



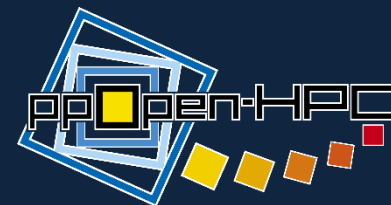
全国共同利用施設

東京大学情報基盤センター

Information Technology Center, The University of Tokyo



東京大学
THE UNIVERSITY OF TOKYO



ppOpen-HPCの概要

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

松本正晴, 片桐孝洋, 中島 研吾

東京大学情報基盤センター

第52回お試しアカウント付き並列プログラミング講習会

「ライブラリ利用: 高性能プログラミング初級入門」

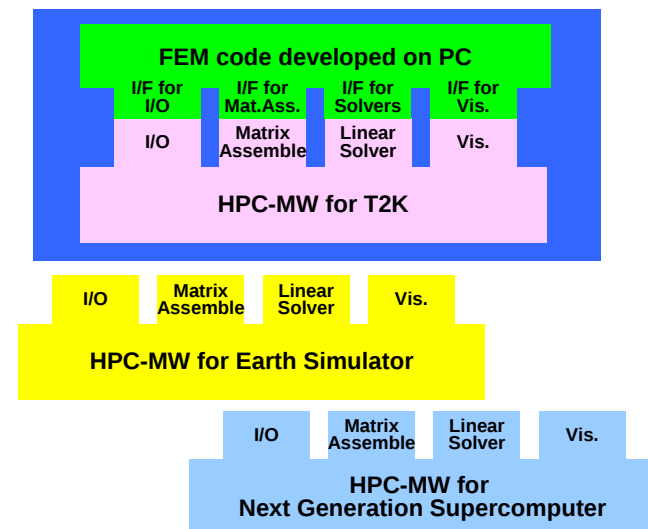
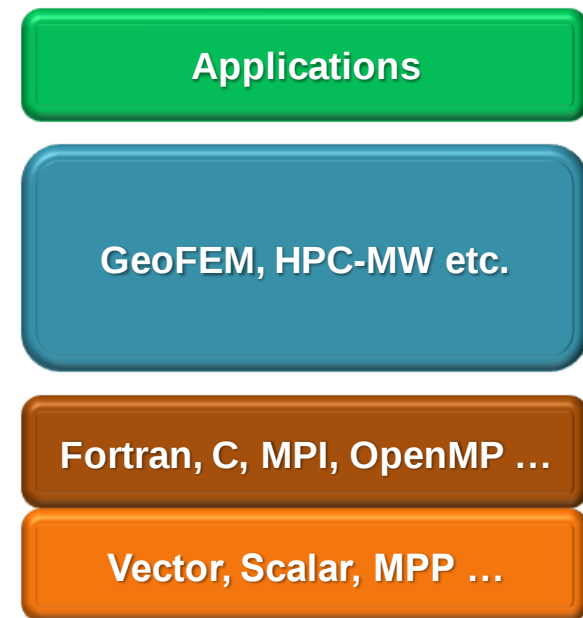
2016年2月3日～4日

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十分に引き出し, 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。
 - これら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できる。
 - アプリケーションMW, HPC-MW, フレームワーク

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

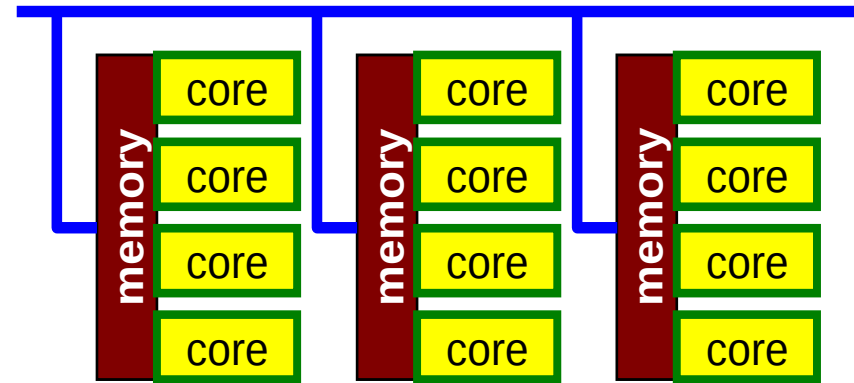


Hybrid並列プログラミングモデル

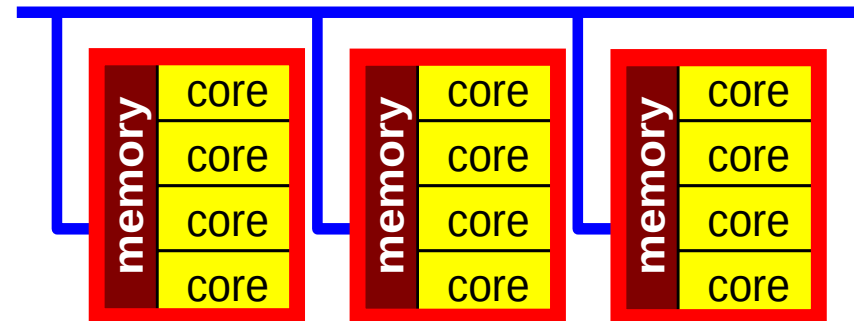
MPI+”X”

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

Flat MPI



OpenMP/MPI Hybrid



	Site	Computer/Year Vendor	Cores	R _{max}	R _{peak}	Power
1	National Supercomputer Center in Guangzhou, China	Tianhe-2A Intel Xeon E5-2692, Xeon Phi 31S1P, 2013 NUDT	3,120,000	33,863 (= 33.9 PF)	54,902	17,808
2	Oak Ridge National Laboratory, USA	Titan Cray XK7/NVIDIA K20x, 2012 Cray	560,640	17,590	27,113	8,209
3	Lawrence Livermore National Laboratory, USA	Sequoia BlueGene/Q, 2011 IBM	1,572,864	17,173	20,133	7,890
4	RIKEN AICS, Japan	K computer , SPARC64 VIIIfx , 2011 Fujitsu	705,024	10,510	11,280	12,660
5	Argonne National Laboratory, USA	Mira BlueGene/Q, 2012 IBM	786,432	8,586	10,066	3,945
6	DOE/NNSA/LANL/SNL, USA	Trinity Cray XC40, Xeon E5-2698v3, 2015 Cray	301,056	8,101	11,079	
7	Swiss National Supercomputing Ctr. (CSCS), Switzerland	Piz Daint Cray XC30, Xeon E5-2670 8C, NVIDIA K20x, 2013 Cray	115,984	6,271	7,789	2,325
8	HLRS, Germany	Hazel Hen Cray XC40, Xeon E5-2680v3, 2015 Cray	185,088	5,640	7,404	
9	KAUST, Saudi Arabia	Shaheen II Cray XC40, Xeon E5-2698v3 2015 Cray	196,608	5,537	7,235	2,834
10	TACC, USA	Stampede Xeon E5-2680/Xeon Phi, 2012 Dell	462,462	5,168	8,520	4,510
75	ITC/U. Tokyo, Japan	Oakleaf-FX SPARC64 IXfx, 2012 Fujitsu	76800	1,043	1,135	1,177

TOP500 List Highlights, Nov. 2015

- ✓ 2015年11月現在, Top500に掲載されたシステムのうち, 104システム(2015年6月: 90システム)はGPUかMICを搭載(66: NVIDIA, 3: ATI-Radeon, 27: Intel Xeon Phi, 4: Intel/NVIDIA combined)
- ✓ 1位にランクインした中国のTianhe-2(33.86 petaFLOPS)はXeon Ivy Bridge(CPU)とXeon Phi(MIC)のコンバインドシステムで計3,120,000コア。
- ✓ 1位のTianhe-2と10位のStanpedeはintel Xeon Phi、2位のTitanと7位のPiz DaintはNVIDIA GPUを搭載。

東大情報基盤センターのスパコン

(retired, March 2014)

Oakleaf-fx (Fujitsu PRIMEHPC FX10)

Total Peak performance : 1.13 PFLOPS
Total number of nodes : 4800
Total memory : 150 TB
Peak performance / node : 236.5 GFLOPS
Main memory per node : 32 GB
Disk capacity : 1.1 PB + 2.1 PB
SPARC64 lxfx 1.84GHz

T2K-Todai (Hitachi HA8000-tc/RS425)

Total Peak performance : 140 TFLOPS
Total number of nodes : 952
Total memory : 32000 GB
Peak performance / node : 147.2 GFLOPS
Main memory per node : 32 GB, 128 GB
Disk capacity : 1 PB
AMD Quad Core Opteron 2.3GHz

Yayoi (Hitachi SR16000/M1)

Total Peak performance : 54.9 TFLOPS
Total number of nodes : 56
Total memory : 11200 GB
Peak performance / node : 980.48 GFLOPS
Main memory per node : 200 GB
Disk capacity : 556 TB
IBM POWER 7 3.83GHz



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

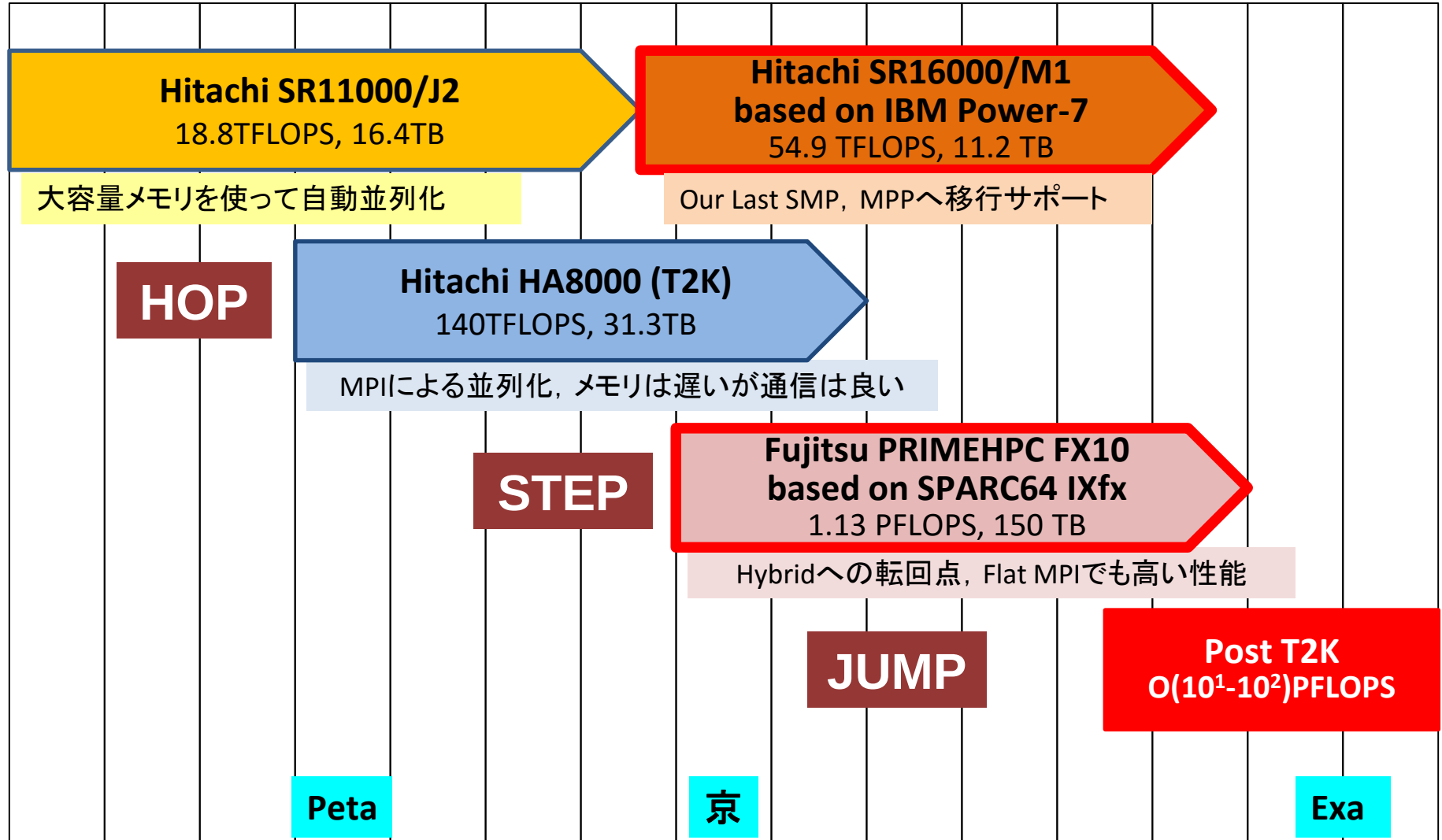


Total Users > 2,000

東大情報基盤センターのスパコン

FY

05 06 07 08 09 10 11 12 13 14 15 16 17 18 19



Post T2K System

- 25+ PFLOPS, 2016年度
- Many-core based (e.g. (only) Intel MIC/Xeon Phi)
- Joint Center for Advanced High Performance Computing (最先端共同HPC基盤施設, JCAHPC, <http://jcahpc.jp/>)
 - 筑波大学計算科学研究センター, 東京大学情報基盤センター
- Intelコンパイラが動くもののプログラミングは困難
 - MPI+OpenMP
 - 複雑なチューニングが必須
 - ユーザーを支援するためのフレームワークが必要



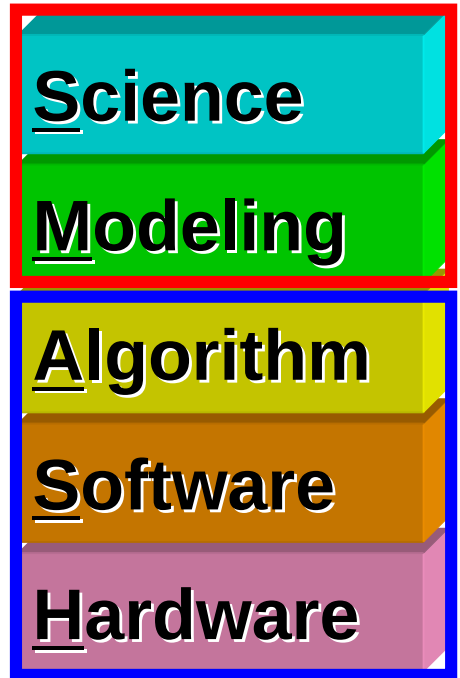
Key-Issues for Appl's/Algorithms towards Post-Peta & Exa Computing

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

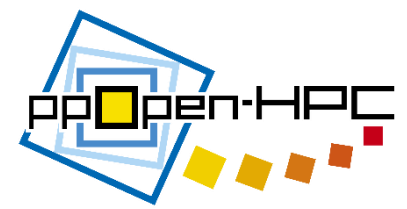
HPCミドルウェア：何がうれしいか

- 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
 - SMASHの探求に専念
 - 一生SMASHと付き合うのはきつい
 - SMASH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



HPCのCo-Design

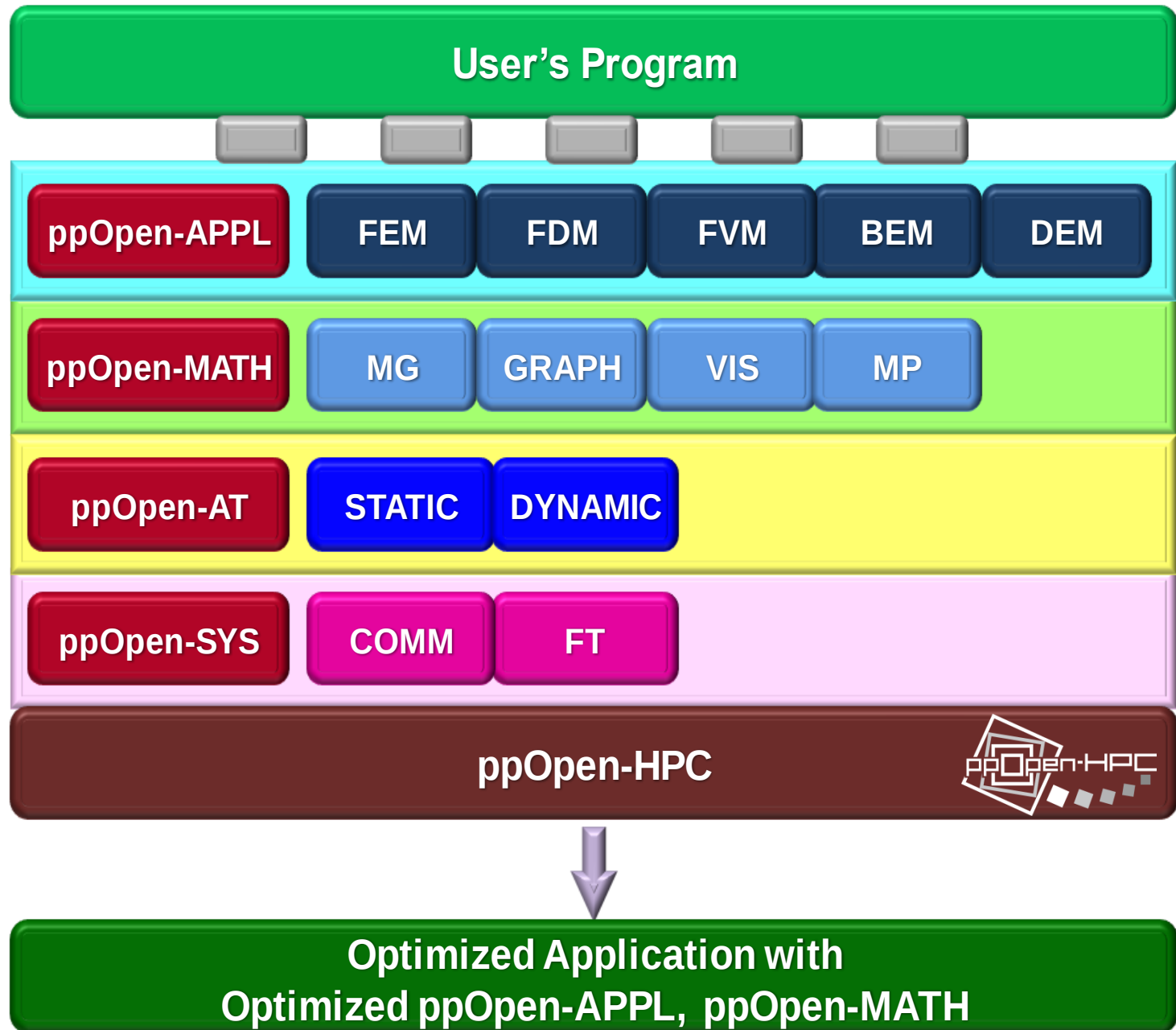
ppOpen-HPC



- 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十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」を開発中。
 - 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)(領域統括：佐藤三久(理化学研究所計算科学研究機構))
 - 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

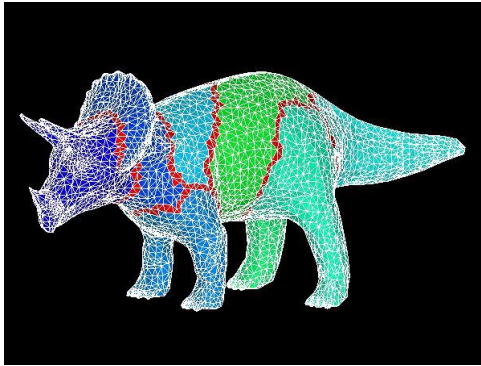
概要(1/3)

- 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28年度導入予定の25+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

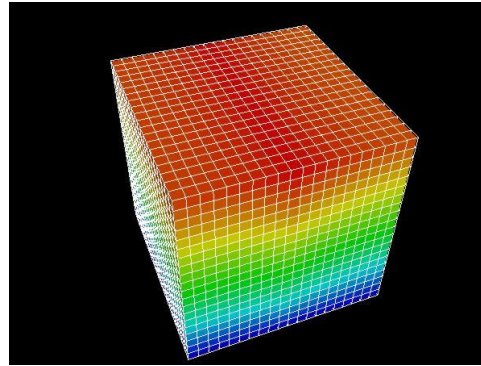


対象とする離散化手法

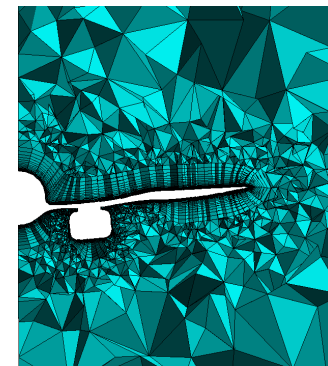
局所的, 隣接通信中心, 疎行列



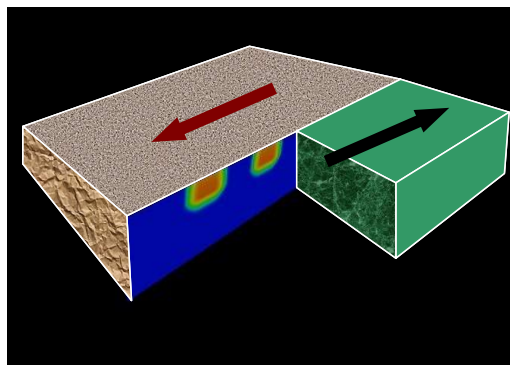
有限要素法
Finite Element Method
FEM



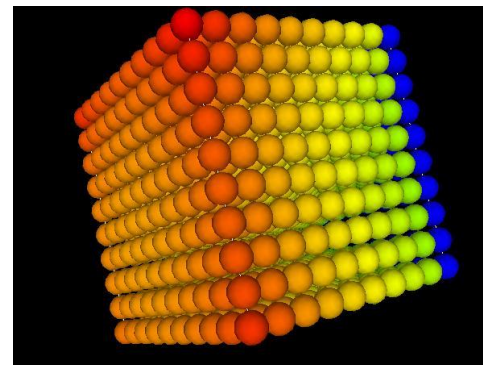
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



境界要素法
Boundary Element Method
BEM



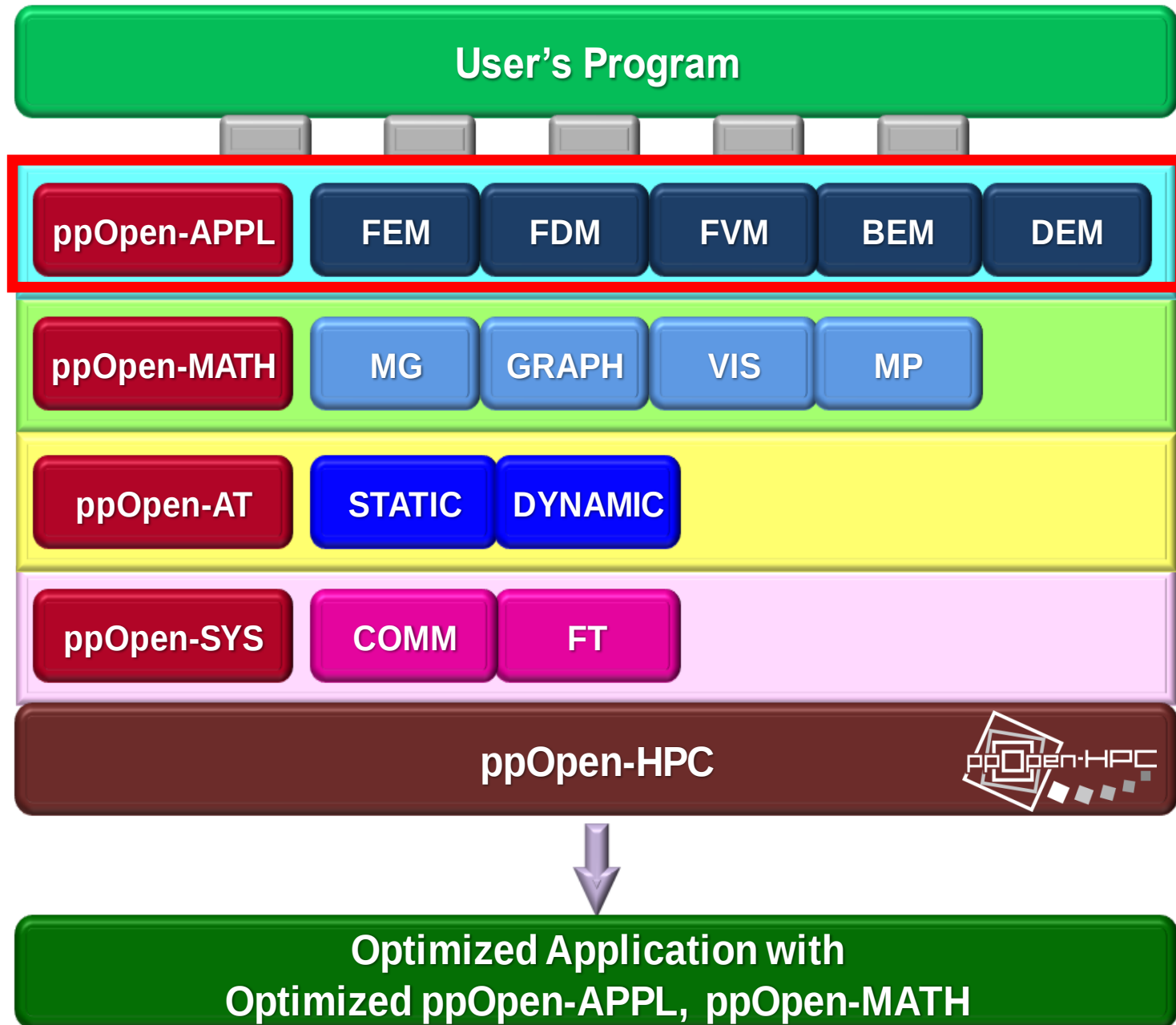
個別要素法
Discrete Element Method
DEM

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
 - 共通データ入出カインタフェース, 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月にVer.0.2.0公開
 - 平成26年11月にVer.0.3.0公開
 - 平成27年11月にVer.1.0.0公開
- 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



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
```

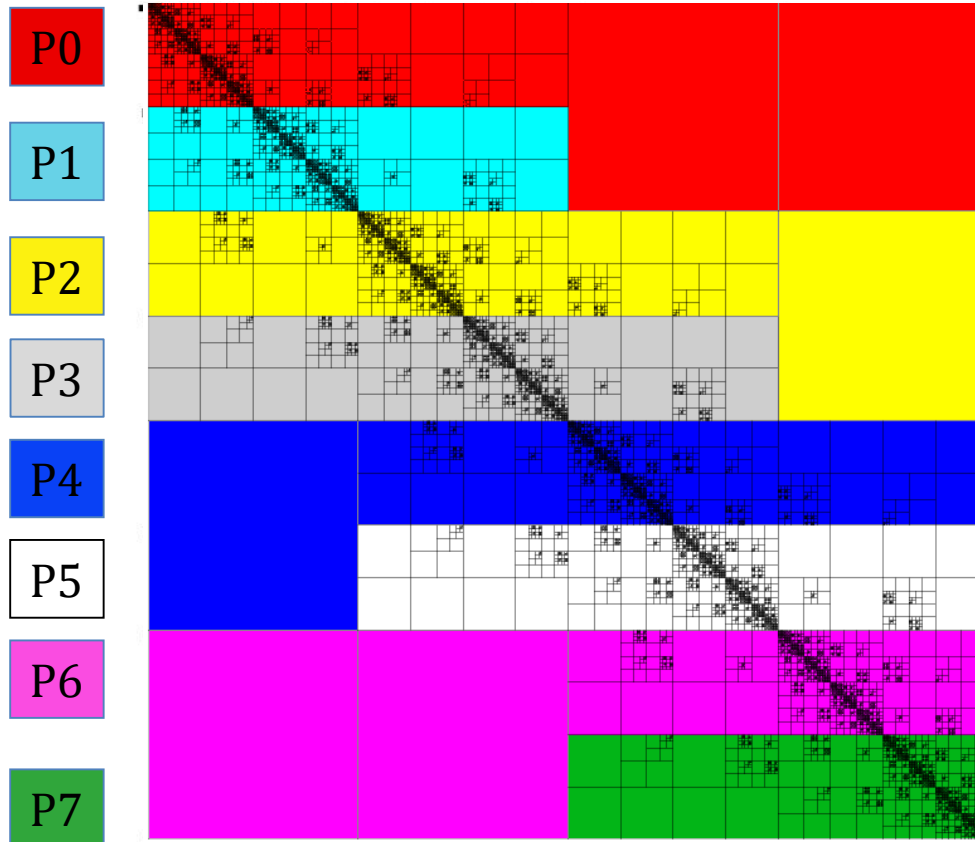
ppOpen-APPL

- 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

Processor

Assigned data



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

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

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

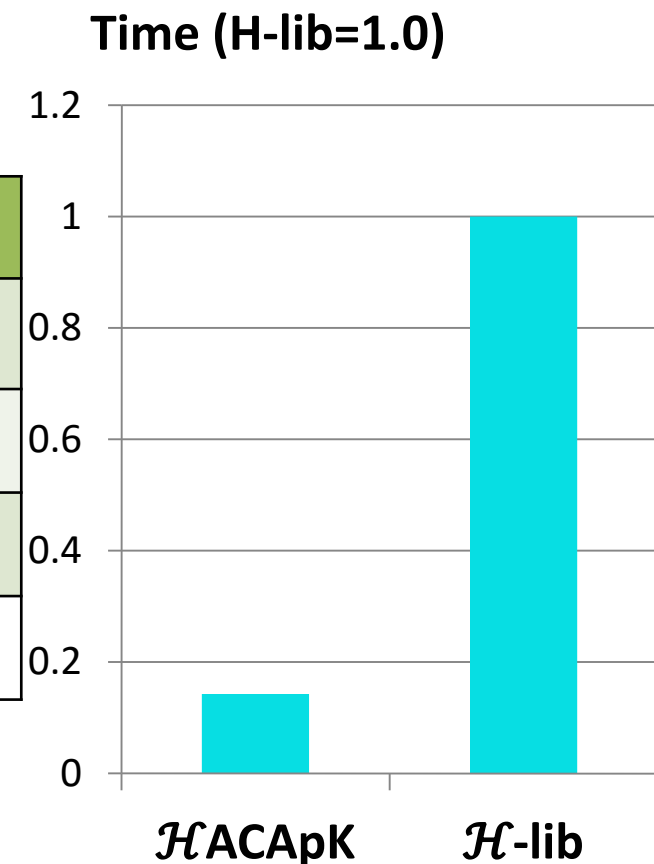
$S : S(a^c_{ij}) = I$ when $a^c_{ij} = A_{IJ}$

Calculation time of $\mathcal{H}$ ACApK

■ 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
 - Elastic wave propagation
 - 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)

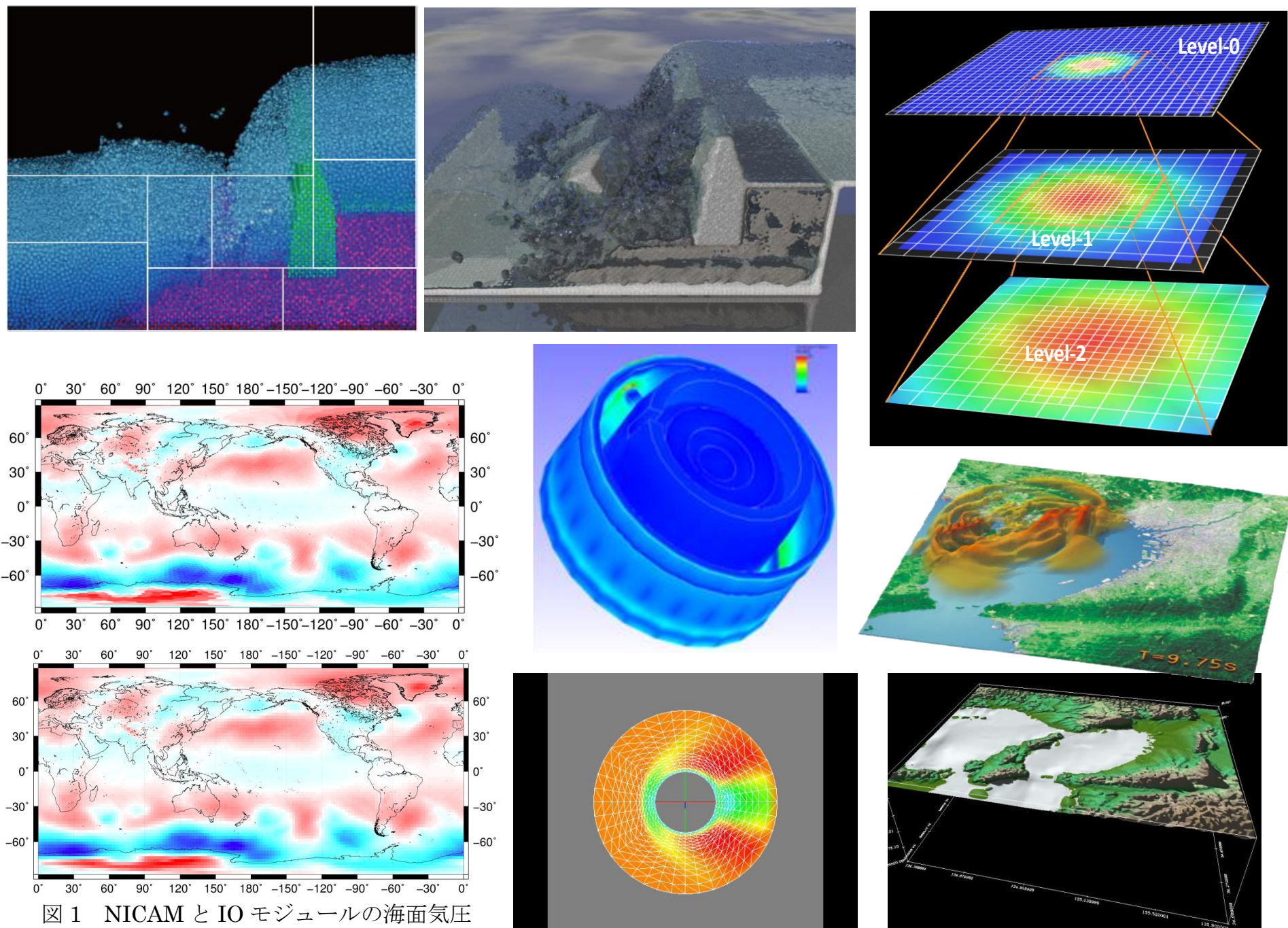
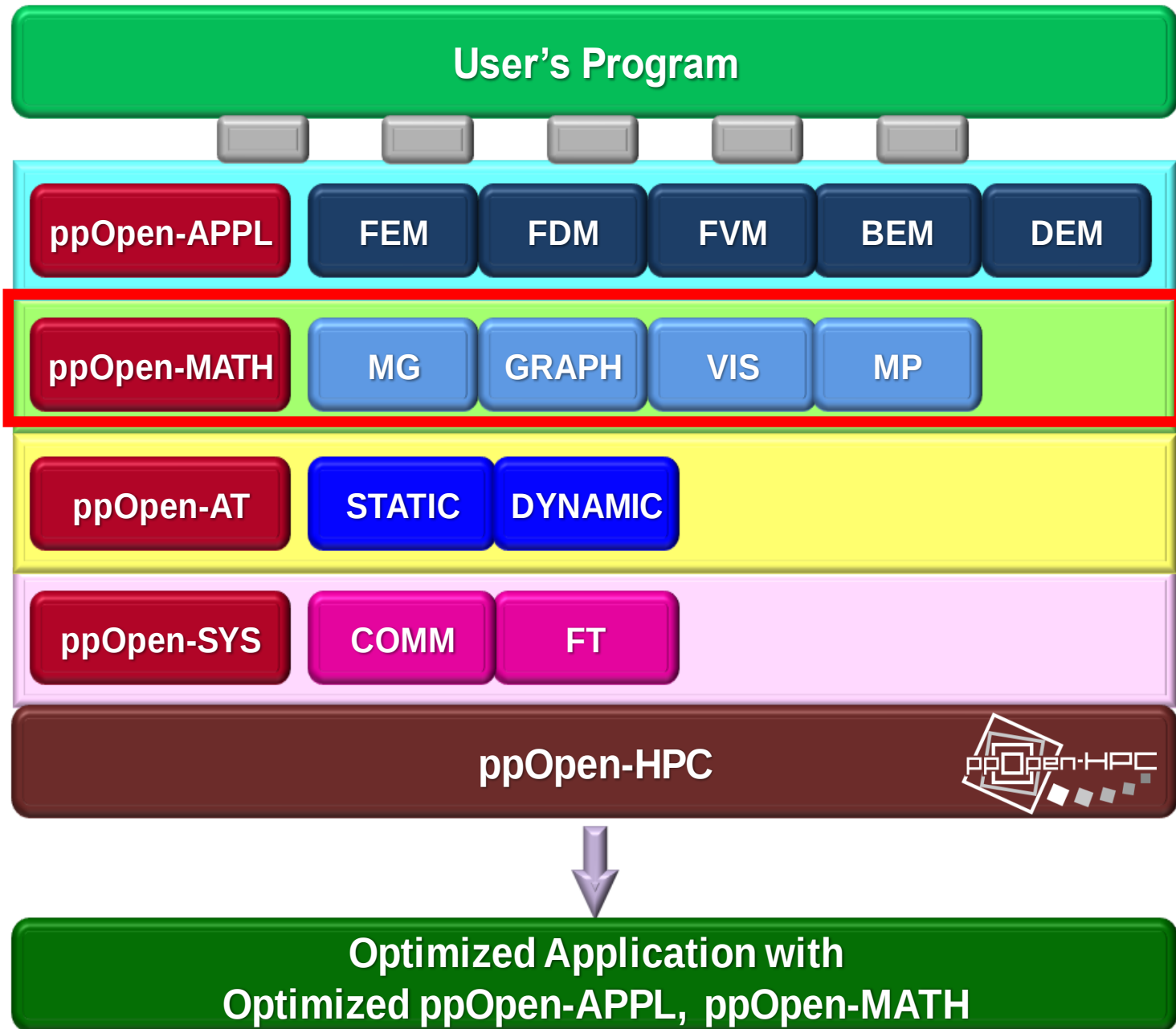
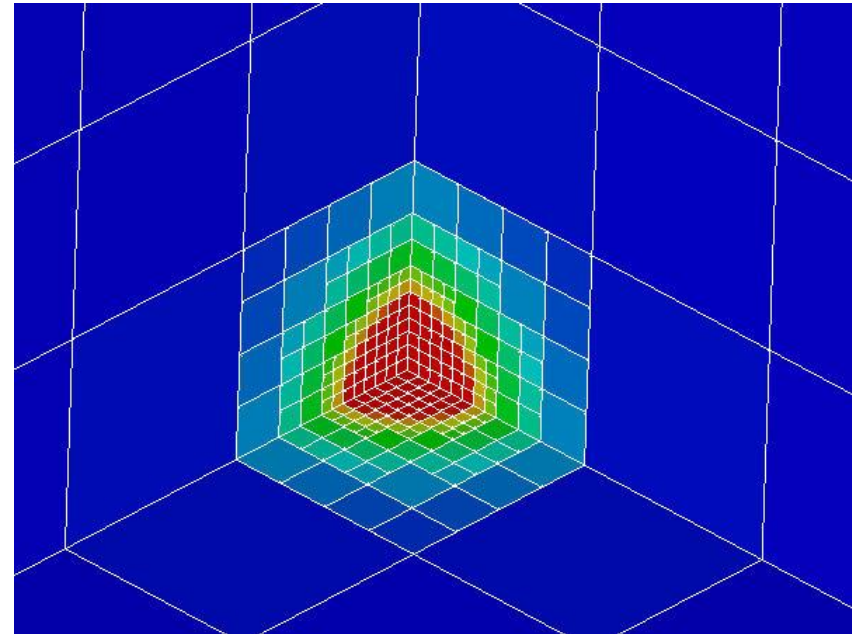


図1 NICAM と IO モジュールの海面気圧



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

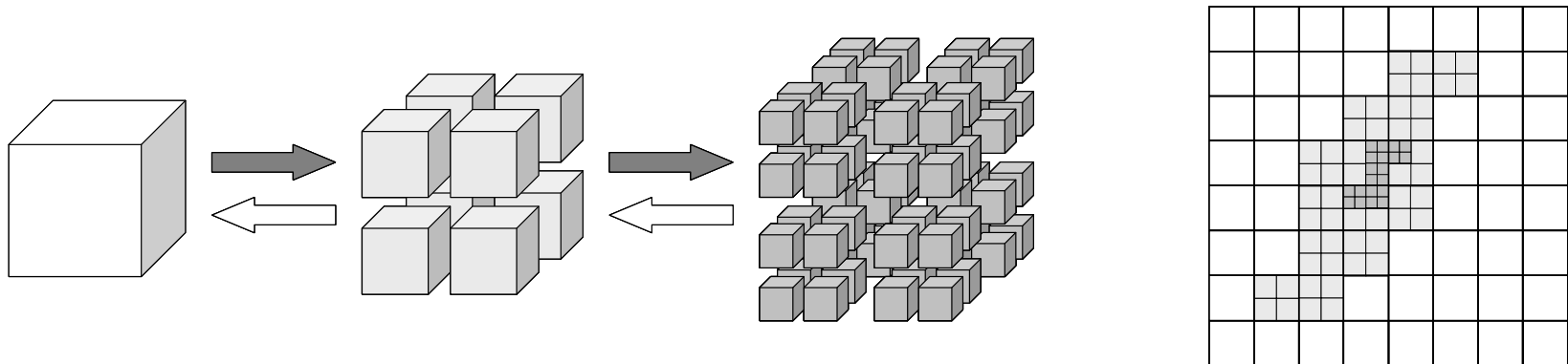
MaxRefineLevel = 20

Maximum number of refinement levels

Simplified Parallel Visualization using Background Voxels

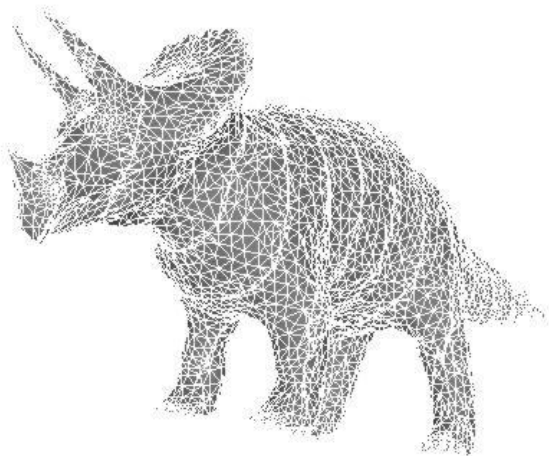
[KN, Chen 2006]

- 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 controlled, a single file with 10^5 meshes is possible, even though entire problem size is 10^9 with distributed data sets.

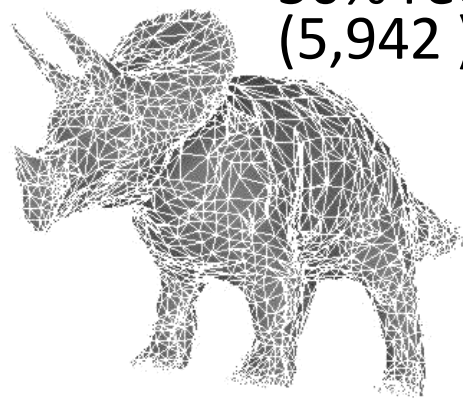


Example of Surface Simplification

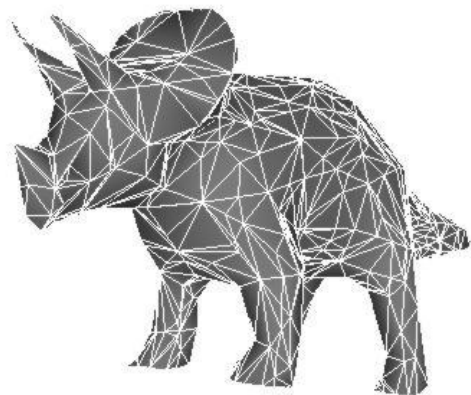
Initial
(11,884 tri's)



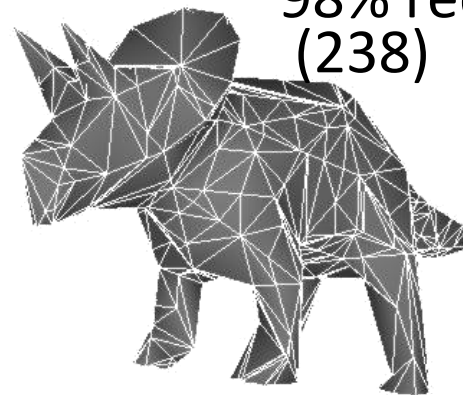
50% reduction
(5,942)



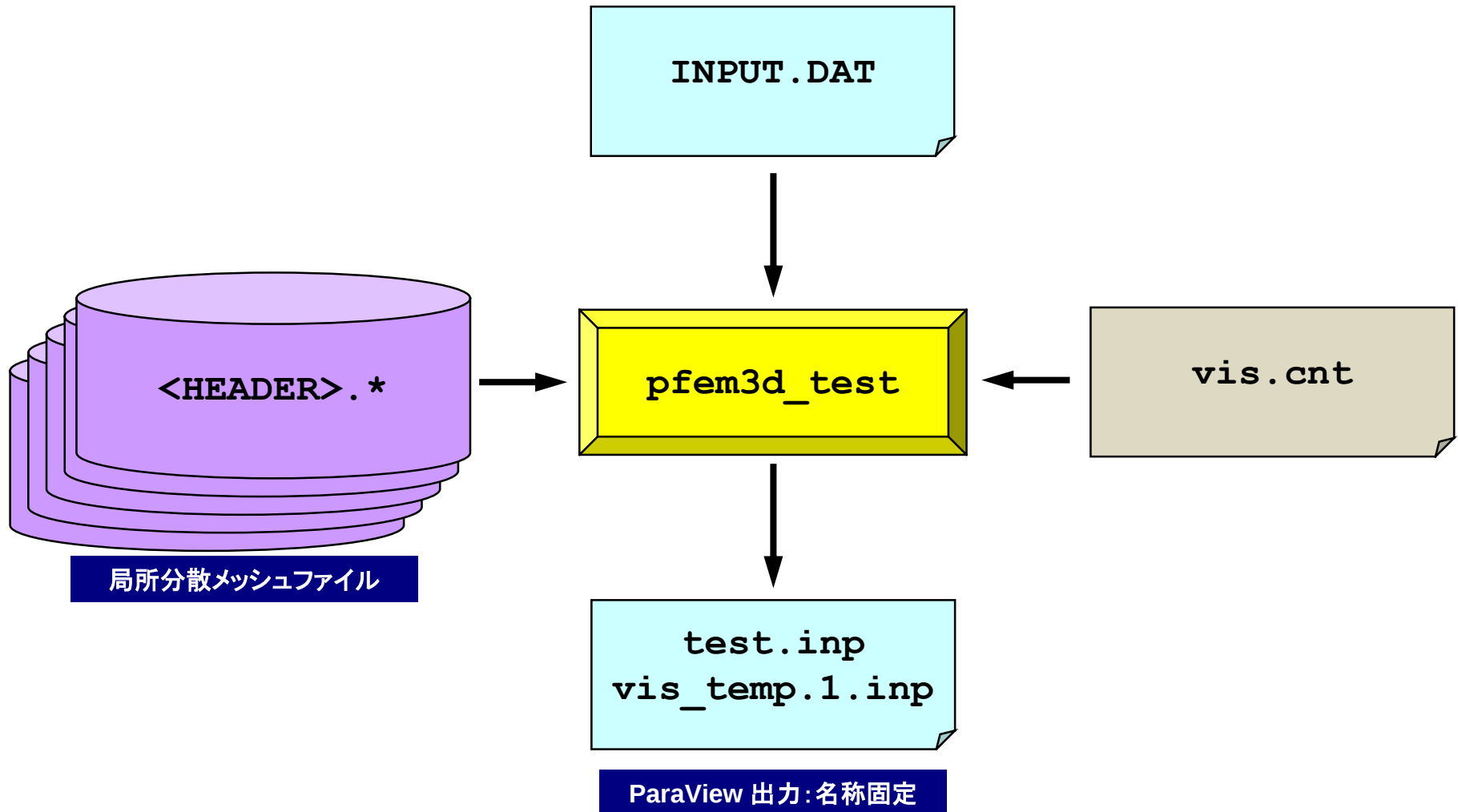
95% reduction
(594)



98% reduction
(238)



pFEM3D + ppOpen-MATH/VIS



Fortran/main (1/2)

```

use solver11
use pfem_util
use ppohvis_fdm3d_util

implicit REAL*8(A-H,O-Z)
type(ppohVIS_FDM3D_stControl)           :: pControl
type(ppohVIS_FDM3D_stResultCollection)  :: pNodeResult
type(ppohVIS_FDM3D_stResultCollection)  :: pElemResult
character(len=PPOHVIS_FDM3D_FILE_NAME_LEN) :: CtrlName
character(len=PPOHVIS_FDM3D_FILE_NAME_LEN) :: VisName
character(len=PPOHVIS_FDM3D_LABEL_LEN)    :: ValLabel
integer(kind=4)                          :: iErr

CtrlName = ""
CtrlName = "vis.cnt"

VisName = ""
VisName = "vis"

ValLabel = ""
ValLabel = "temp"

call PFEM_INIT

call ppohVIS_PFEM3D_Init(MPI_COMM_WORLD, iErr)
call ppohVIS_PFEM3D_GetControl(CtrlName, pControl, iErr);
call INPUT_CNTL
call INPUT_GRID

call ppohVIS_PFEM3D_SETMESHEX(
&      NP,      N,      NODE_ID, XYZ,      &
&      ICELTOT, ICELTOT_INT, ELEM_ID, ICELNOD, &
&      NEIBPETOT, NEIBPE, IMPORT_INDEX, IMPORT_ITEM, &
&      EXPORT_INDEX, EXPORT_ITEM, iErr)

```

Fortran/main (2/2)

```
call MAT_ASS_MAIN
call MAT_ASS_BC

call SOLVE11

call OUTPUT_UCD

call ppohVIS_PFEM3D_ConvResult(N, ValLabel, X, &
&                               pNodeResult, pElemResult, iErr)
call ppohVIS_PFEM3D_Visualize(pNodeResult, pElemResult, pControl, &
&                             VisName, 1, iErr)

call ppohVIS_PFEM3D_Finalize(iErr)

call PFEM_FINALIZE

end program heat3Dp
```

C/main (1/2)

```

#include <stdio.h>
#include <stdlib.h>
FILE* fp_log;
#define GLOBAL_VALUE_DEFINE
#include "pfem_util.h"
#include "ppohVIS_FDM3D_Util.h"
extern void PFEM_INIT(int, char**);
extern void INPUT_CNTL();
extern void INPUT_GRID();
extern void MAT_CON0();
extern void MAT_CON1();
extern void MAT_ASS_MAIN();
extern void MAT_ASS_BC();
extern void SOLVE11();
extern void OUTPUT_UCD();
extern void PFEM_FINALIZE();
int main(int argc, char* argv[])
{
    double START_TIME, END_TIME;
    struct ppohVIS_FDM3D_stControl *pControl = NULL;
    struct ppohVIS_FDM3D_stResultCollection *pNodeResult = NULL;

    PFEM_INIT(argc, argv);

    ppohVIS PFEM3D_Init(MPI_COMM_WORLD);
    pControl = ppohVIS_FDM3D_GetControl("vis.cnt");

    INPUT_CNTL();
    INPUT_GRID();

    if(ppohVIS PFEM3D_SetMeshEx(
        NP, N, NODE_ID, XYZ,
        ICELTOT, ICELTOT_INT, ELEM_ID, ICELNOD,
        NEIBPETOT, NEIBPE, IMPORT_INDEX, IMPORT_ITEM, EXPORT_INDEX, EXPORT_ITEM)) {
        ppohVIS_FDM3D_PrintError(stderr);
    };
};

```

C/main (2/2)

```
MAT_CON0();  
MAT_CON1();  
  
MAT_ASS_MAIN();  
MAT_ASS_BC();  
  
SOLVE11();  
  
OUTPUT_UCD();  
  
pNodeResult = ppohVIS_PFEM3D_ConvResult(N, "temp", X);  
  
if(ppohVIS_PFEM3D_Visualize(pNodeResult, NULL, pControl, "vis", 1)) {  
    ppohVIS_FDM3D_PrintError(stderr);  
}  
  
ppohVIS_PFEM3D_Finalize();  
  
PFEM_FINALIZE();  
}
```

vis.cnt

[Refine]

AvailableMemory = 2.0

MaxVoxelCount = 1000

MaxRefineLevel = 20

[Simple]

ReductionRate = 0.0

細分化制御情報セクション

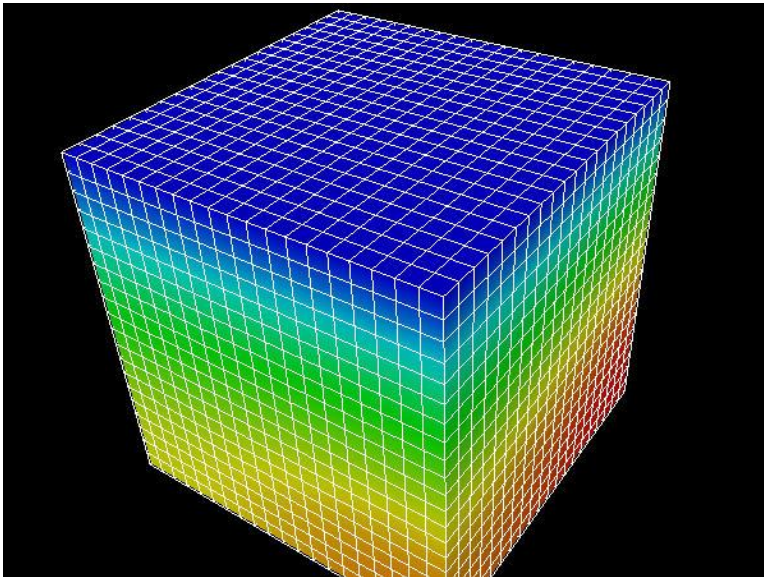
利用可能メモリ容量 (GB) not in use

Max Voxel #

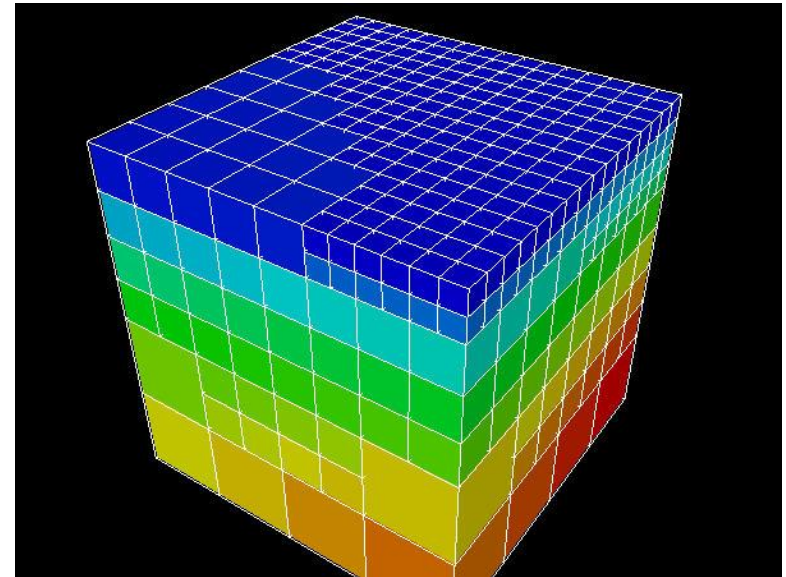
Max Voxel Refinement Level

簡素化制御情報セクション

表面パッチ削減率



1.52 MB
8,000 elements, 10,334 nodes



.385 MB,
813 elements, 6,504 nodes

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
 - H27にライブラリ初公開
 - 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
- 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。
- 構造物モデルと地震モデルの結合など，他分野への応用。

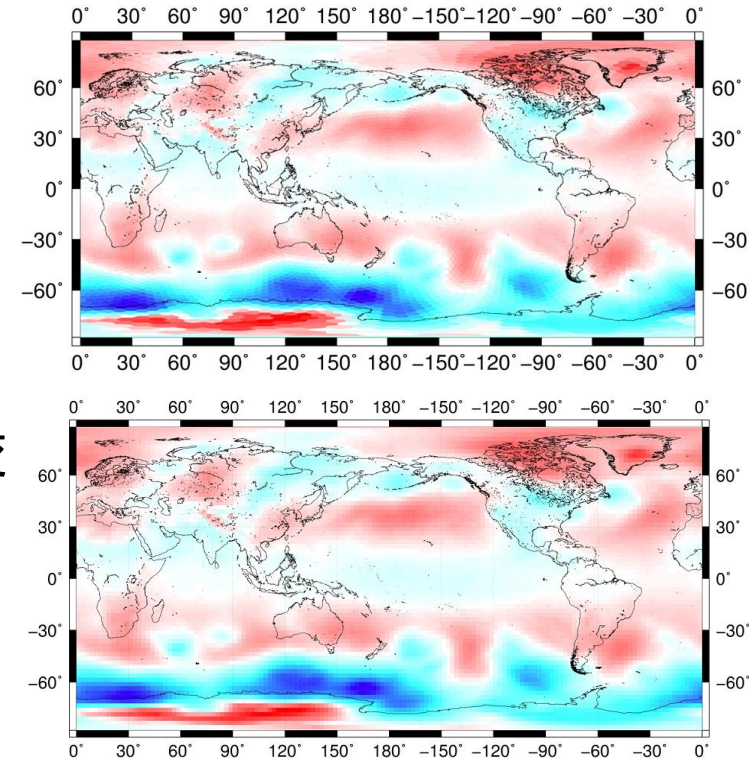
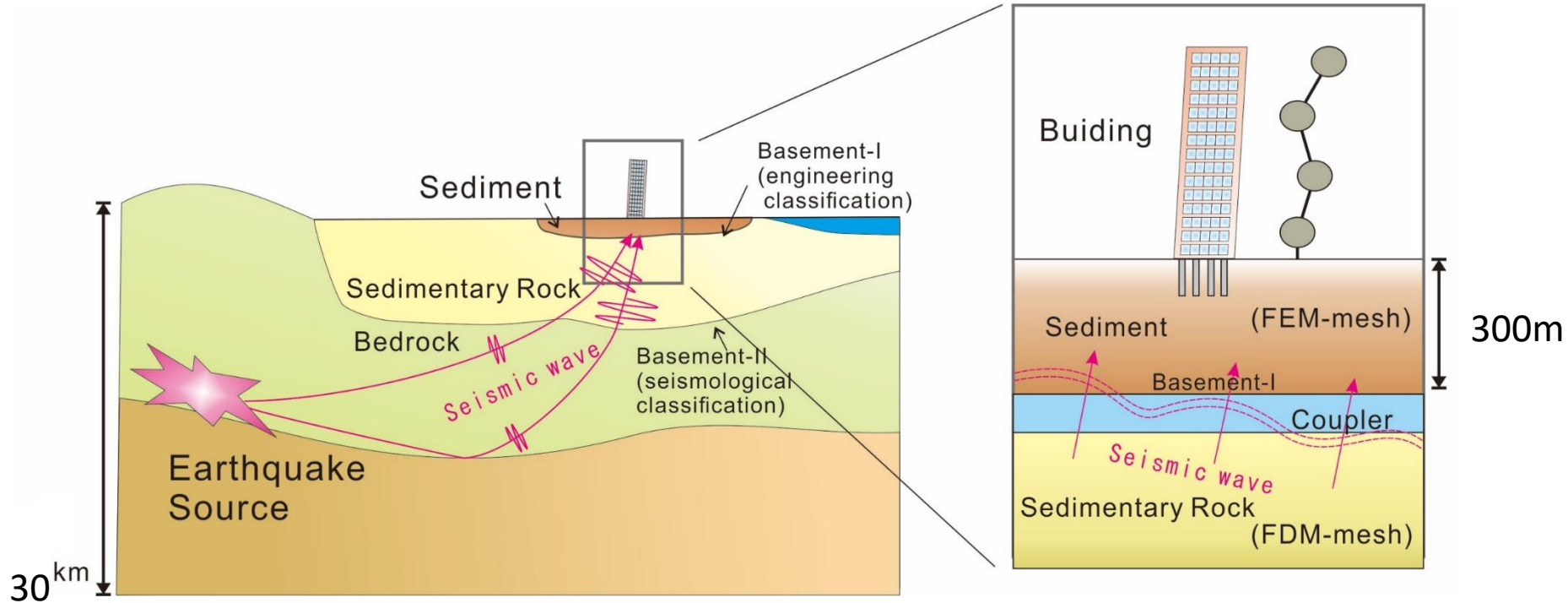


図1 NICAM と IO モジュールの海面気圧

ppOpen-MATH/MPによる地震波動－建築物振動 連成シミュレーション例



地震波動の伝播(FDM)と、それに伴う建築物の振動(FEM)について、FDM→FEMの片側連成で解析を行う。

利用するアプリケーションについて

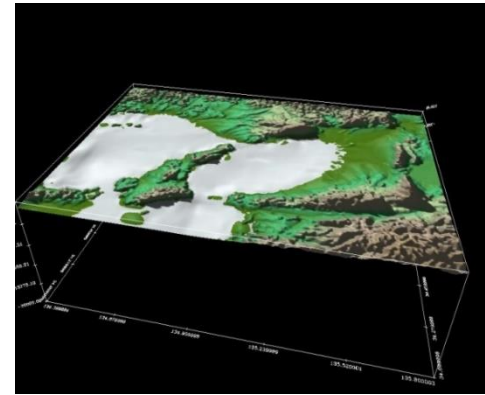
Seism3D+ (composed by ppOpen-APPL/FDM)

広域の地震波動伝播を解析するための陽解法FDMアプリ

$$\rho \frac{\partial v_p}{\partial t} = \left(\frac{\partial \sigma_{xp}}{\partial x} + \frac{\partial \sigma_{yp}}{\partial y} + \frac{\partial \sigma_{zp}}{\partial z} + f_p \right), \quad (p = x, y, z)$$

$$\frac{\partial \sigma_{pq}}{\partial t} = \lambda \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right) \delta_{pq} + \mu \left(\frac{\partial v_p}{\partial q} + \frac{\partial v_q}{\partial p} \right), \quad (p, q = x, y, z)$$

v : 速度
 σ : 応力
 f : 外力
 λ, μ : Lamé定数



FDM計算領域

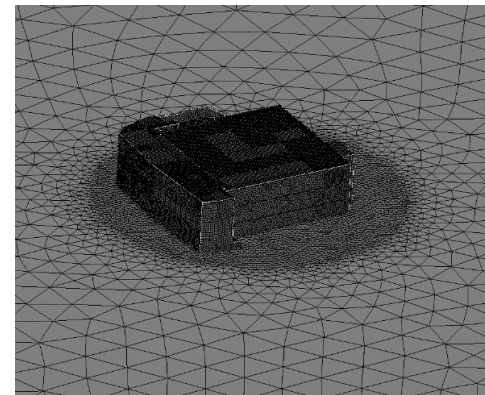
FrontISTR++ (composed by ppOpen-APPL/FEM)

構造解析用陰解法FEMアプリ

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}$$

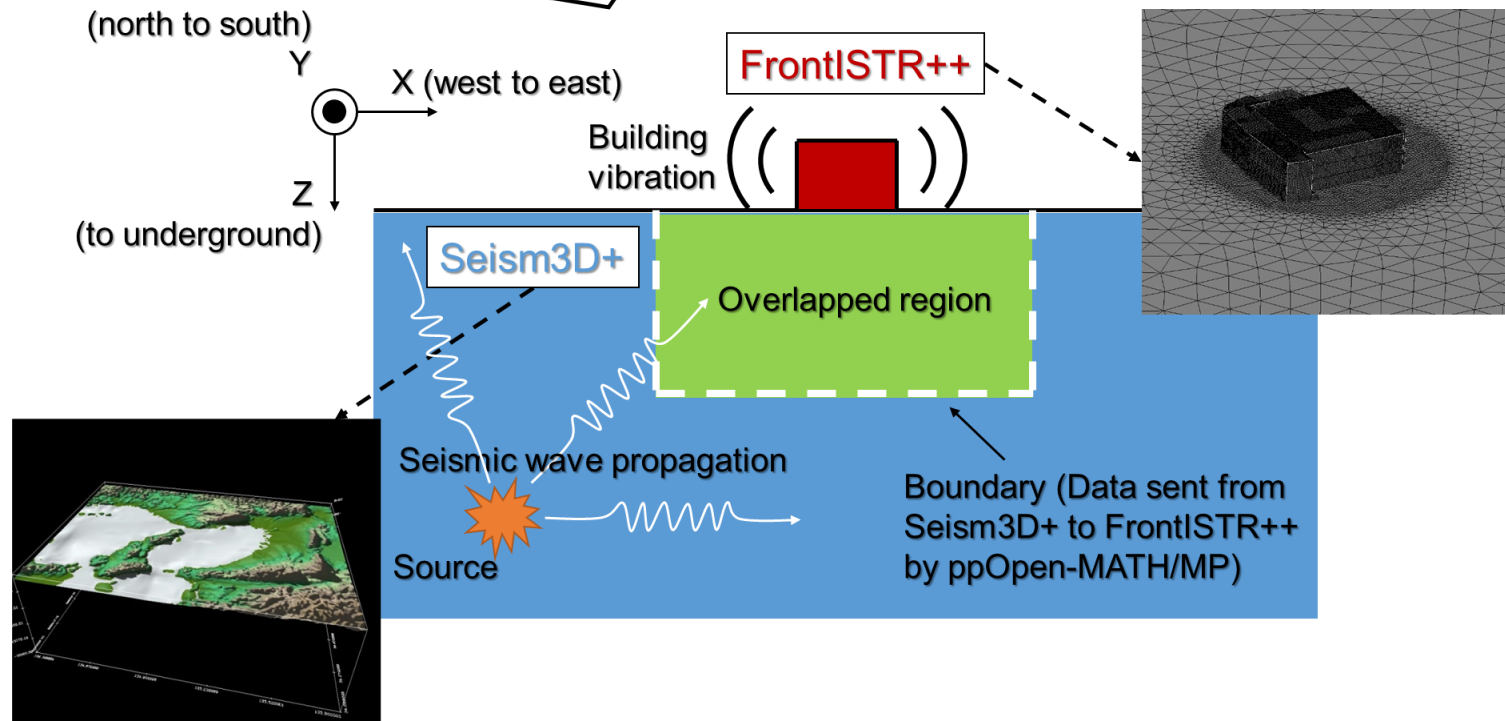
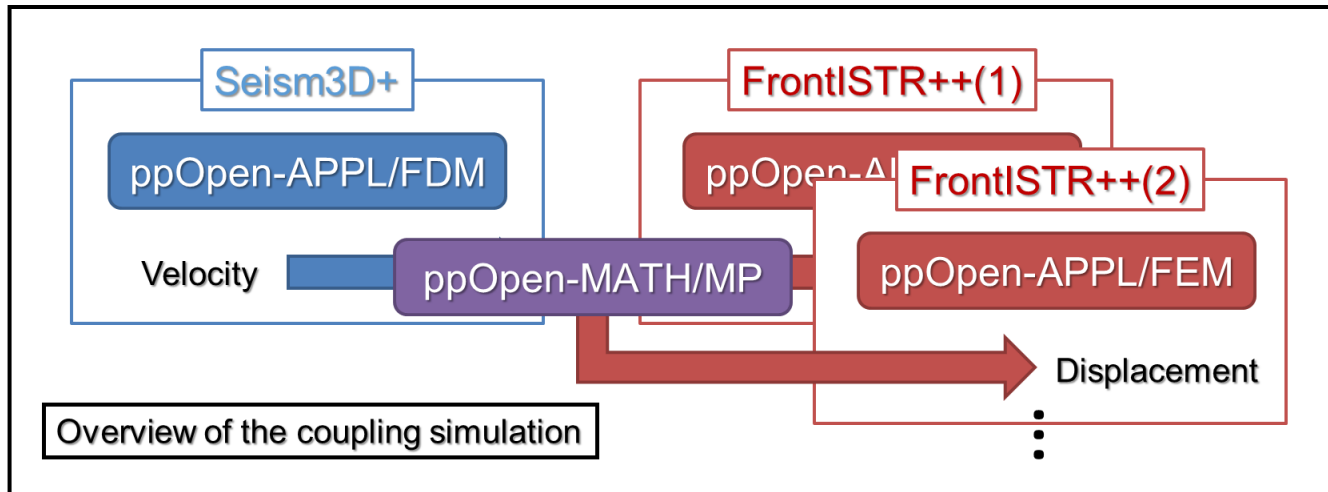
$\mathbf{M}$: 質量行列
 $\mathbf{C}$: 減衰行列
 $\mathbf{K}$: 剛性行列

$\mathbf{F}$: 節点荷重ベクトル
 $\mathbf{d}$: 節点変位ベクトル



FEM建物メッシュ

連成シミュレーションの実装



Oakleaf-FXによる大規模HPCチャレンジの計算条件

阪神淡路地域を対象とした約60km四方のエリアで、震源は2013年（平成25年）4月13日に発生した淡路島地震を再現。構造物は神戸市ポートアイランドにある理化学研究所計算科学研究機構（AICS）の建屋をモデリング。

Seism3D+

全格子点数 $(x, y, z) = (1536, 1536, 1600)$
 並列数 2560計算ノード: 2560プロセス
 16スレッド/1プロセス

FrontISTR++

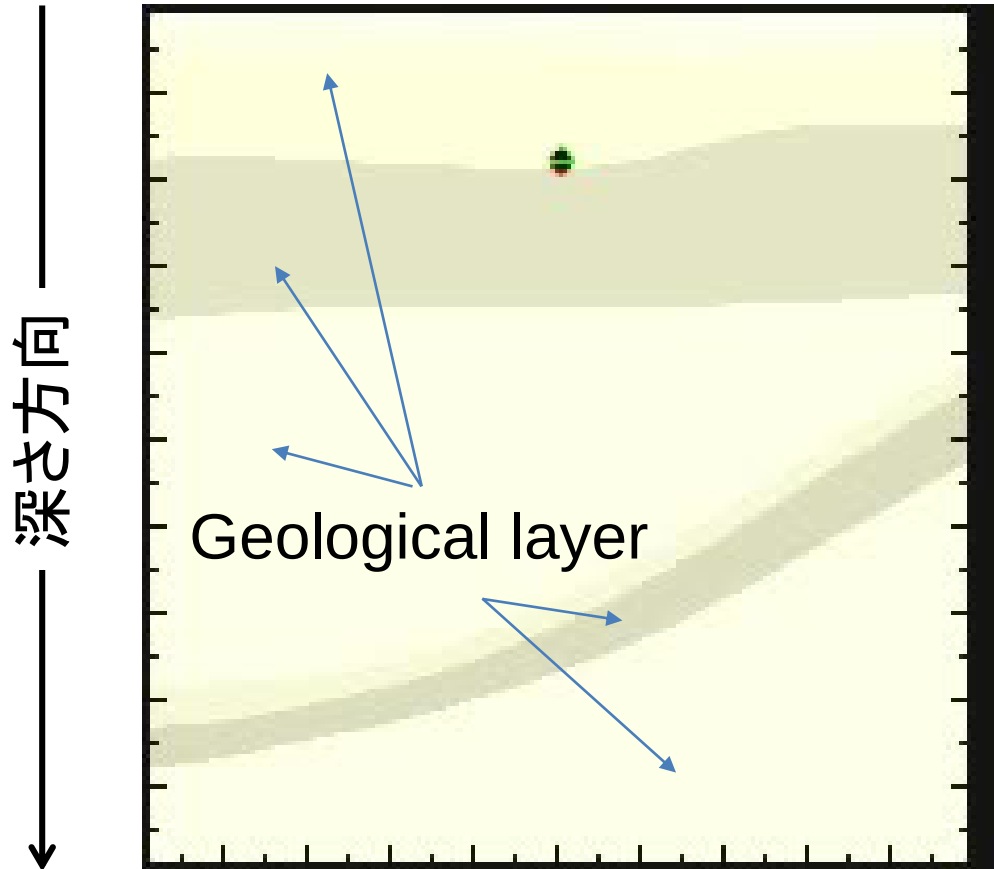
節点数 約600万点（AICSの建屋）
 並列数 1000計算ノード: 1000プロセス: 16スレッド/1プロセス（@AICS）
 1000計算ノード: 1000プロセス: 16スレッド/1プロセス
 （@ほっともっとフィールド神戸）



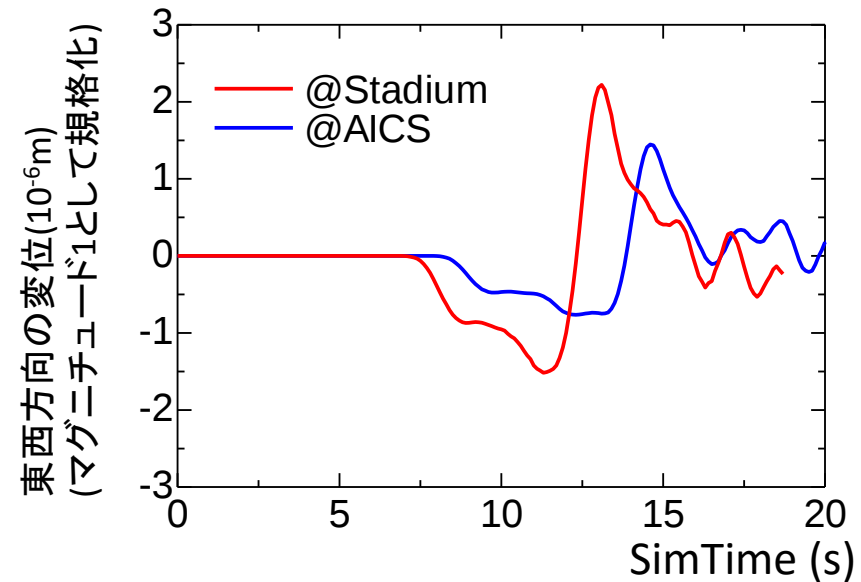
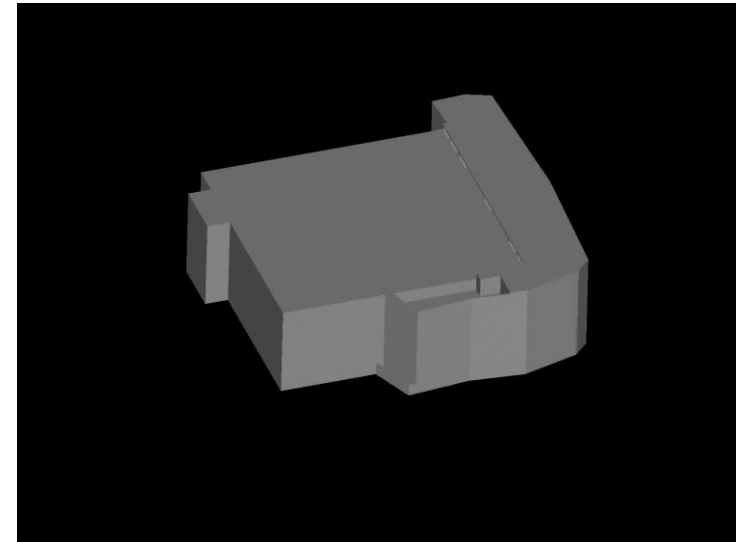
計4560計算ノード、72960コア

Oakleaf-FXによる大規模実行の結果

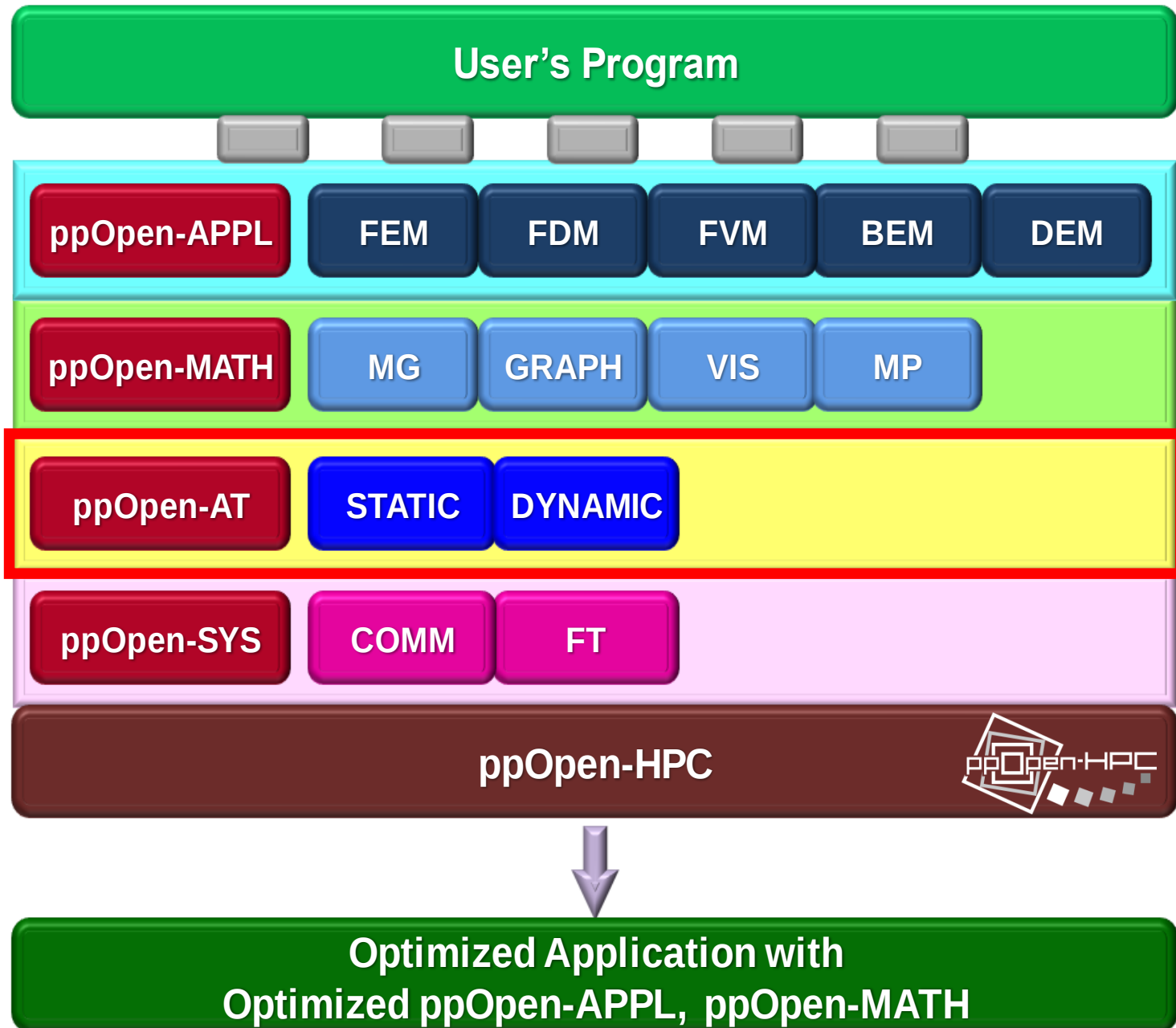
← 東西方向 →



震源からの地震波動の伝播
(赤:p波、緑:s波)



建物の変位



ppOpen-AT

- Automatic tuning (AT) enables smooth and easy shifts to further development on new and future architectures, through use of ppOpen-AT/STATIC and ppOpen-AT/DYNAMIC.
- A special directive-based AT language based on ABCLibscript is developed for specific procedures in scientific computing, focused on optimum memory access.
 - Geometries
 - Problem size
 - H/W Parameters

ppOpen-AT のシステム概要

専門家
ユーザの利用



専門知識

ppOpen-AT
ディレクティブ

ppOpen-APPL/*

自動生成

最適化済
ppOpen-
APPL/*

非専門家
ユーザ
の利用



ppOpen-AT
オート
チューナー

候補の
選択

最適化
候補 n

最適化
候補3

候補2

候補1

性能情報の取得

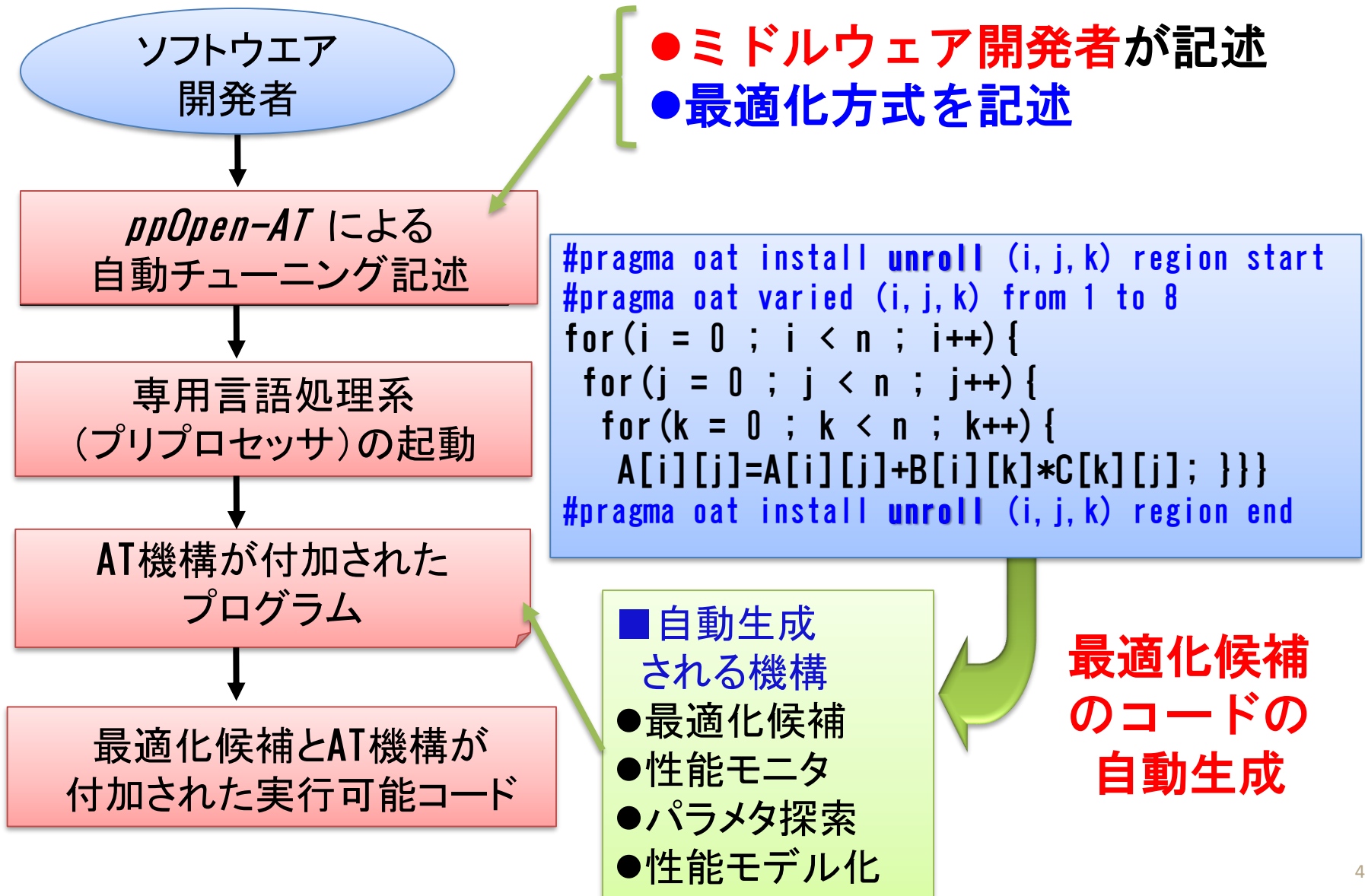
チューニング候補
のコードが
自動生成される

Xeon Phi

FX10

:対象計算機

AT専用言語 *ppOpen-AT* によるソフトウェア開発手順



Optimization of ppOpen-APPL/FDM (Seism3D) by ppOpen-AT

- A single node of Intel Xeon Phi (60 cores, 240 threads)

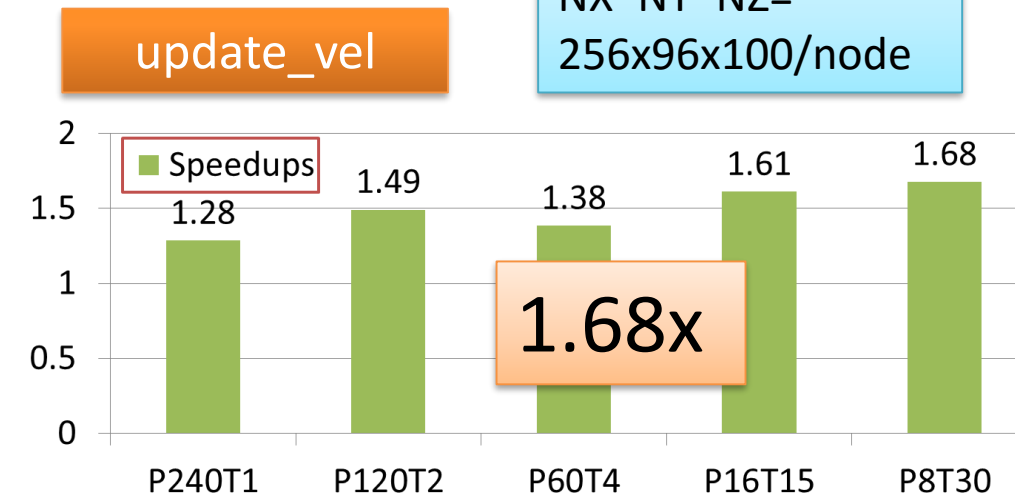
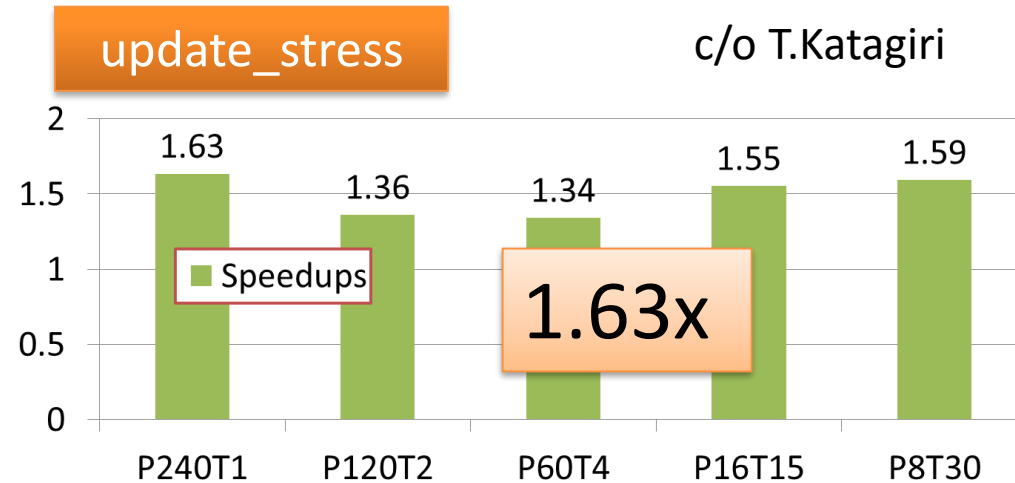
- P240T1: Flat MPI (240 process with 1 thread)
- P8T30: 8 proc's with 30 threads
- Speed-ups based on execution without auto-tuning

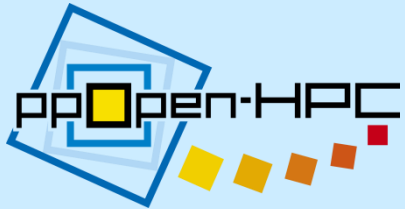
- **update_stress**

- 3-nested FDM loops, with a lot of operations
- “Loop Splitting” is effective

- **update_vel**

- 3-nested FDM loops, medium amount of operations
- “Loop Fusion” is effective (i-j-k -> i*j-k)





コード公開スケジュール

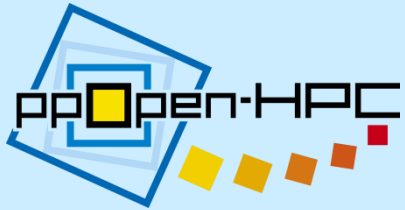
(英語ドキュメント付き, MITライセンス)

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

- 毎年のSC-XYで更新, 公開
- Flat MPI, OpenMP/MPIハイブリッド並列
- Multicore/Manycoreクラスタ向け→Xeon Phi最適化

公開の履歴

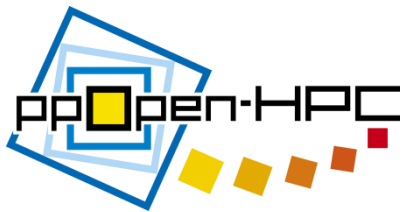
- SC12, Nov 2012 (Ver.0.1.0)
- SC13, Nov 2013 (Ver.0.2.0)
- SC14, Nov 2014 (Ver.0.3.0)
- SC15, Nov 2015 (Ver.1.0.0)(予定)



Ver.1.0における新機能

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

- ppOpen-APPL/FDM
 - OpenACC version
- ppOpen-APPL/FVM
 - Matrix Assembly and Linear Solvers
- ppOpen-APPL/BEM
 - HACApK library for H-matrix computation (OpenMP/MPI Hybrid Version)
- ppOpen-MATH/MP
 - First Release
 - Coupler for Multiphysics Simulations, Loose Coupling of FEM & FDM



研究協力・普及

- 国際的共同研究
 - Lawrence Berkeley National Lab.
 - 国立台湾大学
 - **ESSEX/SPPEXA/DFG, Germany**
 - IPCC (Intel Parallel Computing Ctr.)
- 普及
 - 大規模シミュレーションへの適用
 - CO₂ 地下貯留, 物性物理
 - 宇宙物理, 地震シミュレーション
 - ppOpen-AT, ppOpen-MATH/VIS, ppOpen-MATH/MP, 線形ソルバー群
 - H行列ライブラリ
 - 国際WS (2012,13,15)
 - 講習会 (東大センター), 講義

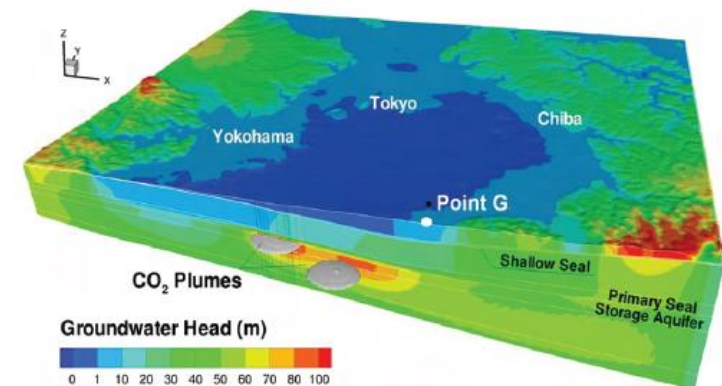
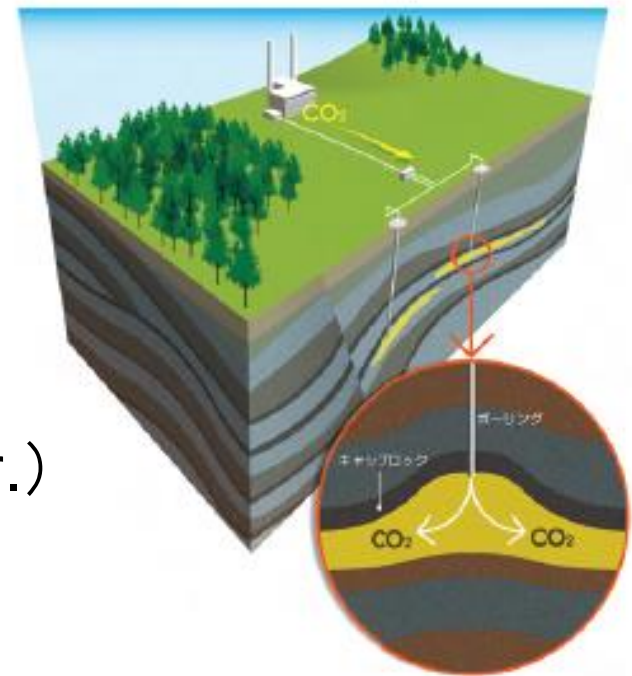
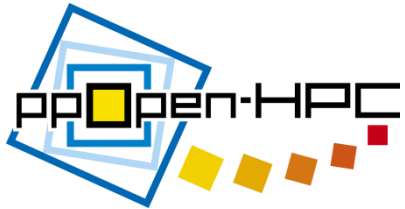
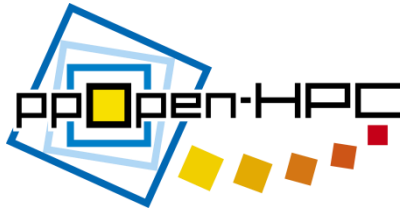


図-4 CO₂ 圧入後の地下水圧 (全水頭換算) の分布 (100 年後)



共同研究等事例(1/3)

- ppOpen-AT関連共同研究
 - 工学院大学 田中研究室
 - 田中研究室開発のAT方式(d-spline方式)の適用対象としてppOpen-ATのAT機能を拡張
 - 東京大学 須田研究室
 - 電力最適化のため, 須田研究室で開発中のAT方式と電力測定の共通APIを利用し, ppOpen-ATを用いた電力最適化方式を提案[Katagiri et al. IEEE/MCSoc 2013 Best Paper Award]
- JHPCN共同研究課題
 - 高精度行列 - 行列積アルゴリズムにおける並列化手法の開発(東大, 早稲田大)(H24年度)(研究としては継続)
 - 高精度行列-行列積演算における行列-行列積の実装方式選択に利用
 - 粉体解析アルゴリズムの並列化に関する研究(東大, 法政大)(H25年度)
 - 粉体シミュレーション向け高速化手法の性能パラメタ選択にAT適用検討



共同研究等事例(2/3)

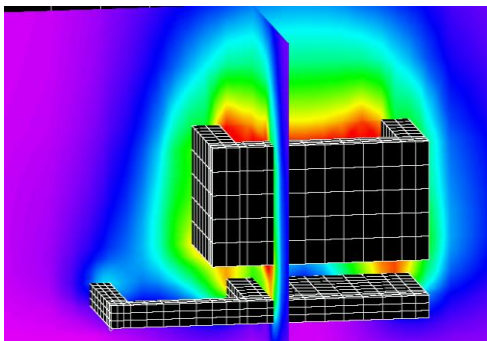
- JHPCN共同研究課題(続き)
 - 巨大地震発生サイクルシミュレーションの高度化(京大, 東大他)(H24・25年度)(研究としては継続: JAMSTEC/AICS)
 - Hマトリクス, 領域細分化
 - ポストペタスケールシステムを目指した二酸化炭素地中貯留シミュレーション技術の研究開発(大成建設, 東大)(H25年度)(研究としては継続)
 - 疎行列ソルバー, 並列可視化
 - 降着円盤シミュレーション(千葉大, 東大)(H22年度～)
 - 疎行列ソルバー, 並列可視化
 - 太陽磁気活動の大規模シミュレーション(東大(地球惑星, 情報基盤センター))(H25年度～)
 - 疎行列ソルバー, 並列可視化

共同研究等事例(3/3)

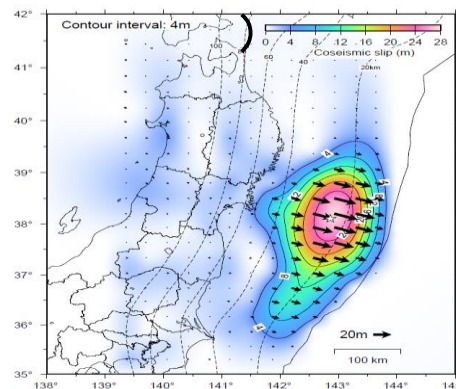
• Hマトリクス関連

- 巨大地震発生サイクルシミュレーションの高度化(京大, JAMSTEC, 理研AICS, 東大地震研)(前述)
- 超伝導送電線シミュレーション(京大院工)
- ボース=アインシュタイン凝縮シミュレーション(University of Belgrade)

電磁場解析

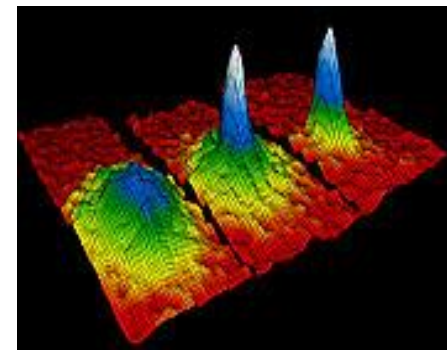


地震サイクル
シミュレーション
(JAMSTEC/AICS)

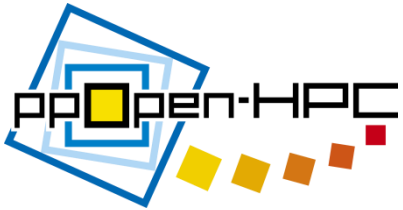


Transcribed from
Ohtani et al (2011)

ボース=アインシュタイン凝縮
(University of Belgrade)



Transcribed from NIST
Image Gallery



講習会・講義

- ppOpen-MATH/VIS
 - 並列可視化のための教材として使用
 - 東京大学大学院講義
 - 並列計算プログラミング・先端計算機演習(理学系研究科)
 - 科学技術計算II(情報理工学系研究科)
 - コンピュータ科学特別講義II(情報理工学系研究科)
 - 理研AICS HPC Summer School
 - 東京大学情報基盤センターお試しアカウント付き並列プログラミング講習会「MPI応用編:並列有限要素法」
- ppOpen-APPL/FDM
 - 東京大学教養学部「全学体験ゼミナール」, 地震波伝播の並列シミュレーションの演習の教材として活用
- スパコン講習会による普及
 - 東京大学情報基盤センターお試しアカウント付き並列プログラミング講習会
 - 2014年以降(複数回)

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)