高速化技術の紹介
& 東大センターにおける産業利用の将来

中島研吾
東京大学情報基盤センター

第11回PCクラスタシンポジウム　パネルディスカッション「実用アプリの加速のために何が必要か：PCクラスタコンソーシアムに何ができるか」
2011年12月8･9日

最近10年くらいやっていること 2008年も2009年にも話したこと, 2010年にも話した・・・

- 様々な科学技術手法の様々な処理に対応するライブラリの集合体に基づく科学技術アプリケーション開発基盤を整備し，ポストペタスケール環境における移植性，信頼性の高い科学技術アプリケーションの効率的な開発のためのフレームワークを提供する。
  - 多様なHPC環境（SC11：10種類以上のアーキテクチャ）
  - その急速な進歩・変動
- GeoFEM，HPC-MW/HEC-MW，OpenMM

Overview
Statistics
Geography of use
Team
Downloads
Documents
Wiki
Publications
News
Advanced

Downloads & Source Code

OpenMM

OpenMM-Accelerated GROMACS

OpenMM and AMBER

Solvent Model - ObcGbsa

This project uses Simtk's **Subversion code repository**, but has restricted access to project members.

News

OpenMM 2.0 Enables MD Acceleration on ATI and NVIDIA GPUs

New Protein Folding DBP

OpenMM

Project Overview

Description: OpenMM is a library which provides tools for modern molecular modeling simulation. As a library it can be hooked into any code, allowing that code to do molecular modeling with minimal extra coding.

Moreover, OpenMM has a strong emphasis on hardware acceleration, thus providing not just a consistent API, but much greater performance than what one could get from just about any other code available.

AVAILABILITY: See the download section for our latest preview release. A roadmap of future releases can be found on the Wiki.

MAILING LIST: Sign up for the OpenMM-news mailing list to receive updates about the project. (Click on Advanced -> Mailing Lists)

NEED HELP? Check out the discussion forums under Advanced -> Public Forums and the material from our workshops under Downloads. If you're new to molecular dynamics, check out the Wiki and OpenMM Zephyr.

CITING OPENMM: Any work that uses OpenMM should cite the papers listed on the Publications page.

Purpose/Synopsis: The functionality of OpenMM will (eventually) include everything that one would need to run modern molecular simulation.

Audience: Computational researchers interested in molecular simulations

Long Term Goals and Related Uses: While Molecular Dynamics is not new, the key advance here is the hardware acceleration (which has been extremely limited in acceptance due to the challenges involved) and the extensibility (allowing for rapid prototyping and development of new MD methods).

The functionality of OpenMM will (eventually) include everything that one would need to have to run modern molecular simulation, including

- Use of modern force fields (CHARMM, AMBER, OPLS, GROMOS, GROMACS, AMOEBA)

Project Lead

Vijay Pande
Contact

Peter Eastman
Contact

Mark Friedrichs
Contact
[Show All Leads]

Driving Biological Problems

This project is part of

Myosin Dynamics

Protein Folding

RNA Folding

Related Projects

OpenMM Zephyr

Normal Mode Langevin OpenMM Extension

Amber/Sander plus OpenMM

PyOpenMM

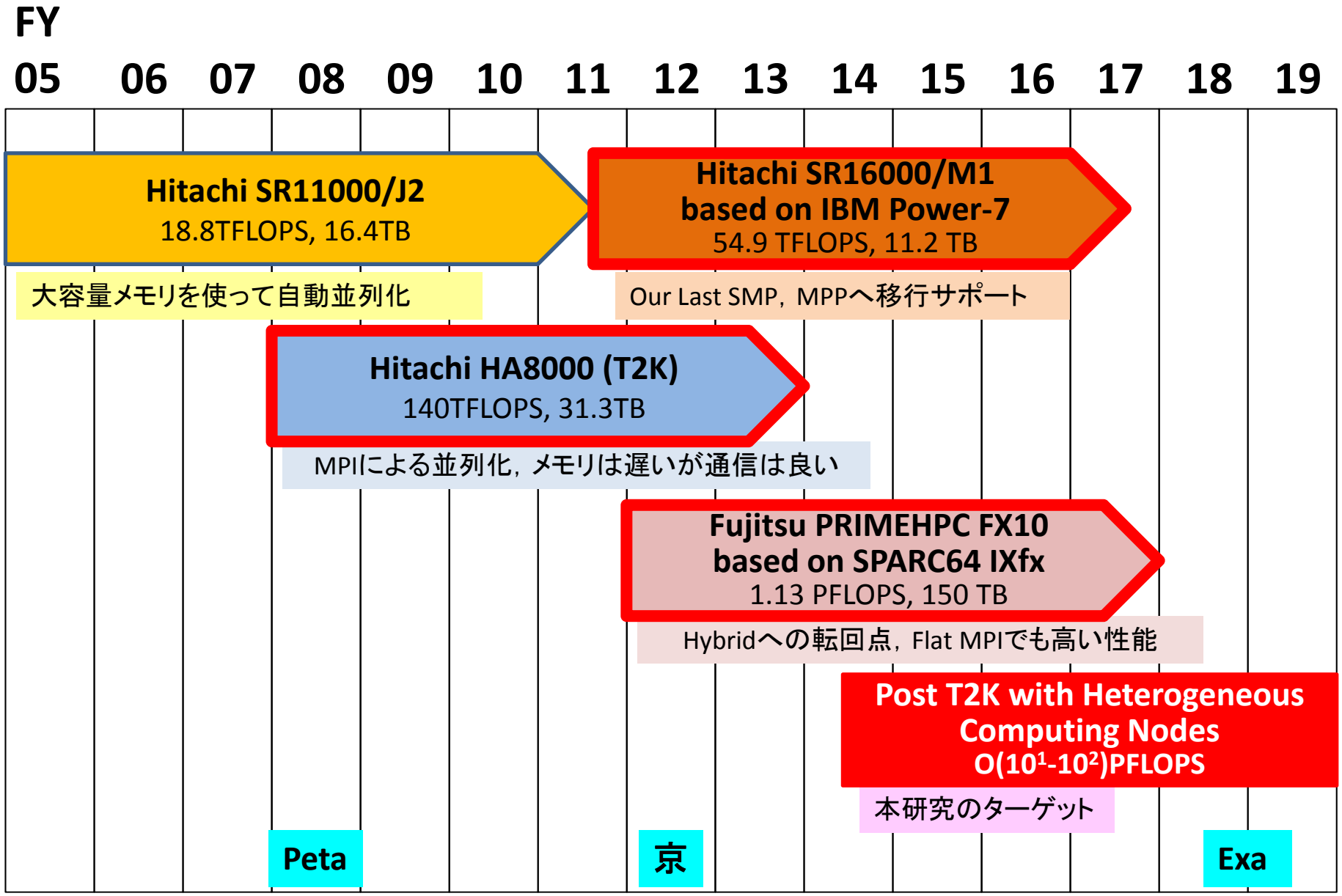
ppOpen-HPC

- 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
- ヘテロジニアスなアーキテクチャによる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
  - pp= post petascale，Five Key-Issues（J.J. Dongarra）を考慮
- 東大情報基盤センターに2014年度に導入予定の数10 PFLOPS級システム（ポストT2K）を最終的なターゲット：
  - スパコンユーザーの円滑な移行支援
  - T2K，東大FX10（1.13PFLOPS），京，TSUBAME，GPUクラスタ
- JST-CREST「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（H23～H27）」

対象とするアーキテクチャ

- ホストCPU
  - 8～16コア程度（またはそれ以上）を有するマルチコア低消費電力CPU
- Co-Processor
  - 数十～数百のコアを有する複数のCo-Processor
  - PCI Expressに接続
    - GPGPU：ホストCPUからCUDAまたはOpenCLによって利用する
    - Intel Many Integrated Core（MIC）アーキテクチャに基づくメニーコア（Knights Series）：軽量OS，コンパイラが稼働する
    - GPGPU＋MICメニーコアもあり得る
- ノード間ネットワーク：Infiniband等
- Co-Processor間通信はホストCPUを経ないで可能
- Hybrid並列プログラミングモデルは必須

東大情報基盤センターのスパコン

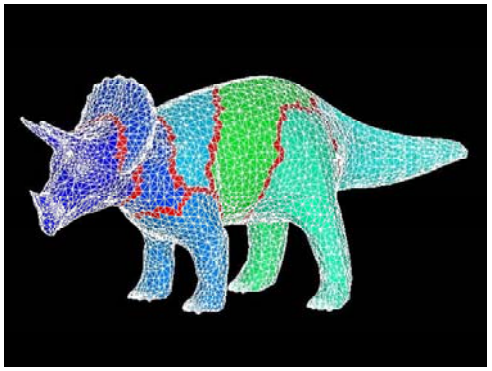


ppOpen-HPC：基本的方針

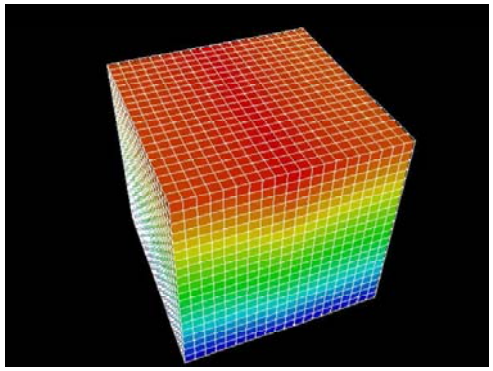
- 対象離散化手法による分類（次ページ）
- ハードウェアに依存しない共通インタフェースを有するアプリケーション開発用ライブラリ群，耐故障機能を含む実行環境を提供
  - 各離散化手法の特性に基づく
- 自動チューニング技術の導入により，様々な環境下における最適化ライブラリ，最適化アプリケーション自動生成を目指す
- 数値ライブラリ，システムソフトウェア，アプリケーションの専門家の協力：Co-Design
- 既存ソフトウェア資産（e.g. HEC-MW）の効率的利用
- 実際に動いているアプリケーションから機能を切り出す

対象とする離散化手法

局所的，隣接通信中心，疎行列

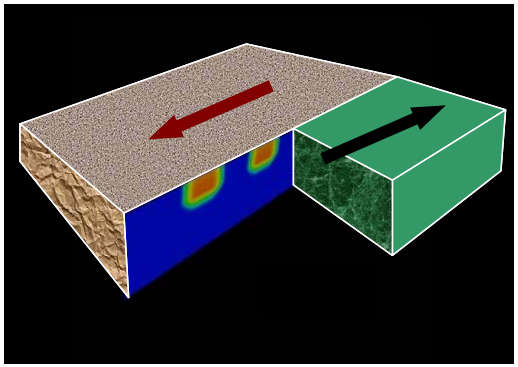


有限要素法
Finite Element Method
FEM

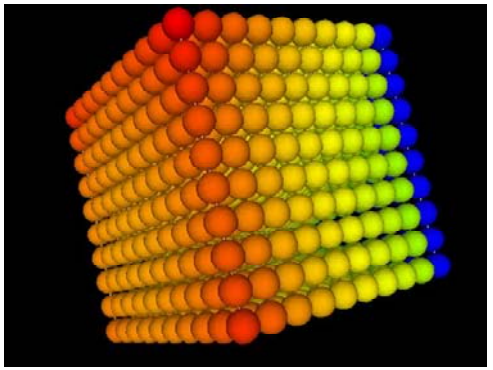


差分法
Finite Difference Method
FDM

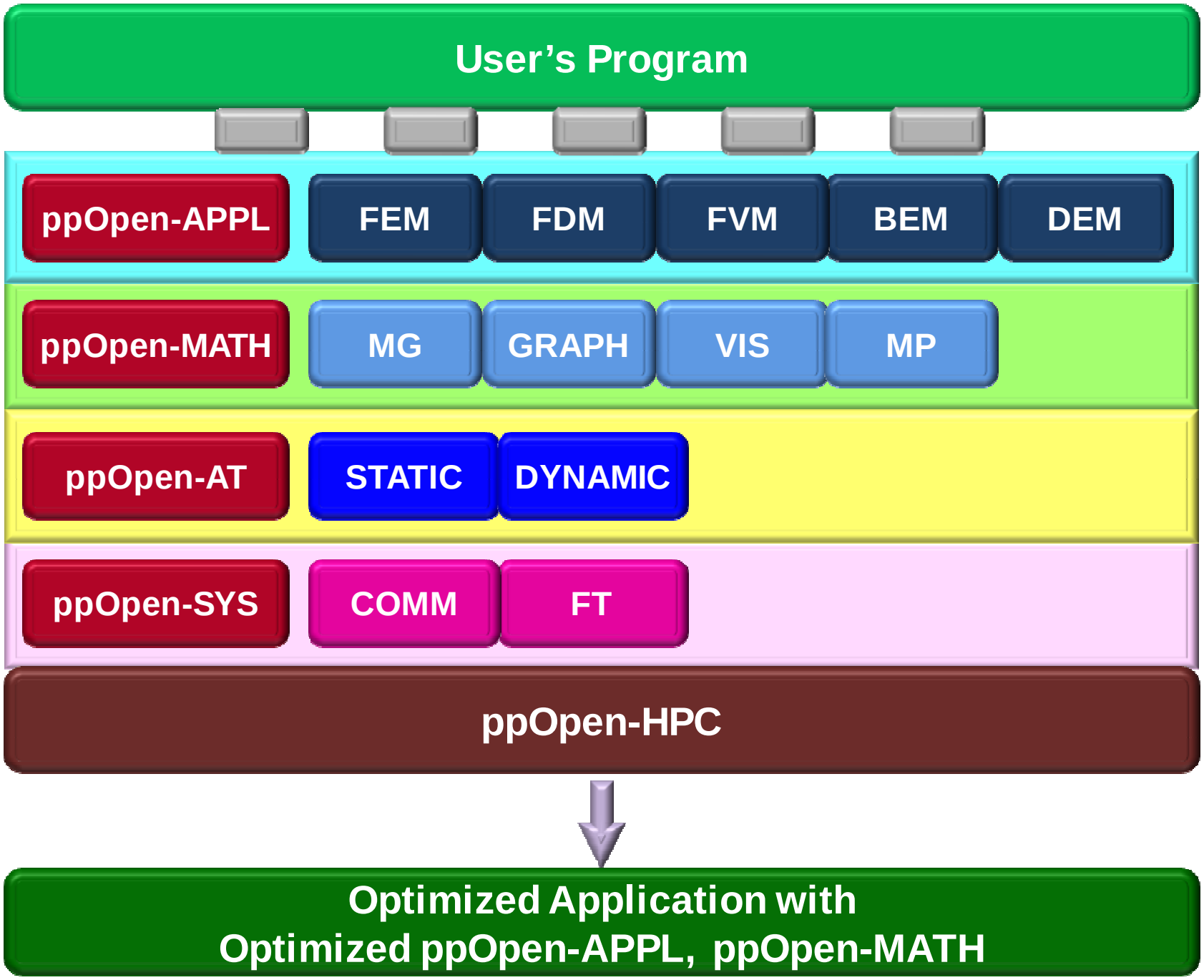
有限体積法
Finite Volume Method
FVM



境界要素法
Boundary Element Method
BEM



個別要素法
Discrete Element Method
DEM


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-HPC：何を作るか（1/2）

- ppOpen-APPL
  - 各手法個別の並列プログラム開発用ライブラリ群またはテンプレート
    - ①共通データ入出力インタフェース
    - ②領域間通信
    - ③係数マトリクス生成
    - ④離散化手法の特性を考慮した前処理付き反復法
    - ⑤適応格子，動的負荷分散
- ppOpen-MATH
  - 各手法に共通の数値演算ライブラリ群
  - 単独で切り出しても使える

ppOpen-HPC：何を作るか（2/2）

- ppOpen-AT
  - 自動チューニング（AT）コンパイラ：Directive Base
    - 最適化ライブラリ群，アプリケーションを自動生成
    - OpenFOAM等オープンソースアプリ，ライブラリも対象
    - Hybrid，CUDA，OpenCL
    - メモリアクセス最適化に主眼
    - CPU～GPU
      - タスク分散
      - 負荷分散
- ppOpen-SYS
  - ノード（アクセラレータ）間通信，耐故障機能に関連するライブラリ群

OpenACC

- C/C++/FORTRAN＋Directive
  - ppOpen-ATと類似
- 最適化の範囲：現行のコンパイラと同じかそれ以下・・・であろう
  - 密行列，構造格子差分法（陽解法）には一定の効果
  - 全てをカバーできるわけではない
- 非構造格子，陰解法，データ依存性がある場合には別途最適化が必要
  - ライブラリ等で対応する必要がある

ppOpen-AT Extension: Loop Splitting

```
!oat$ install LoopSplit region start
!oat$ name MyFDMkernel
    DO K = 1, NZ
    DO J = 1, NY
    DO I = 1, NX
        RL(I)  = LAM (I,J,K)
        RM(I)  = RIG (I,J,K)
        RM2(I) = RM(I) + RM(I)
        RMAXY(I) = 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(I) = 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(I) = 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))
!oat$ SplitPoint
        RLTHETA(I) = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL(I)
        QG(I) = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
!oat$ SplitPoint
        SXX (I,J,K) = ( SXX (I,J,K)  + (RLTHETA(I) + RM2(I)*DXVX(I,J,K))*DT )*QG(I)
        SYY (I,J,K) = ( SYY (I,J,K)  + (RLTHETA(I) + RM2(I)*DYVY(I,J,K))*DT )*QG(I)
        SZZ (I,J,K) = ( SZZ (I,J,K)  + (RLTHETA(I) + RM2(I)*DZVZ(I,J,K))*DT )*QG(I)
!oat$ SplitPoint
        SXY (I,J,K) = ( SXY (I,J,K)  + (RMAXY(I)*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG(I)
        SXZ (I,J,K) = ( SXZ (I,J,K)  + (RMAXZ(I)*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG(I)
        SYZ (I,J,K) = ( SYZ (I,J,K)  + (RMAYZ(I)*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG(I)
    END DO
    END DO
    END DO
!oat$ install LoopSplit region end
```

ppOpen-AT Extension: Loop Fusion

```
!oat$ install LoopFusion region start
!oat$ name MyFDMkernel
      DO K = 1, NZ
      DO J = 1, NY
      DO I = 1, NX
          RL(I)  = LAM (I,J,K)
          RM(I)  = RIG (I,J,K)
          RM2(I) = RM(I) + RM(I)
          RMAXY(I) = 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(I) = 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(I) = 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(I) = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL(I)
          QG(I) = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
          SXX (I,J,K) = ( SXX (I,J,K)  + (RLTHETA(I) + RM2(I)*DXVX(I,J,K))*DT )*QG(I)
          SYY (I,J,K) = ( SYY (I,J,K)  + (RLTHETA(I) + RM2(I)*DYVY(I,J,K))*DT )*QG(I)
          SZZ (I,J,K) = ( SZZ (I,J,K)  + (RLTHETA(I) + RM2(I)*DZVZ(I,J,K))*DT )*QG(I)
          SXY (I,J,K) = ( SXY (I,J,K)  + (RMAXY(I)*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG(I)
          SXZ (I,J,K) = ( SXZ (I,J,K)  + (RMAXZ(I)*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG(I)
          SYZ (I,J,K) = ( SYZ (I,J,K)  + (RMAYZ(I)*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG(I)
      END DO
      END DO
      END DO
!oat$ install LoopFusion region end
```

ppOpen-AT Extension: Loop Splitting & Fusion Selection

```
!oat$ install LoopFusionSplit region start
!oat$ name MyFDMkernel
      DO K = 1, NZ
      DO J = 1, NY
      DO I = 1, NX
          RL(I)  = LAM (I,J,K)
          RM(I)  = RIG (I,J,K)
          RM2(I) = RM(I) + RM(I)
          RMAXY(I) = 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(I) = 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(I) = 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))
!oat$ SplitPoint
          RLTHETA(I) = (DXVX(I,J,K)+DYVY(I,J,K)+DZVZ(I,J,K))*RL(I)
          QG(I) = ABSX(I)*ABSY(J)*ABSZ(K)*Q(I,J,K)
!oat$ SplitPoint
          SXX (I,J,K) = ( SXX (I,J,K)  + (RLTHETA(I) + RM2(I)*DXVX(I,J,K))*DT )*QG(I)
          SYY (I,J,K) = ( SYY (I,J,K)  + (RLTHETA(I) + RM2(I)*DYVY(I,J,K))*DT )*QG(I)
          SZZ (I,J,K) = ( SZZ (I,J,K)  + (RLTHETA(I) + RM2(I)*DZVZ(I,J,K))*DT )*QG(I)
!oat$ SplitPoint
          SXY (I,J,K) = ( SXY (I,J,K)  + (RMAXY(I)*(DXVY(I,J,K)+DYVX(I,J,K)))*DT )*QG(I)
          SXZ (I,J,K) = ( SXZ (I,J,K)  + (RMAXZ(I)*(DXVZ(I,J,K)+DZVX(I,J,K)))*DT )*QG(I)
          SYZ (I,J,K) = ( SYZ (I,J,K)  + (RMAYZ(I)*(DYVZ(I,J,K)+DZVY(I,J,K)))*DT )*QG(I)
      END DO
      END DO
      END DO
!oat$ install LoopFusion Split region  end
```

マイルストーン＝動くソフトウェアの公開

	公開内容	趣旨，目標	ターゲットマシン	開発環境
第１回 H24秋	ppOpen-APPL プロトタイプ（適応格子・動的負荷分散を除く），領域分割ユーティリティ，ppOpen-MATH/VIS・ppOpen-AT/STATIC プロトタイプ	ppOpen-APPL ， ppOpen-ATが様々なマルチコアクラスタ，GPUクラスタで稼働することを検証する。大学（学部，大学院）講義への導入，講習会開催を継続的に実施する。	東大T2K 東大FX10 京コンピュータ GPUクラスタ	東大T2K 東大FX10 GPUクラスタ（東大） 京コンピュータ
第２回 H25秋	ppOpen-HPC（ ppOpen-AT ／DYNAMIC を除く）	ppOpen-HPC の各機能（ppOpen-AT／DYNAMICを除く）をこの時期までに一通り完成する。実アプリケーションへ適用することにより更なる改良を図る。	東大FX10 京コンピュータ GPUクラスタ	東大FX10 GPUクラスタ（東大） 京コンピュータ
第３回 H26秋	ポストペタスケースシステム向けppOpen-HPCプロトタイプ	ターゲットとするポストペタスケールシステム向けのppOpen-HPCをマシン稼働開始と同時に利用可能とする。	東大ppシステム	東大FX10 小型メニーコアクラスタ 京コンピュータ
第４回 H27秋	ポストペタスケースシステム向けppOpen-HPC完成版	第3回で公開したppOpen-HPCを実アプリケーションへ適用することにより，更に改良，最適化した最終バージョンを公開する。	東大ppシステム	東大ppシステム 東大次期システム 小型メニーコアクラスタ 京コンピュータ

ppOpen-HPC：いくつかのチャレンジ

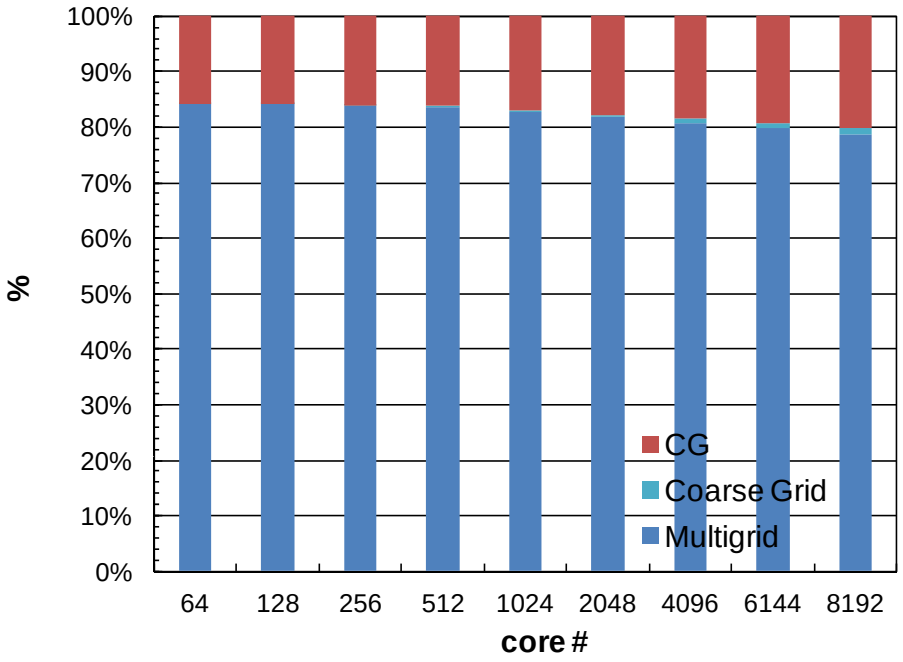
- レガシーコードへの対応
  - FEMコードは基本的にO（10^4）プロセスまではスケールするはず
    - 衝突解析除く
  - これまでの試み（HPC-MW等）：余り使われなかった原因
- オープンソースアプリ・ライブラリのppOepn-AT適用
- マルチコア・メニーコアクラスタ環境におけるロバストな前処理付反復法
  - ILU，IC系：データ依存性
  - NUMAアーキテクチャ： First Touch，Sequential Reordering
- ＋ Scalableな前処理付反復法
  - Parallel Multigrid
    - 粗いレベルにおける通信削減
  - MPI_Allreduce
- 陽解法＋差分法＋非同期通信・・・解けない問題がある

Weak Scaling: MGCG

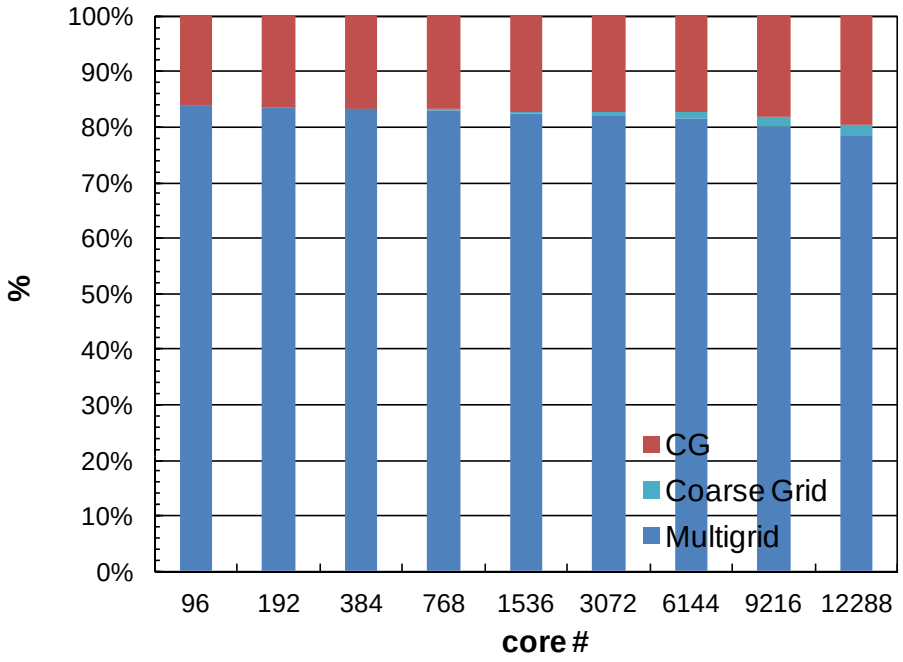
64^3cells/core, up to 512 nodes(2.05×10^9 cells)

T2K (Tokyo): 8,192 cores, Cray-XE6 (Hopper): 12,288 cores

T2K (Tokyo): HB 4x4



Cray XE6 (Hopper@LBNL): HB 6x4



将来展望

- 大規模アプリ開発促進，計算科学の進歩に貢献
- 商品化，カスタマイズ
- エクサスケールへ向けてのCo-Design体制推進
- 人材育成
  - 共通教材としての使用
  - e-Learning環境整備

- 産業界におけるサイエンス（エンジニアリング）ロードマップ

東大センターと企業利用

- 現行（T2K）：社会貢献
  - 2008年10月～
  - 基本的には資源提供型
    - 8ノード（1.18 TFLOPS）：240万円／年（大学・公共機関：85万円）
  - 講習会：産業界からの参加者も受講・お試しアカウント使用可能
- 将来（FX10）：社会貢献＋α
  - In-Houseコード，オープンソース（例：OpenFOAM，生研アプリはプリインストール）中心，商用アプリはセンターでは提供しない
    - 12ノード（2.84 TFLOPS）：140万円／年（大学・公共機関：50万円）（予定）
  - 共同研究型利用を増やして行く予定
    - センターと共同研究を実施すれば負担金は大学・公共機関並
    - ppOpen-HPC利用・高度化　→　センターサービスへのフィードバック
    - 商用アプリケーションもターゲットとなり得る
  - 講習会：継続
  - オープンCAE学会（理事：大島聡史）