

並列有限要素法による
三次元定常熱伝導解析プログラム
並列可視化

中島 研吾

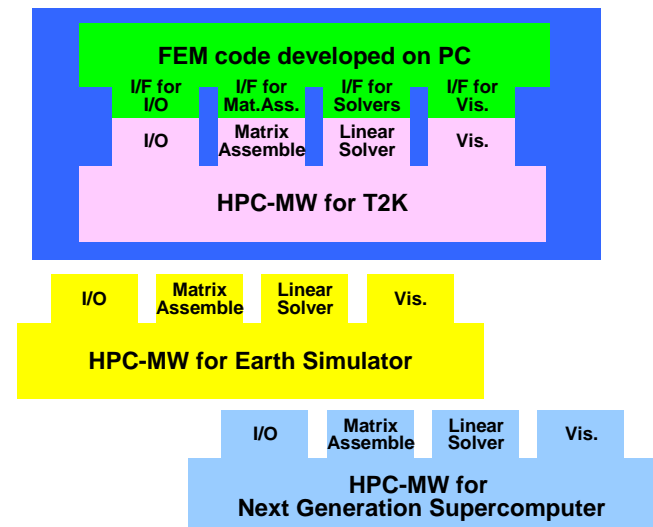
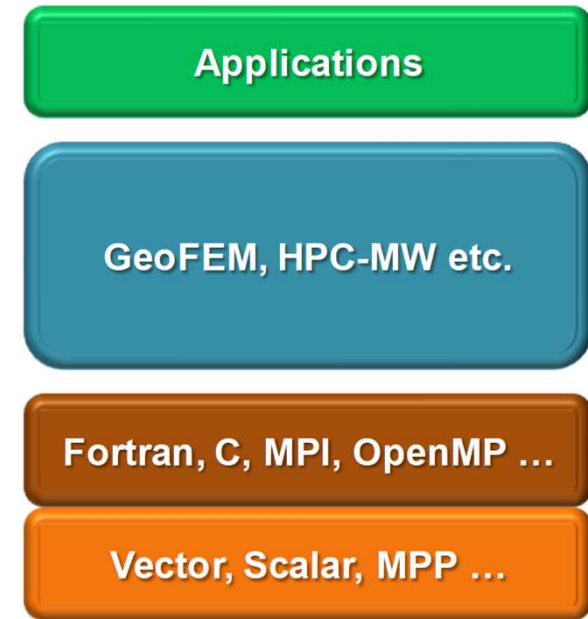
東京大学情報基盤センター

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

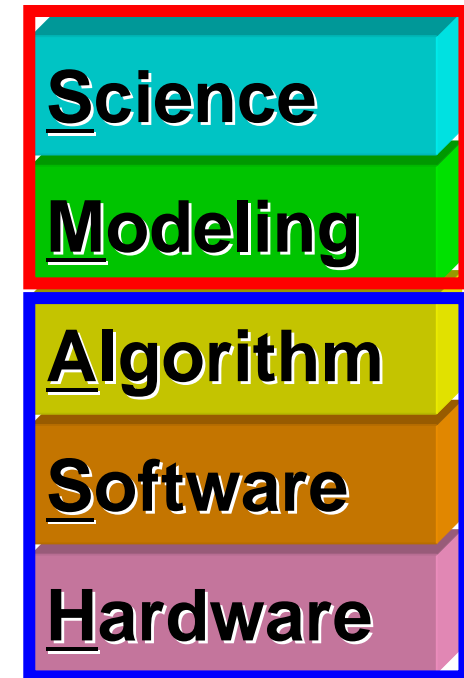
背景(2/2)

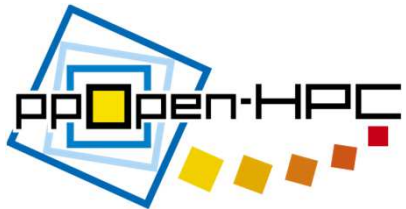
- A.D.2000年前後
 - GeoFEM, HPC-MW
 - 地球シミュレータ, Flat MPI, FEM
- 現在: より多様, 複雑な環境
 - マルチコア, GPU
 - ハイブリッド並列
 - MPIまでは何とかたどり着いたが...
 - 「京」でも重要
 - CUDA, OpenCL, OpenACC
 - ポストペタスケールからエクサスケールへ
 - より一層の複雑化



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**

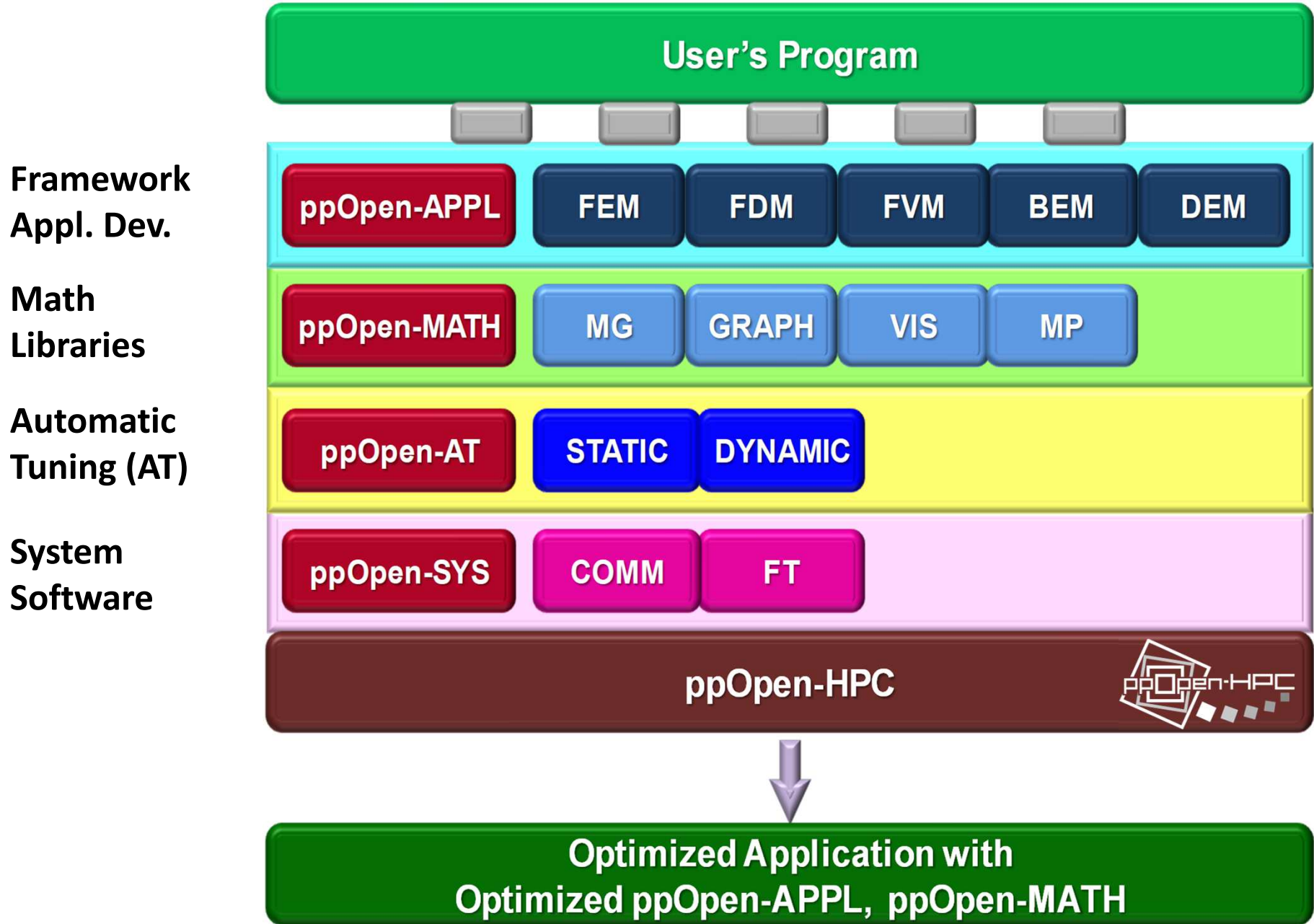


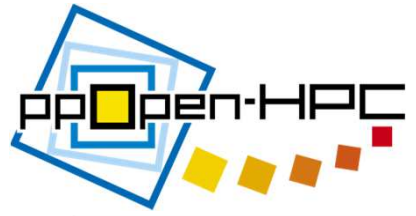


ppOpen-HPC: Overview

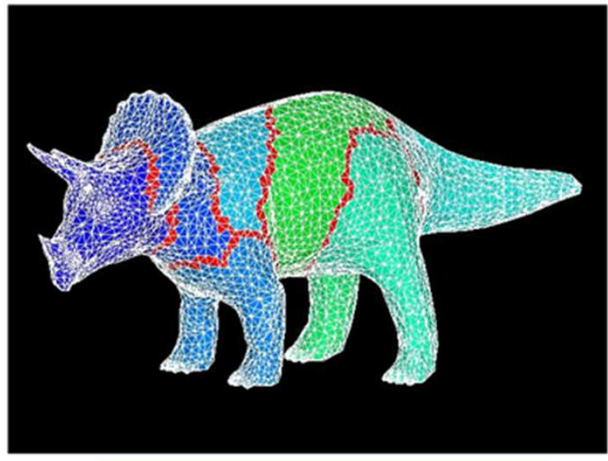
- Application framework with automatic tuning (AT)
 - ✓ “pp” : post-peta-scale
- Five-year project (FY.2011-2015) (since April 2011)
 - ✓ P.I.: Kengo Nakajima (ITC, The University of Tokyo)
 - ✓ Part of “Development of System Software Technologies for Post-Peta Scale High Performance Computing” funded by JST/CREST (Supervisor: Prof. Mitsuhsa 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/>
 - ✓ <https://github.com/Post-Peta-Crest/ppOpenHPC>
 - ✓ English Documents, MIT License



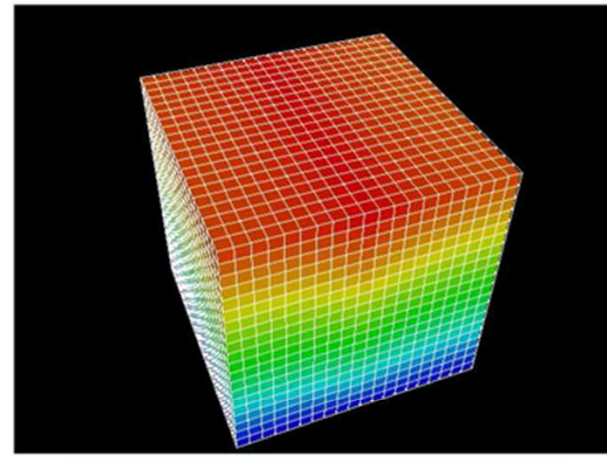




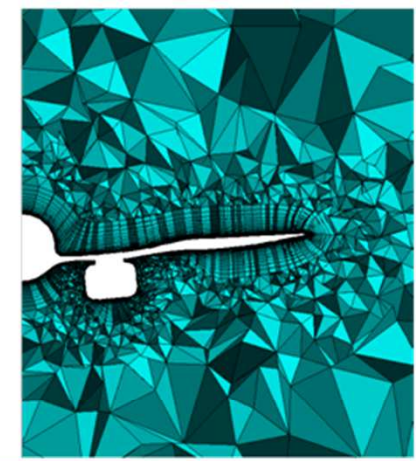
ppOpen-HPC covers ...



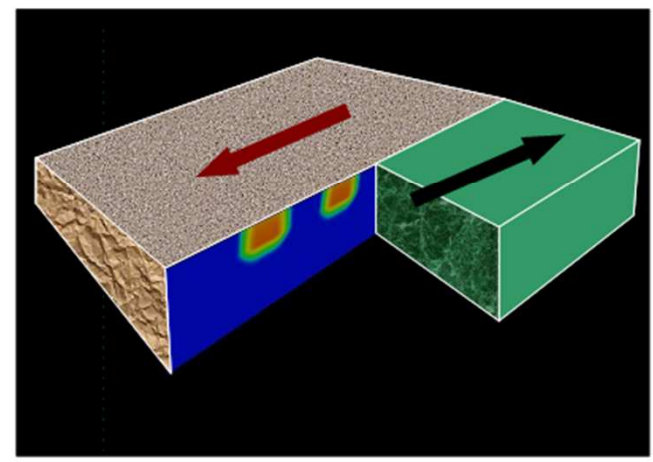
FEM
Finite Element Method



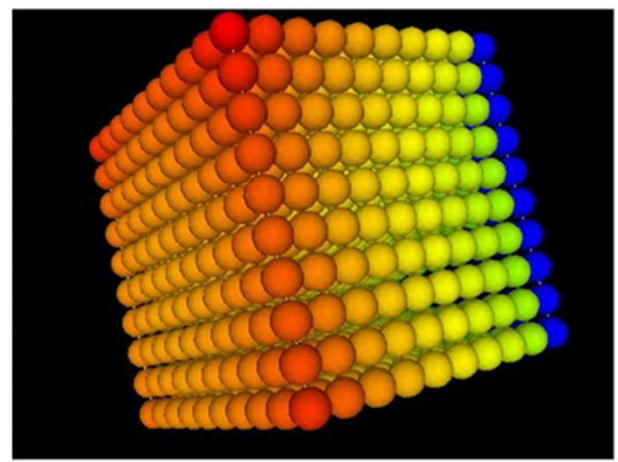
FDM
Finite Difference Method



FVM
Finite Volume Method



BEM
Boundary Element Method



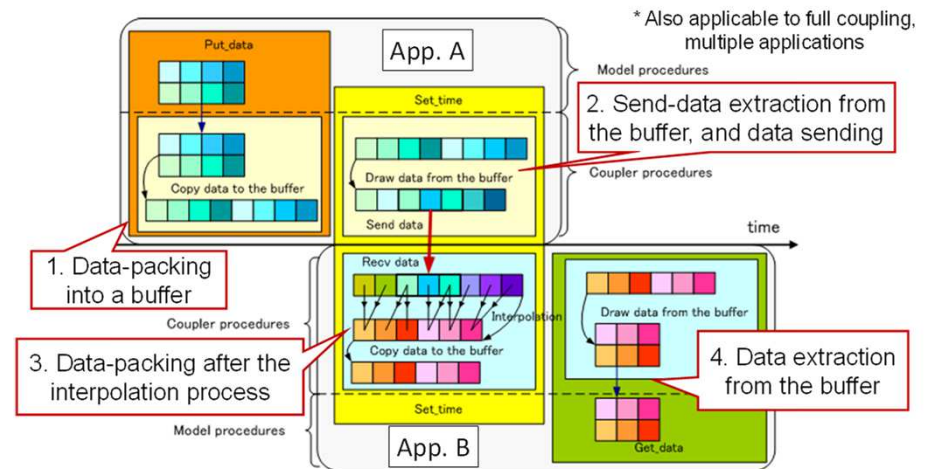
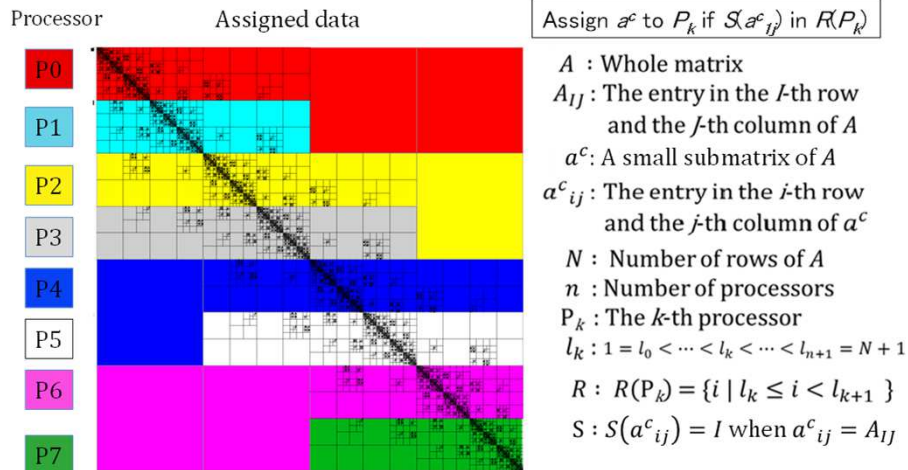
DEM
Discrete Element Method

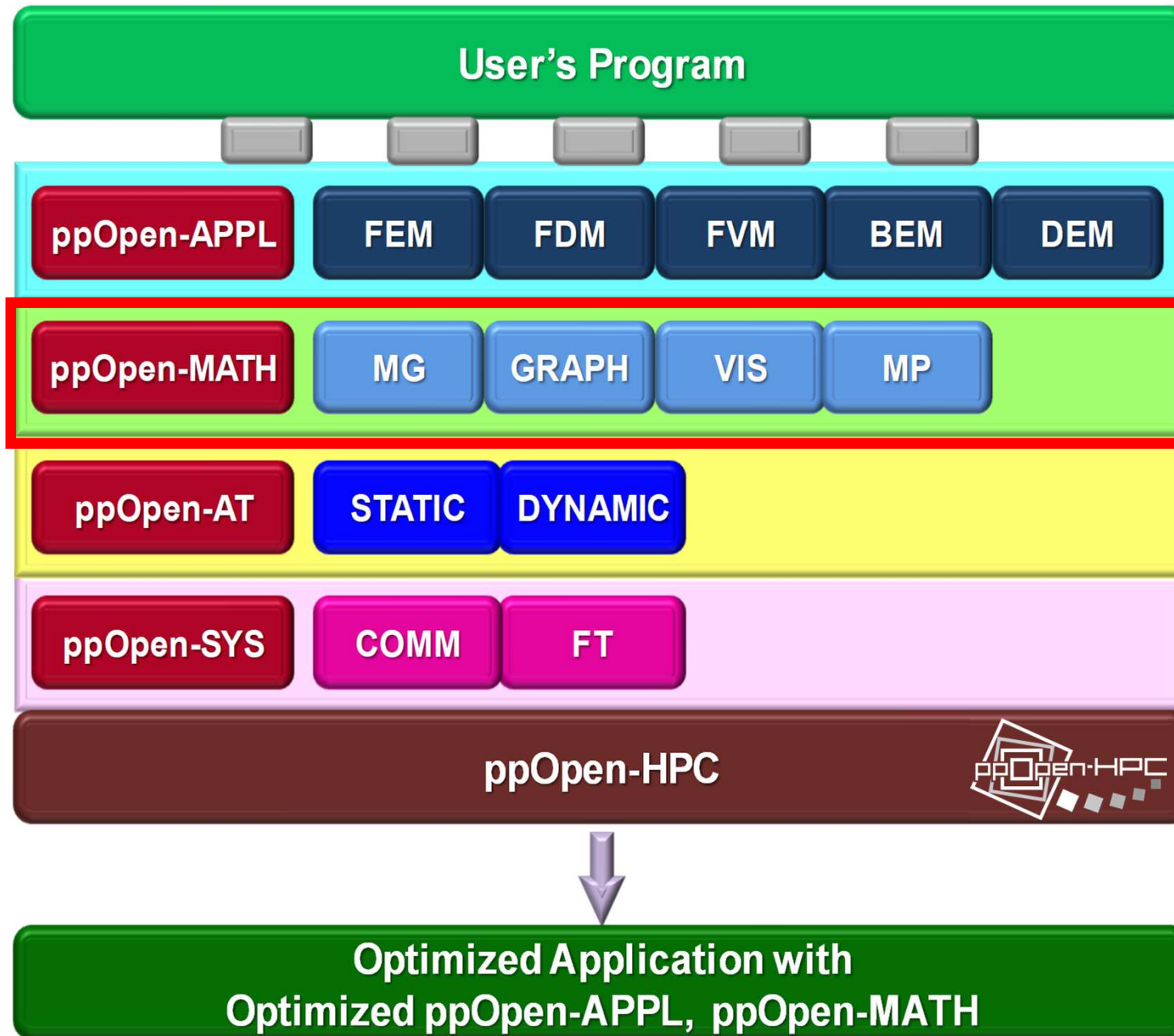


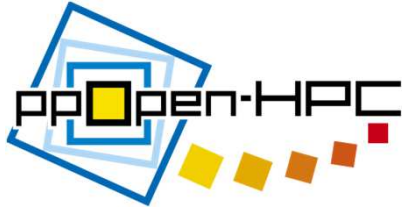
New Features in Ver.1.0.0

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

- **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)
- Matrix Assembly and Linear Solvers for ppOpen-APPL/FVM



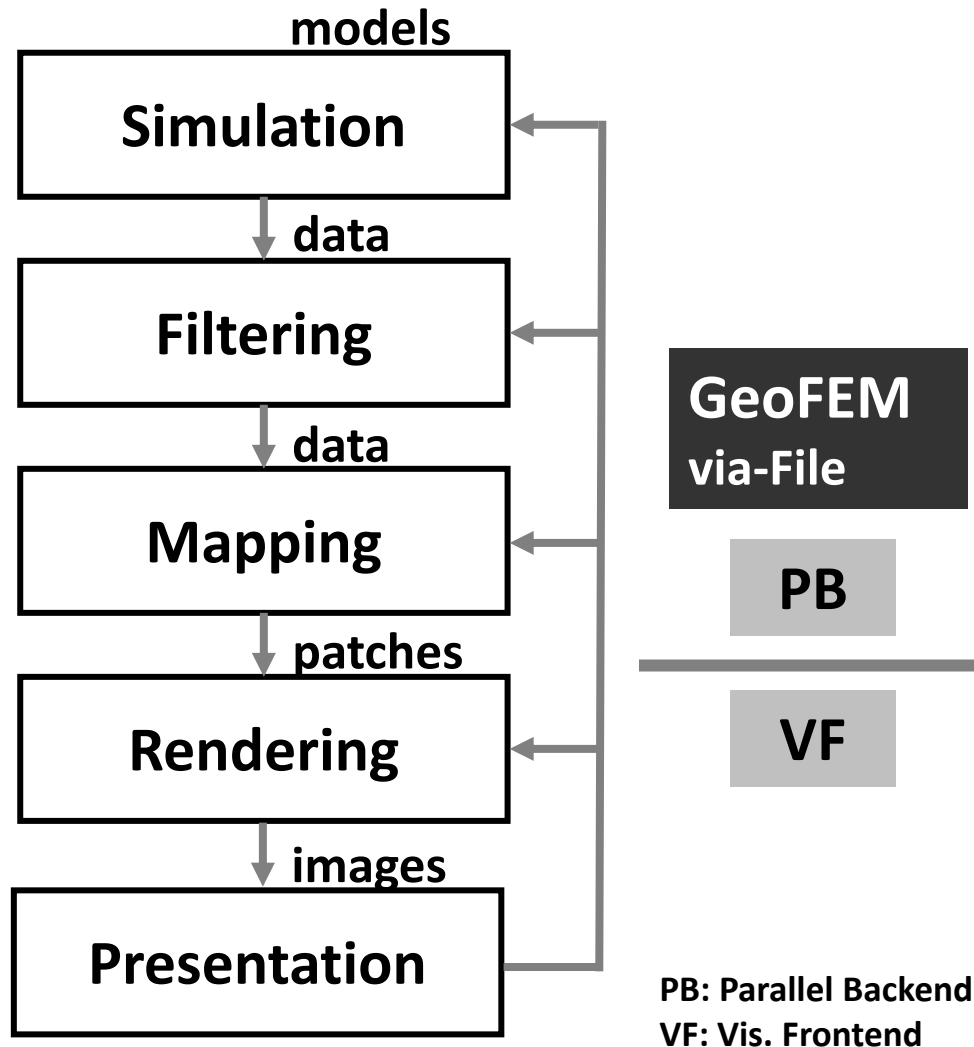




ppOpen-MATH

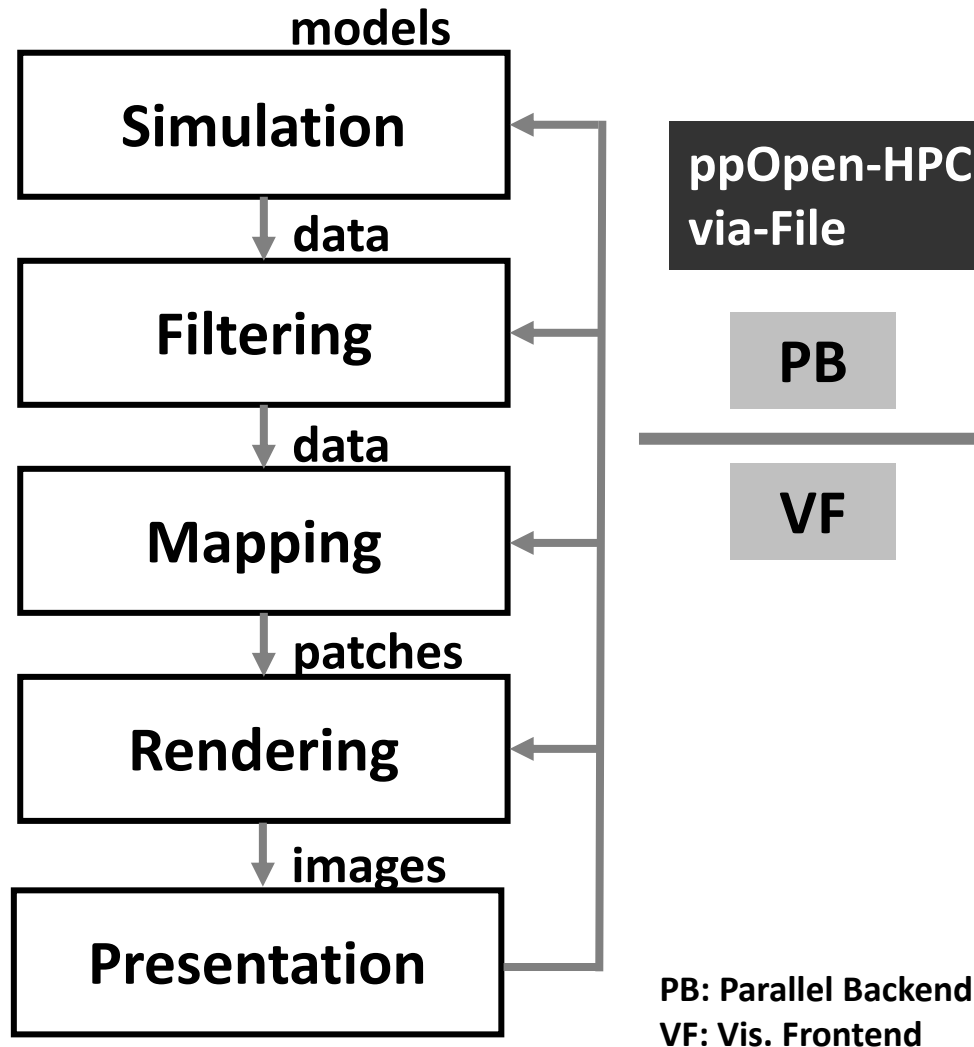
- ppOpen-HPCにおける共通ライブラリ
- ppOpen-MATH/MG
 - 多重格子法ソルバー
- ppOpen-MATH/GRAPH
 - 並列グラフライブラリ(マルチスレッド版): RCM, MC(開発中)
- ppOpen-MATH/VIS
 - 並列可視化ライブラリ
- ppOpen-MATH/MP
 - 弱連成カプラー

ppOpen-HPCにおける 並列可視化の考え方



- 並列計算・並列可視化を並行して同時に実行
- 自己完結的なファイルを生成してPCで見る (e.g. MicroAVS, ParaView)。
- GeoFEMはPatch抽出型で、例えば視点を変えることはできたが、可視化する変数、切り出す面等を変更することはできなかった。
- **ピーク(最大, 最小), 分布を抑えることが大事, 形状もある程度再現できてほしい。**

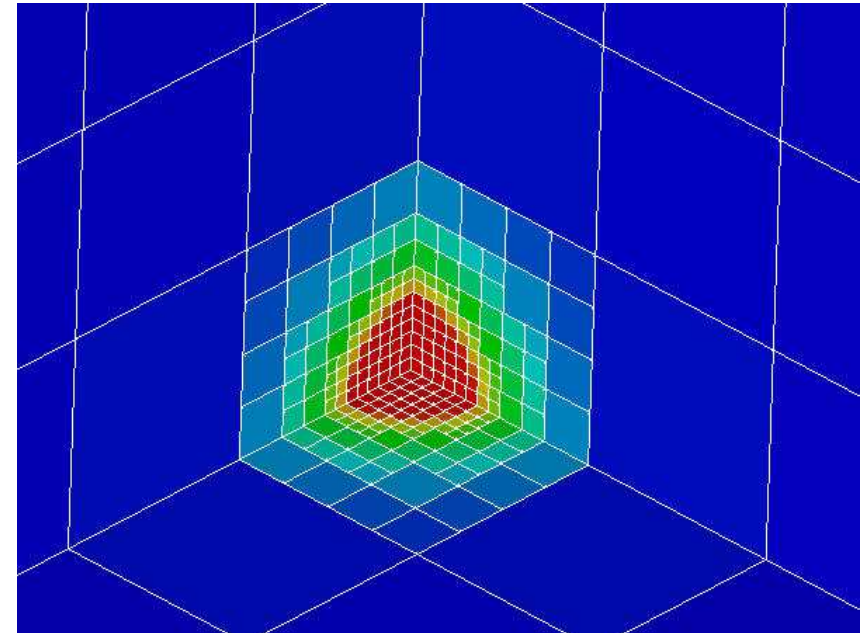
ppOpen-HPCにおける 並列可視化の考え方



- 並列計算・並列可視化を並行して同時に実行
- 自己完結的なファイルを生成してPCで見る(e.g. MicroAVS, ParaView)。
- GeoFEMはPatch抽出型で、例えば視点を変えることはできたが、可視化する変数、切り出す面等を変更することはできなかった。
- ピーク(最大, 最小), 分布を抑えることが大事, 形状もある程度再現できてほしい。

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

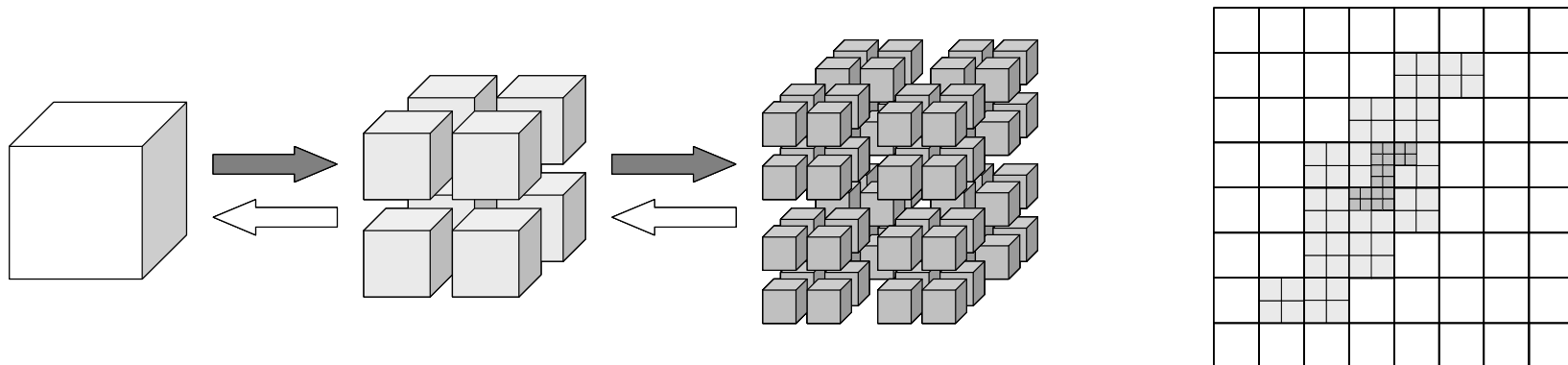


[Refine]

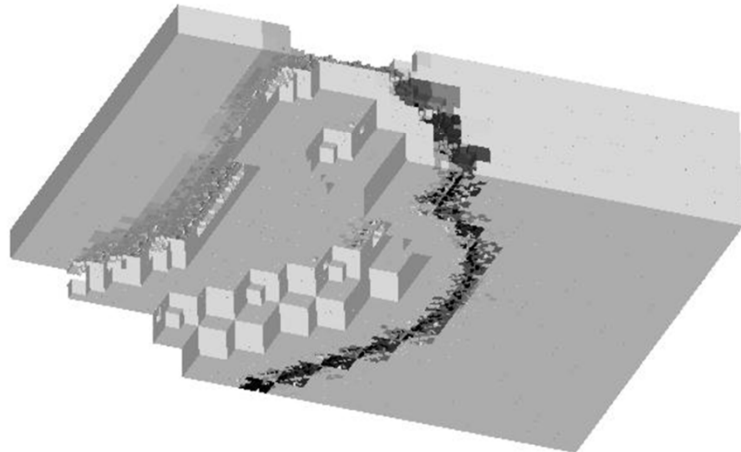
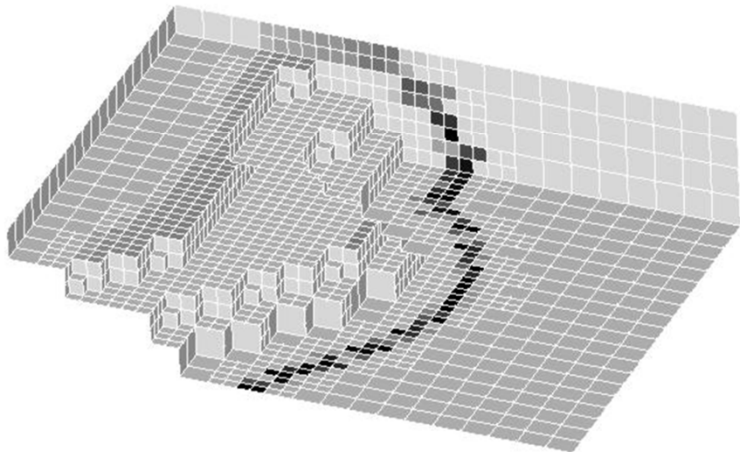
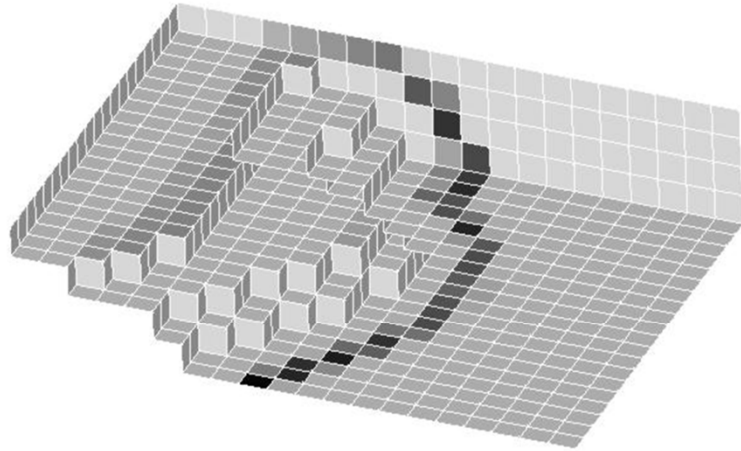
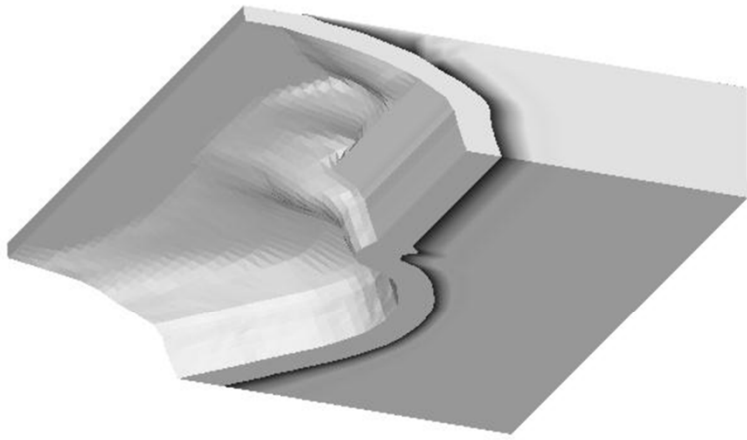
AvailableMemory = 2.0	Available memory size (GB), not available in this version.
MaxVoxelCount = 500	Maximum number of voxels
MaxRefineLevel = 20	Maximum number of refinement levels

Simplified Parallel Visualization using Background Voxels

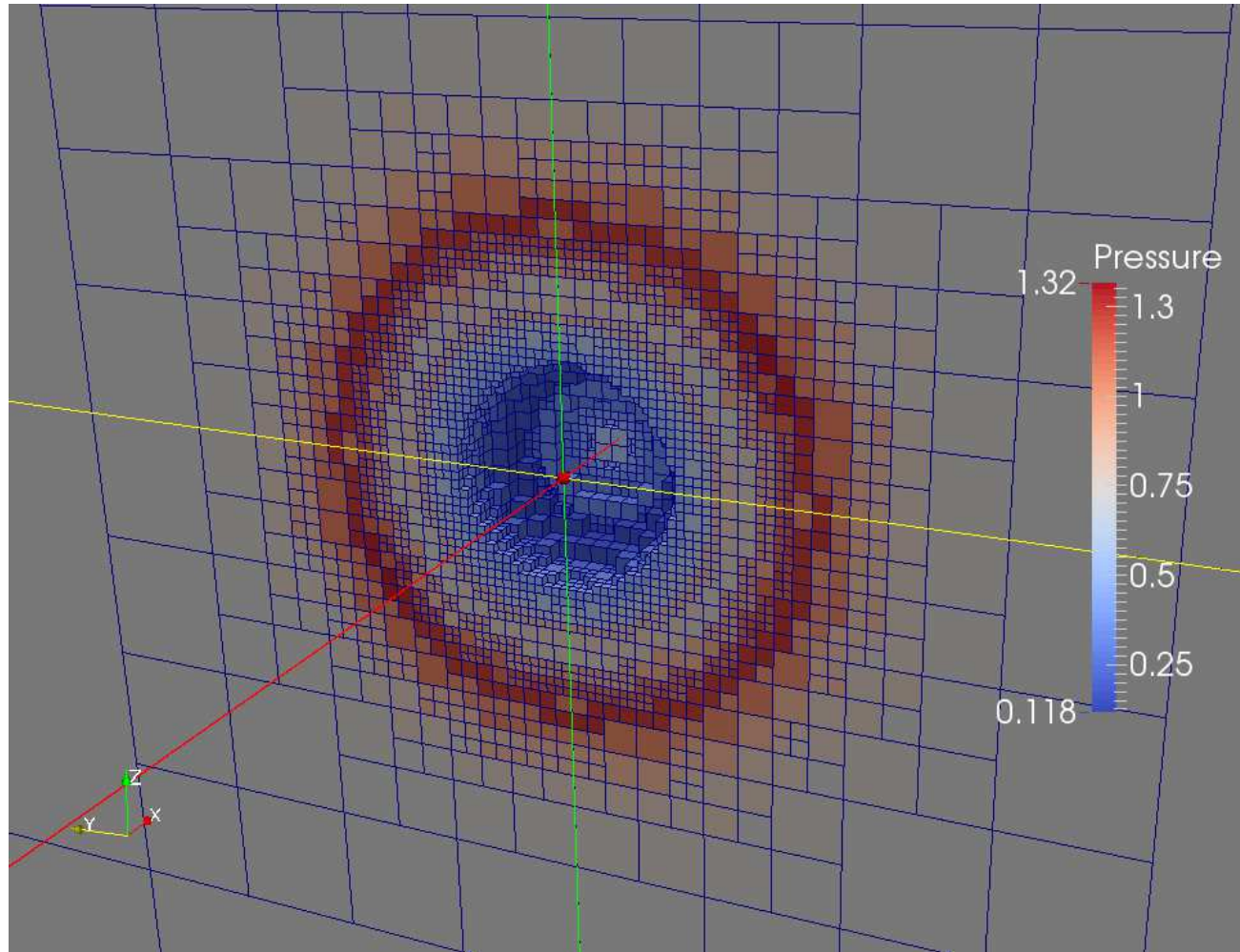
- Octree-based AMR
- AMR applied to the region where gradient of field values are large
 - stress concentration, shock wave, separation etc.
- If the number of voxels are controlled, a single file with 10^5 meshes is possible, even though entire problem size is 10^9 with distributed data sets.



Voxel Mesh (adapted)



Flow around a sphere



pFEM3D + ppOpen-MATH/VIS

Fortran

```
>$ cd /work/gt00/t00XXX/pFEM
>$ cp /work/gt00/z30088/pFEM/F/pfem3dVIS.tar .
>$ tar xvf pfem3dVIS.tar
>$ cd pfem3d/srcV
>$ make
>$ cd ../runV
>$ ls solv
    solv
```

C

```
>$ cd /work/gt00/t00XXX/pFEM
>$ cp /work/gt00/z30088/pFEM/C/pfem3dVIS.tar .
>$ tar xvf pfem3dVIS.tar
>$ cd pfem3d/srcV
>$ make
>$ cd ../runV
>$ ls solv
    solv
```

pFEM/pfem3d/srcV/Makefile (F)

```
include Makefile.in
```

```
FFLAGSL    = -I/work/gt00/z30088/ppohVIS/include  
FLDFLAGSL  = -L/work/gt00/z30088/ppohVIS/lib  
LIBSL     = -lfpohvispfem3d -lppohvispfem3d
```

```
.SUFFIXES:
```

```
.SUFFIXES: .o .f90 .f
```

```
.f.o:
```

```
$(FC) -c $(FFLAGS) $(FFLAGSL) $< -o $@
```

```
.f90.o:
```

```
$(F90) -c $(F90FLAGS) $(FFLAGSL) $< -o $@
```

```
TARGET = ../run/solv
```

```
OBJS = ¥
```

```
pfem_util.o ...
```

```
all: $(TARGET)
```

```
$(TARGET): $(OBJS)
```

```
$(F90) -o $(TARGET) $(F90FLAGS) $(FFLAGSL) $(OBJS)
```

```
$(LDFLAGSL) $(LIBS) $(LIBSL) $(FLDFLAGSL)
```

```
clean:
```

```
rm -f *.o *.mod $(TARGET)
```

```
distclean:
```

```
rm -f *.o *.mod $(TARGET)
```

pFEM/pfem3d/srcV/Makefile (C)

```
include Makefile.in
```

```
CFLAGSL = -I/work/gt00/z30088/ppohVIS/include
```

```
LDFLAGSL = -L/work/gt00/z30088/ppohVIS/lib
```

```
LIBSL = -lppohvispfem3d
```

```
.SUFFIXES:
```

```
.SUFFIXES: .o .c
```

```
.c.o:
```

```
$(CC) -c $(CFLAGS) $(CFLAGSL) $< -o $@
```

```
TARGET = ../run/solv
```

```
OBJS = test1.o ...
```

```
all: $(TARGET)
```

```
$(TARGET): $(OBJS)
```

```
$(CC) -o $(TARGET) $(CFLAGS) $(CFLAGSL) $(OBJS)
```

```
$(LDFLAGSL) $(LIBS) $(LIBSL)
```

```
clean:
```

```
rm -f *.o *.mod $(TARGET)
```

```
distclean:
```

```
rm -f *.o *.mod $(TARGET)
```

pFEM/pfem3d/srcV/Makefile.in (1/2)

```
# Install directory
PREFIX      = /work/gt00/z30088/ppohVIS
BINDIR      = $(PREFIX)/bin
INCDIR      = $(PREFIX)/include
LIBDIR      = $(PREFIX)/lib

# TetGen directory
TETGENDIR   = $(HOME)/usr/local/tetgen1.4.3
TETINCDIR   = $(TETGENDIR)
TETLIBDIR   = $(TETGENDIR)

# C compiler settings
CC          = mpiicc
CFLAGS      = $(CINCDIR) $(COPTFLAGS)
COPTFLAGS   = -O3 -axCORE-AVX512 -align -qopenmp
CINCDIR     = -I. -I$(TETINCDIR)

# C++ compiler settings
CXX         = mpiicpc
CXXFLAGS    = $(CXXINCDIR) $(CXXOPTFLAGS) -DTETGEN -DTETLIBRARY
CXXOPTFLAGS = -O3 -axCORE-AVX512 -align -qopenmp
CXXINCDIR   = -I. -I$(TETINCDIR)

# Fortran 77 compiler settings
FC          = mpiifort
FFLAGS      = $(FINCDIR) $(FOPTFLAGS)
FOPTFLAGS   = -O3 -axCORE-AVX512 -align array64byte -qopenmp
FINCDIR     = -I.
```


pFEM/pfem3d/srcV/Makefile.in (2/2)

```
# Fortran 90 compiler settings
F90          = mpiifort
F90FLAGS     = $(F90INCDIR) $(F90OPTFLAGS)
F90OPTFLAGS  = -O3 -axCORE-AVX512 -align array64byte -qopenmp
F90INCDIR   = -I.

# Linker settings
LD           = $(CC)
LDFLAGS     = $(LIBS)
#LIBS       = -L$(TETLIBDIR) -ltet -lm
LIBS        = -L$(TETLIBDIR) -lm
LDLIBDIR    =

# Archiver settings
AR          = ar rv

# ETC
CP          = cp -f
RM          = rm -rf
MKDIR      = mkdir -p
```

Fortran/main (1/2)

```

use solver11
use pfem_util
use ppohvis_pfem3d_util

implicit REAL*8(A-H,O-Z)
type(ppohVIS_BASE_stControl)           :: pControl
type(ppohVIS_BASE_stResultCollection)  :: pNodeResult
type(ppohVIS_BASE_stResultCollection)  :: pElemResult
character(len=PPOHVIS_BASE_FILE_NAME_LEN) :: CtrlName
character(len=PPOHVIS_BASE_FILE_NAME_LEN) :: VisName
character(len=PPOHVIS_BASE_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

pNodeResult%ListCount = 1
pElemResult%ListCount = 0
allocate (pNodeResult%Results (1))

call ppohVIS_PFEM3D_ConvResultNodeItem1N(                &
&      NP, ValLabel, X, pNodeResult%Results(1), iErr)

call ppohVIS_PFEM3D_Visualize(pNodeResult, pElemResult, pControl, &
&      VisName, 1, 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_PFEM3D_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_BASE_PrintError(stderr);
        MPI_Abort(MPI_COMM_WORLD, errno);
    };
};

```

C/main (2/2)

```
MAT_CON0 ();
MAT_CON1 ();

MAT_ASS_MAIN ();
MAT_ASS_BC ();

SOLVE11 ();

OUTPUT_UCD ();

pNodeResult=ppohVIS_BASE_AllocateResultCollection();
    if(pNodeResult == NULL) {
        ppohVIS_BASE_PrintError(stderr);
        MPI_Abort(MPI_COMM_WORLD,errno);
    };
    if(ppohVIS_BASE_InitResultCollection(pNodeResult, 1)) {
        ppohVIS_BASE_PrintError(stderr);
        MPI_Abort(MPI_COMM_WORLD,errno);
    };

    pNodeResult->Results[0] =

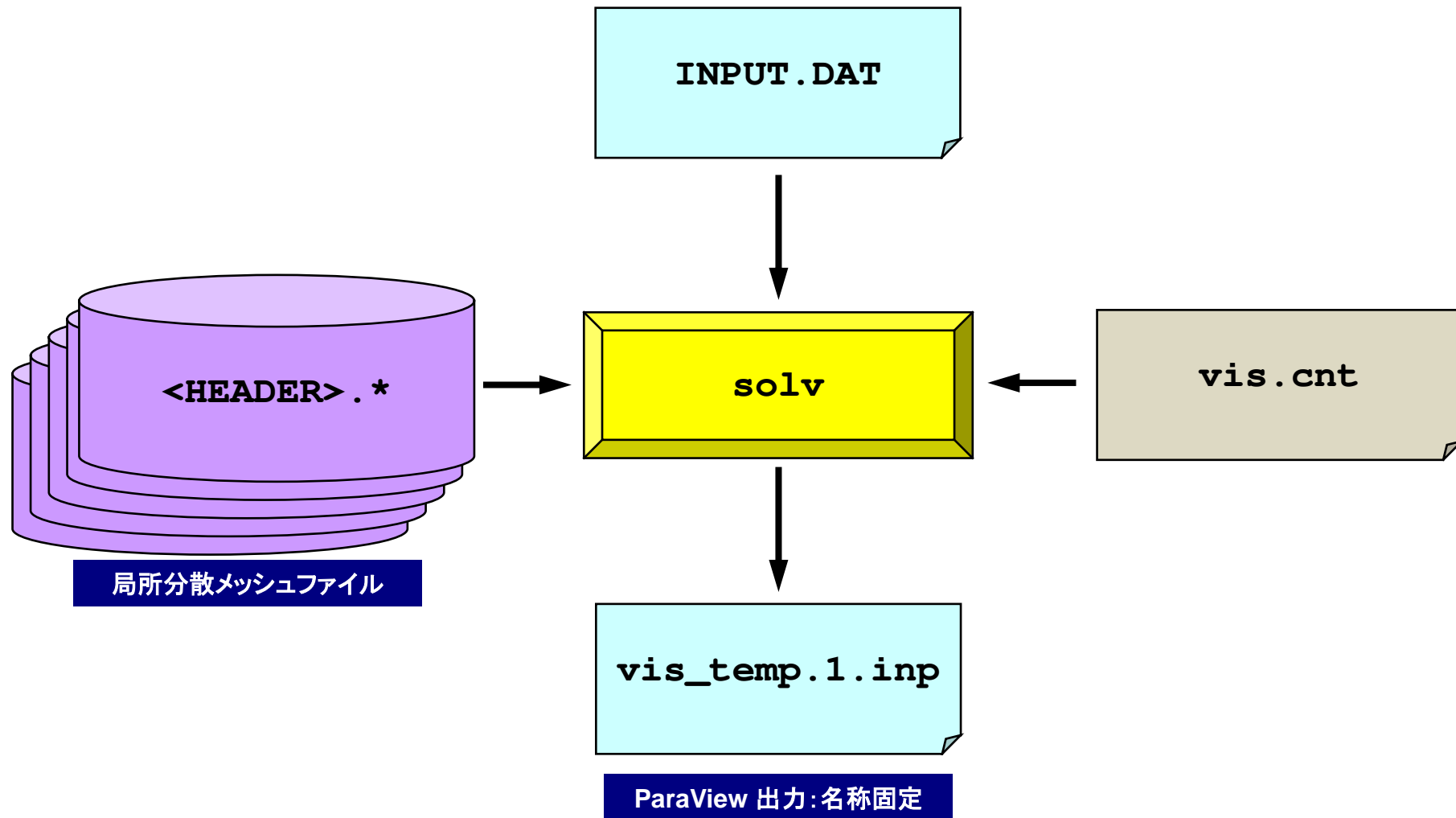
ppohVIS_PFEM3D_ConvResultNodeItemPart (NP, 1, 0, "temp", X);

START_TIME= MPI_Wtime();
    if(ppohVIS_PFEM3D_Visualize(pNodeResult, NULL, pControl, "vis", 1)) {
        ppohVIS_BASE_PrintError(stderr);
        MPI_Abort(MPI_COMM_WORLD,errno);
    };

ppohVIS_PFEM3D_Finalize();

PFEM_FINALIZE() ;
}
```

pFEM3D + ppOpen-MATH/VIS



Preparing Distributed Mesh Files

```
>$ cd /work/gt00/t00XXX/pFEM/pfem3d/runV  
(mesh.inp, mg.sh)  
>$ pjsub mg.sh
```

```
mesh.inp  
256 256 256  
 8   8   4  
pcube
```

256³ nodes into 8 × 8 × 4=256 partitions
16,777,216 nodes
16,581,375 elements
Each MPI process has 32x32x64 nodes

```
mg.sh  
#!/bin/sh  
#PJM -N "pmg"  
#PJM -L rscgrp=tutorial  
#PJM -L node=8  
#PJM --mpi proc=256  
#PJM -L elapse=00:10:00  
#PJM -g gt00  
#PJM -j  
#PJM -e err  
#PJM -o pmg.lst  
  
mpiexec.hydra -n ${PJM_MPI_PROC} ./pmesh  
  
rm wk.*
```

Computation + Visualization

```
>$ cd /work/gt00/t00XXX/pFEM/pfem3d/runV  
(INPUT.DAT, go.sh)  
>$ pjsub go.sh
```

INPUT.DAT

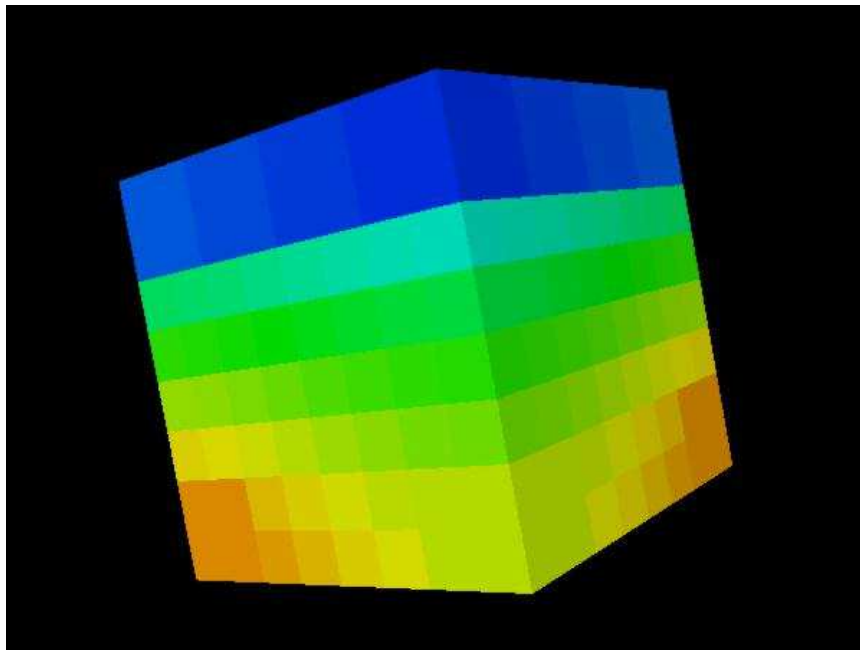
```
pcube  
2000  
1.0 1.0  
1.0e-08
```

go.sh

```
#!/bin/sh  
#PJM -N "VIS"  
#PJM -L rscgrp=tutorial  
#PJM -L node=8  
#PJM --mpi proc=256  
#PJM -L elapse=00:15:00  
#PJM -g gt00  
#PJM -j  
#PJM -e err  
#PJM -o testV.lst  
  
mpiexec.hydra -n ${PJM_MPI_PROC} ./solv
```

vis.cnt

[Refine]	細分化制御情報セクション
AvailableMemory = 2.0	利用可能メモリ容量 (GB) not in use
MaxVoxelCount = 1000	Max Voxel #
MaxRefineLevel = 20	Max Voxel Refinement Level
[Simple]	簡素化制御情報セクション
ReductionRate = 0.0	表面パッチ削減率
[Output]	出力形式
FileFormat = 2	=1:MicroAVS, =2:ParaView



Movie: 要素中心の値

16,777,216節点
16,581,375要素, 64 MPIプロセス



vis_temp.1.inp
1,436節点
1,002要素

COPY the File to Your PC

Oakbridge-CX

```
>$ cd /work/gt00/t00XXX/pFEM/pfem3d/runV  
>$ cp vis_temp.1.inp ~/.
```

PC

```
>$ scp t00XXX@obcx.cc.u-tokyo.ac.jp:~/vis_temp.1.inp .
```