,K+1))
    RLTHETA = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL
    QG  = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
    SXX (I,J,K) = ( SXX (I,J,K)  + (RLTHETA + RM2*DXVX(I,J,K))*DT )*QG
    SYY (I,J,K) = ( SYY (I,J,K)  + (RLTHETA + RM2*DYVY(I,J,K))*DT )*QG
    SZZ (I,J,K) = ( SZZ (I,J,K)  + (RLTHETA + RM2*DZVZ(I,J,K))*DT )*QG
    SXY (I,J,K) = ( SXY (I,J,K)  + (RMAXY*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG
    SXZ (I,J,K) = ( SXZ (I,J,K)  + (RMAXZ*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG
    SYZ (I,J,K) = ( SYZ (I,J,K)  + (RMAYZ*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG
  ENDDO
END DO
```

Longer loops

Inner loop: nice for prefetching

c/o T.Katagiri

Example of Directives of ppOpen-AT

```
!oat$ install LoopFusionSplit region start
!$omp parallel do private(k,j,i,STMP1,STMP2,STMP3,STMP4,RL,RM,RM2,RMAXY,RMAXZ,RMAYZ,RLTHETA,QG)
      DO K = 1, NZ
      DO J = 1, NY
      DO I = 1, NX
           RL  = LAM (I,J,K);   RM  = RIG (I,J,K);   RM2 = RM + RM
           RLTHETA  = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL
!oat$ SplitPointCopyDef  region  start
           QG  = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
!oat$ SplitPointCopyDef  region  end
           SXX (I,J,K) = ( SXX (I,J,K) + (RLTHETA + RM2*DXVX(I,J,K))*DT )*QG
           SYY (I,J,K) = ( SYY (I,J,K) + (RLTHETA + RM2*DYVY(I,J,K))*DT )*QG
           SZZ (I,J,K) = ( SZZ (I,J,K) + (RLTHETA + RM2*DZVZ(I,J,K))*DT )*QG
!oat$ SplitPoint  (K, J, I)
           STMP1 = 1.0/RIG(I,J,K);  STMP2 = 1.0/RIG(I+1,J,K);  STMP4 = 1.0/RIG(I,J,K+1)
           STMP3 = STMP1 + STMP2
           RMAXY = 4.0/(STMP3 + 1.0/RIG(I,J+1,K) + 1.0/RIG(I+1,J+1,K))
           RMAXZ = 4.0/(STMP3 + STMP4 + 1.0/RIG(I+1,J,K+1))
           RMAYZ = 4.0/(STMP3 + STMP4 + 1.0/RIG(I,J+1,K+1))
!oat$ SplitPointCopyInsert
           SXY (I,J,K) = ( SXY (I,J,K) + (RMAXY*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG
           SXZ (I,J,K) = ( SXZ (I,J,K) + (RMAXZ*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG
           SYZ (I,J,K) = ( SYZ (I,J,K) + (RMAYZ*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG
      END DO;  END DO;  END DO
!$omp end parallel do
!oat$ install LoopFusionSplit region end
```

Automatic Generated Codes for the kernel 1

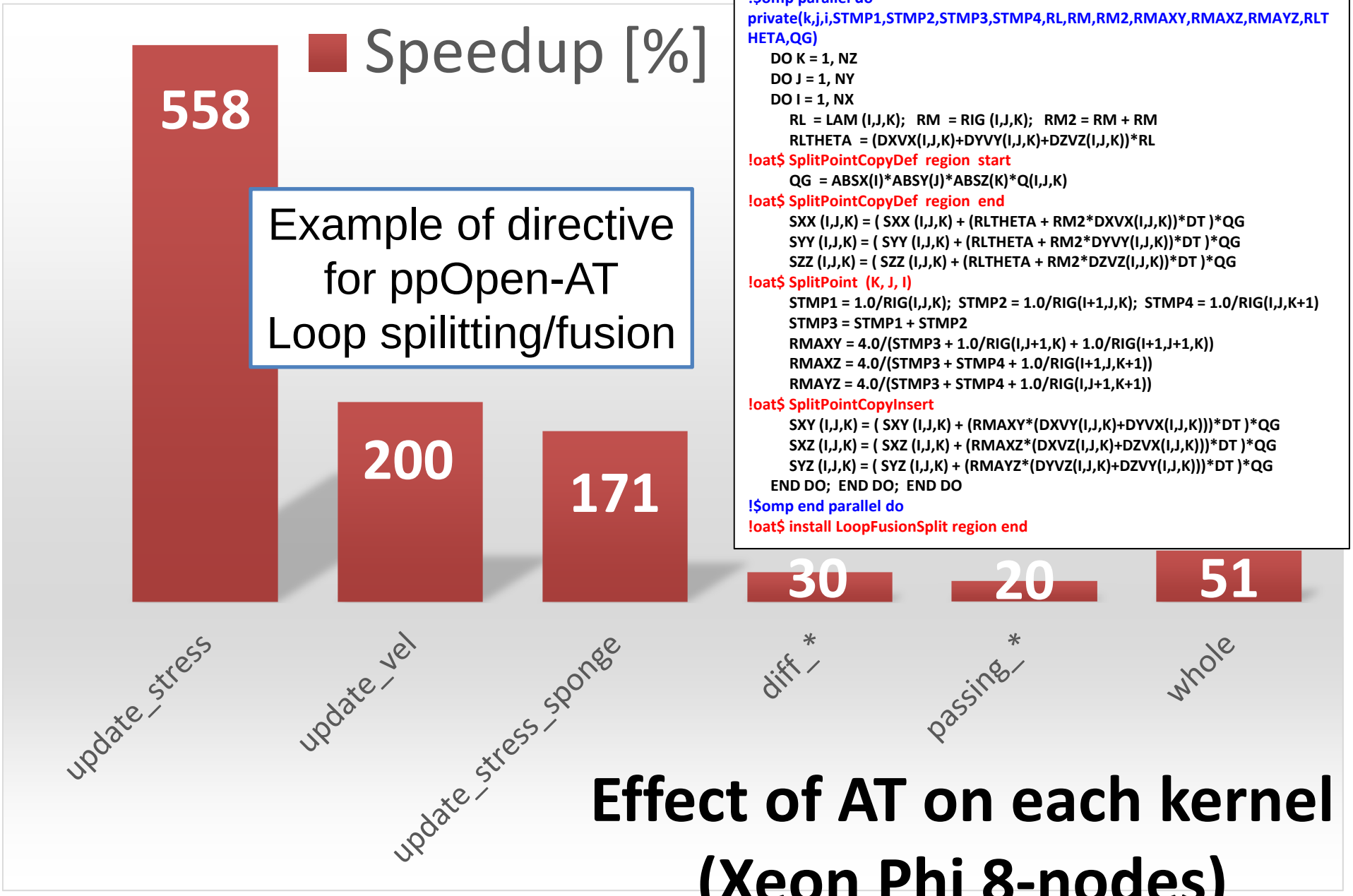
ppohFDM_update_stress

- #1 [Baseline]: Original 3-nested Loop
- #2 [Split]: Loop Splitting with K-loop
  (Separated, two 3-nested loops)
- #3 [Split]: Loop Splitting with J-loop
- #4 [Split]: Loop Splitting with I-loop
- #5 [Split&Fusion]: Loop Fusion to #1 for K and J-loops
  (2-nested loop)
- #6 [Split&Fusion]: Loop Fusion to #2 for K and J-Loops
  (2-nested loop)
- #7 [Fusion]: Loop Fusion to #1
  (loop collapse)
- #8 [Split&Fusion]: Loop Fusion to #2
  (loop collapse, two one-nest loop)

Automatic Generated Codes for the kernel 2

ppohFDM_update_vel

- #1 [Baseline]: Original 3-nested Loop.
- #2 [Fusion]: Loop Fusion for K and J-Loops. (2-nested loop)
- #3 [Fusion]: Loop Split for K, J, and I-Loops. (Loop Collapse)
- #4 [Fusion&Re-order]: Re-ordering of sentences to #1.
- #5 [Fusion&Re-order]: Re-ordering of sentences to #2.
- #6 [Fusion&Re-order]: Re-ordering of sentences to #3.



Example of directive for ppOpen-AT Loop spilitting/fusion

```
!oat$ install LoopFusionSplit region start
!$omp parallel do
private(k,j,i,STMP1,STMP2,STMP3,STMP4,RL,RM,RM2,RMAXY,RMAXZ,RMAYZ,RLT
HETA,QG)
    DO K = 1, NZ
    DO J = 1, NY
    DO I = 1, NX
       RL  = LAM (I,J,K);   RM  = RIG (I,J,K);   RM2 = RM + RM
       RLTHETA  = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL
!oat$ SplitPointCopyDef  region  start
       QG  = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
!oat$ SplitPointCopyDef  region  end
       SXX (I,J,K) = ( SXX (I,J,K) + (RLTHETA + RM2*DXVX(I,J,K))*DT )*QG
       SYY (I,J,K) = ( SYY (I,J,K) + (RLTHETA + RM2*DYVY(I,J,K))*DT )*QG
       SZZ (I,J,K) = ( SZZ (I,J,K) + (RLTHETA + RM2*DZVZ(I,J,K))*DT )*QG
!oat$ SplitPoint  (K, J, I)
       STMP1 = 1.0/RIG(I,J,K);  STMP2 = 1.0/RIG(I+1,J,K);  STMP4 = 1.0/RIG(I,J,K+1)
       STMP3 = STMP1 + STMP2
       RMAXY = 4.0/(STMP3 + 1.0/RIG(I,J+1,K) + 1.0/RIG(I+1,J+1,K))
       RMAXZ = 4.0/(STMP3 + STMP4 + 1.0/RIG(I+1,J,K+1))
       RMAYZ = 4.0/(STMP3 + STMP4 + 1.0/RIG(I,J+1,K+1))
!oat$ SplitPointCopyInsert
       SXY (I,J,K) = ( SXY (I,J,K) + (RMAXY*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG
       SXZ (I,J,K) = ( SXZ (I,J,K) + (RMAXZ*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG
       SYZ (I,J,K) = ( SYZ (I,J,K) + (RMAYZ*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG
    END DO;  END DO;  END DO
!$omp end parallel do
!oat$ install LoopFusionSplit region end
```

Effect of AT on each kernel (Xeon Phi 8-nodes)

Schedule of Public Release

(with English Documents, MIT License)

http://ppopenhpc.cc.u-tokyo.ac.jp/

- Released at SC-XY (or can be downloaded)
- Multicore/manycore cluster version (Flat MPI, OpenMP/MPI Hybrid) with documents in English
- We are now focusing on MIC/Xeon Phi
- Collaborations with scientists are welcome

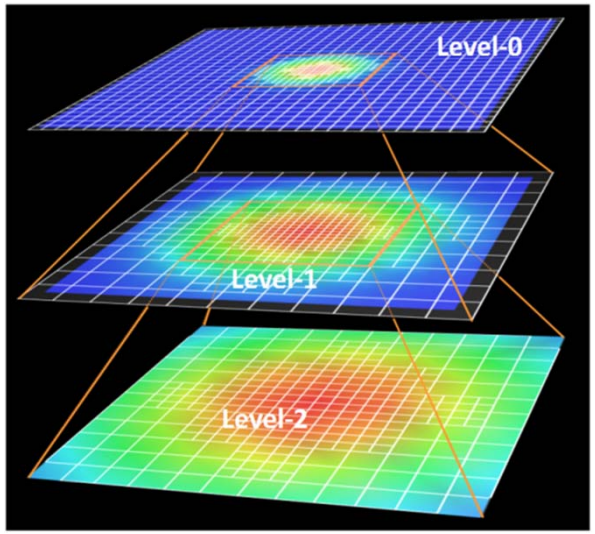
History

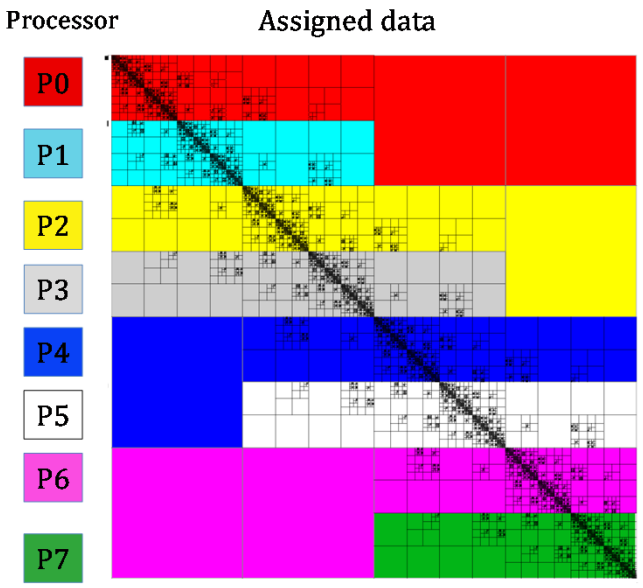
- SC12, Nov 2012 (Ver.0.1.0)
- SC13, Nov 2013 (Ver.0.2.0)
- SC14, Nov 2014 (Ver.0.3.0)

New Features in Ver.0.3.0

http://ppopenhpc.cc.u-tokyo.ac.jp/

- ppOpen-APPL/AMR-FDM: AMR framework with a dynamic load-balancing method for various FDM applications
- HACApK library for H-matrix comp. in ppOpen-APPL/BEM
  - Akihiro Ida (Kyoto U.)
- Utilities for pre-processing in ppOpen-APPL/DEM



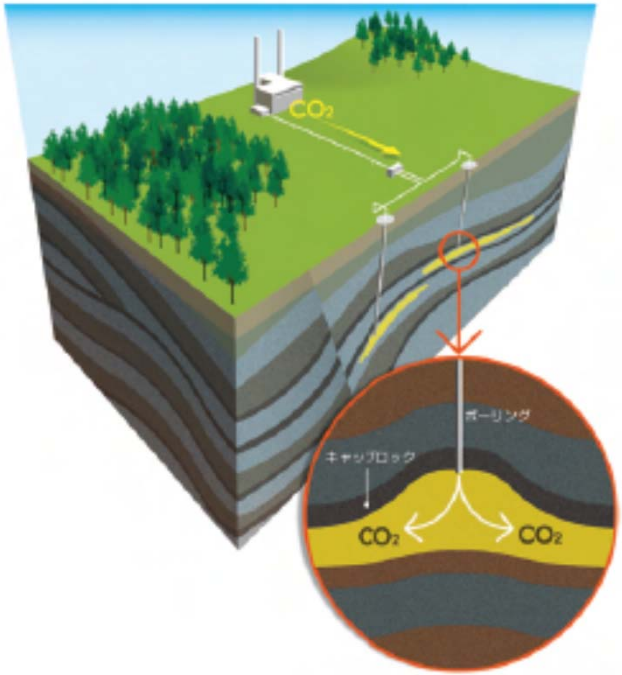


Assign a^c to P_k if $S(a^c{}_{ij})$ in $R(P_k)$

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 : $R(P_k) = \{i \mid l_k \leq i < l_{k+1}\}$
S : $S(a^c{}_{ij}) = I$ when $a^c{}_{ij} = A_{IJ}$

Collaborations, Outreaching

- Collaborations
  - International Collaborations
    - Lawrence Berkeley National Lab.
    - National Taiwan University
    - IPCC（Intel Parallel Computing Center）
- Outreaching, Applications
  - Large-Scale Simulations
    - Geologic CO_2 Storage
    - Astrophysics
    - Earthquake Simulations etc.
    - ppOpen-AT, ppOpen-MATH/VIS, ppOpen-MATH/MP, Linear Solvers
  - Intl. Workshops (2012, 2013)
  - Tutorials, Classes



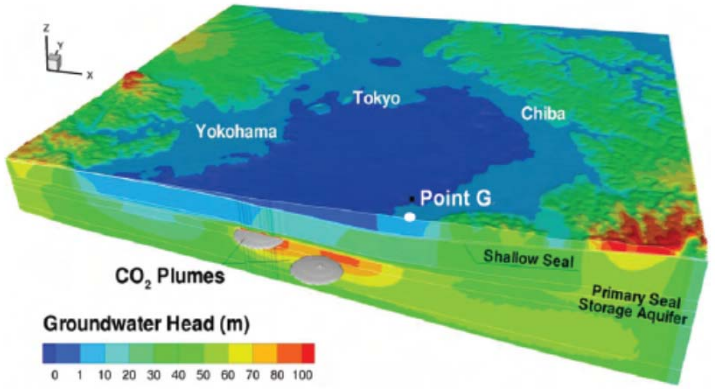


図-4 CO_2圧入後の地下水圧（全水頭換算）の分布（100年後）

ポストペタからエクサスケールへ

- ポストT2Kを念頭に置き，Intel Xeon/Phiなどメニィコアアーキテクチャ向け最適化を中心に研究開発を進める
- 「ポストペタスケール」から「エクサスケール」へ
  - エクサスケールシステムの詳細が具体的になりつつある
    - スーパーコンピュータシステムはより複雑化，巨大化
    - アプリケーション実行性能を引き出すためのプログラミングはより困難に
  - ppOpen-HPCのような環境の必要性はより高まる
  - エクサスケールシステム開発の動向も念頭に置いた開発を継続して実施して行くことで，エクサスケールシステムの利用に当たってもスムーズな移行を支援できると考えられる。
- エクサスケールシステムを念頭においた研究開発
  - 電力最適化
  - Communication/Synchronization Reducing Algorithms（通信・同期削減アルゴリズム）