# Introduction to Parallel Programming for Multicore/Manycore Clusters

## General Introduction

*Invitation to Supercomputing*

Kengo Nakajima
Information Technology Center
The University of Tokyo

Takahiro Katagiri
Information Technology Center
Nagoya University

# Motivation for Parallel Computing (and this class)

- Large-scale parallel computer enables fast computing in large-scale scientific simulations with detailed models. Computational science develops new frontiers of science and engineering.
- Why parallel computing ?
  - faster & larger
  - "larger" is more important from the view point of "new frontiers of science & engineering", but "faster" is also important.
  - + more complicated
  - Ideal: Scalable
    - Solving $N^x$ scale problem using $N^x$ computational resources during same computation time (weak scaling)
    - Solving a fix-sized problem using $N^x$ computational resources in 1/N computation time (strong scaling)

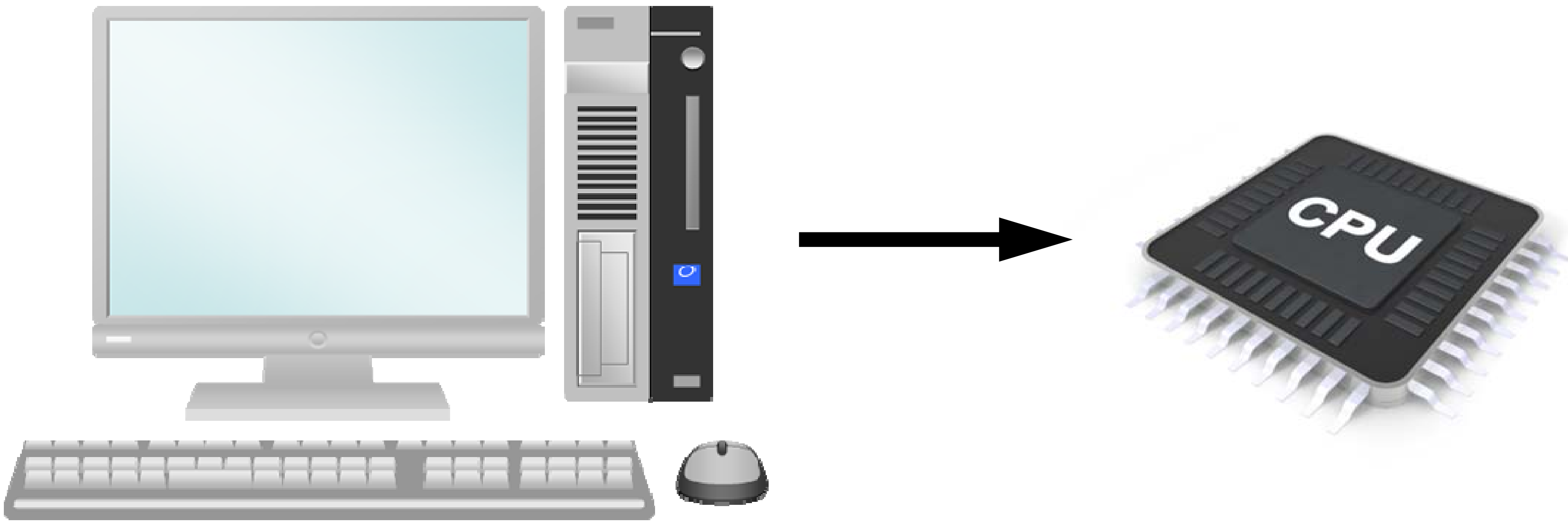
# Scientific Computing = SMASH

- **Science**
- **Modeling**
- **Algorithm**
- **Software**
- **Hardware**

- You have to learn many things.
- Collaboration (or Co-Design) will be important for future career of each of you, as a scientist and/or an engineer.
  - You have to communicate with people with different backgrounds.
  - It is more difficult than communicating with foreign scientists from same area.
- (Q): Computer Science, Computational Science, or Numerical Algorithms ?

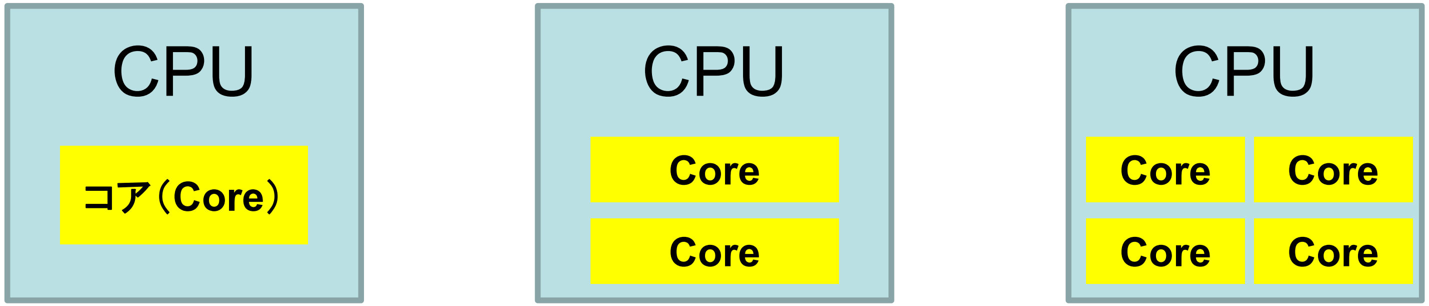
- **Supercomputers and Computational Science**
- Overview of the Class

# Computer & CPU




- Central Processing Unit（中央処理装置）: CPU
- CPU's used in PC and Supercomputers are based on same architecture
- GHz: Clock Rate
  - Frequency: Number of operations by CPU per second
    - GHz -> $10^9$ operations/sec
  - Simultaneous 4-8 instructions per clock

# Multicore CPU




Single Core
1 cores/CPU

Dual Core
2 cores/CPU

Quad Core
4 cores/CPU

- Core= Central part of CPU
- Multicore CPU's with 4-8 cores are popular
  - Low Power

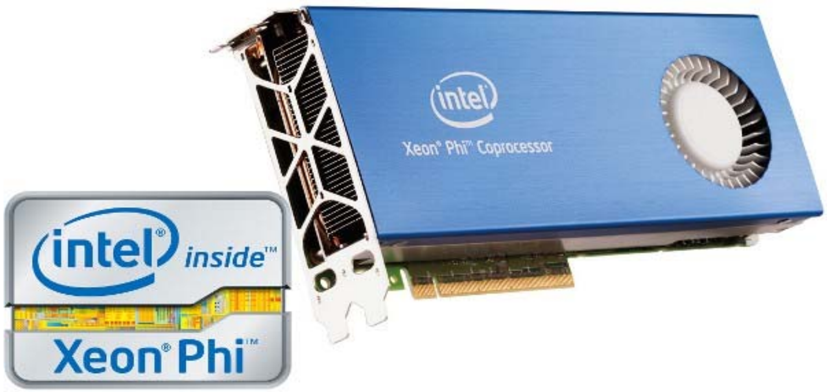



- GPU: Manycore
  - $O(10^1)$-$O(10^2)$ cores
- More and more cores
  - Parallel computing
- Oakleaf-FX at University of Tokyo: 16 cores
  - SPARC64™ IXfx

# GPU/Manycores

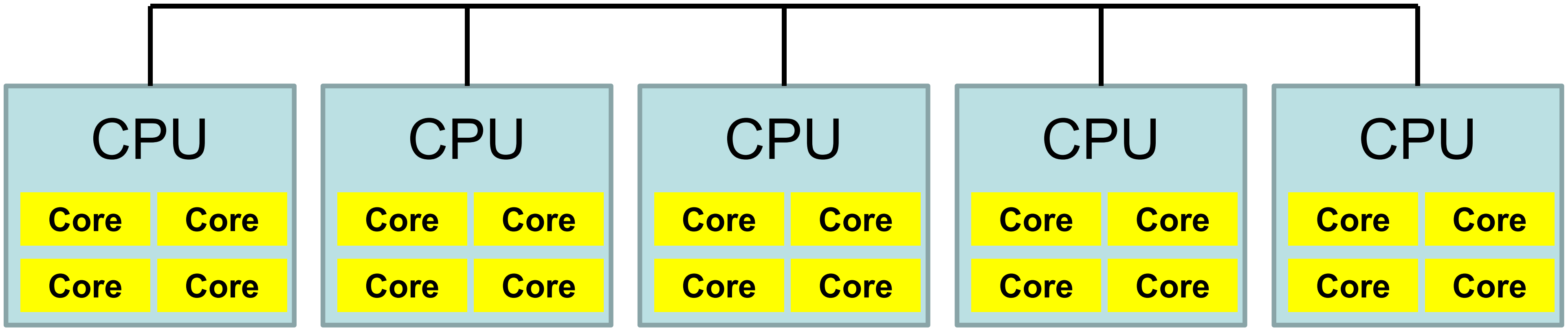
- GPU：Graphic Processing Unit
  - GPGPU: General Purpose GPU
  - $O(10^2)$ cores
  - High Memory Bandwidth
  - Cheap
  - NO stand-alone operations
    - Host CPU needed
  - Programming: CUDA, OpenACC

- Intel Xeon/Phi: Manycore CPU
  - >60 cores
  - High Memory Bandwidth
  - Unix, Fortran, C compiler
  - Currently, host CPU needed
    - Stand-alone: Knights Landing

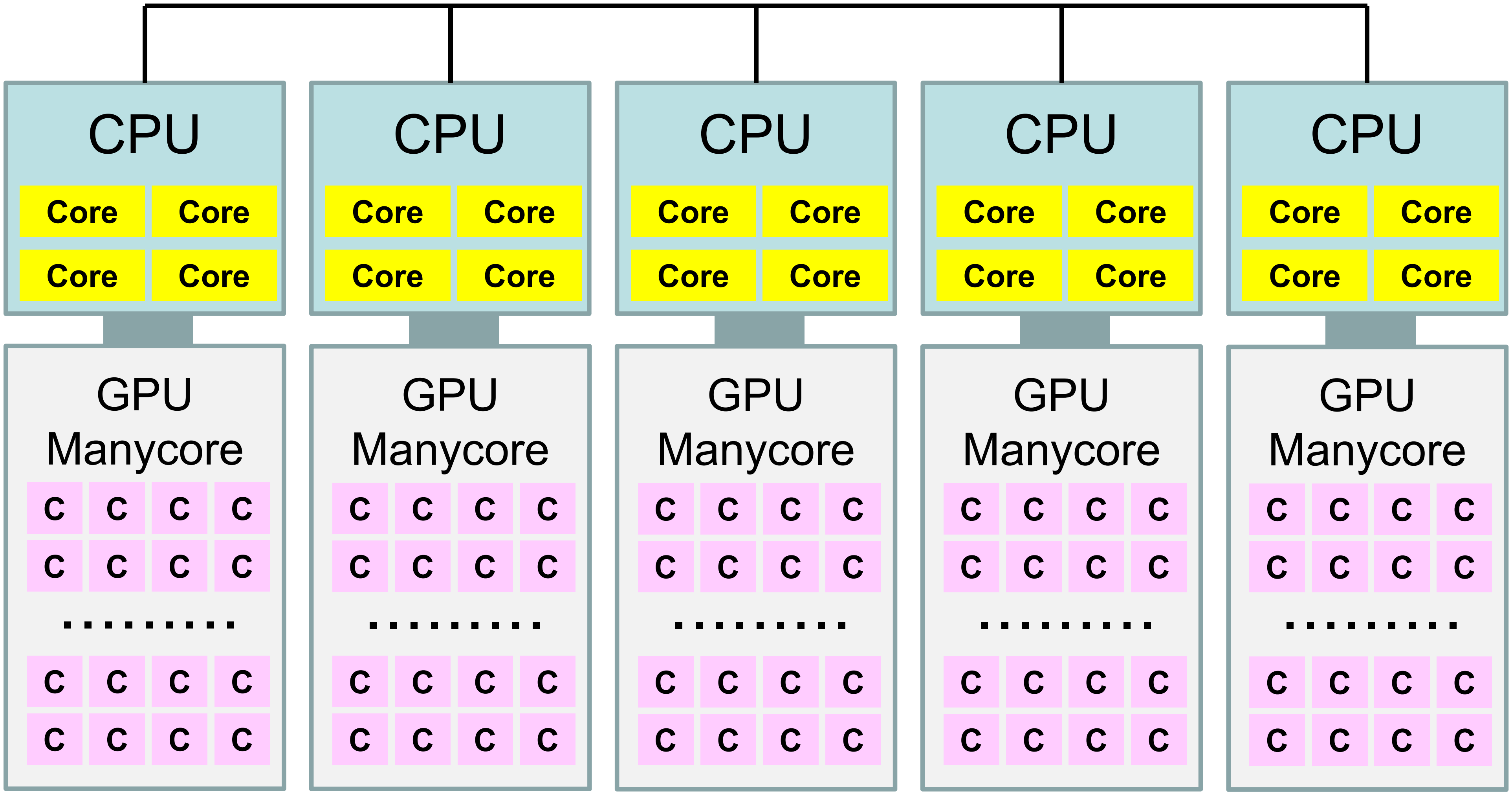


# Parallel Supercomputers

Multicore CPU's are connected through network



# Supercomputers with Heterogeneous/Hybrid Nodes



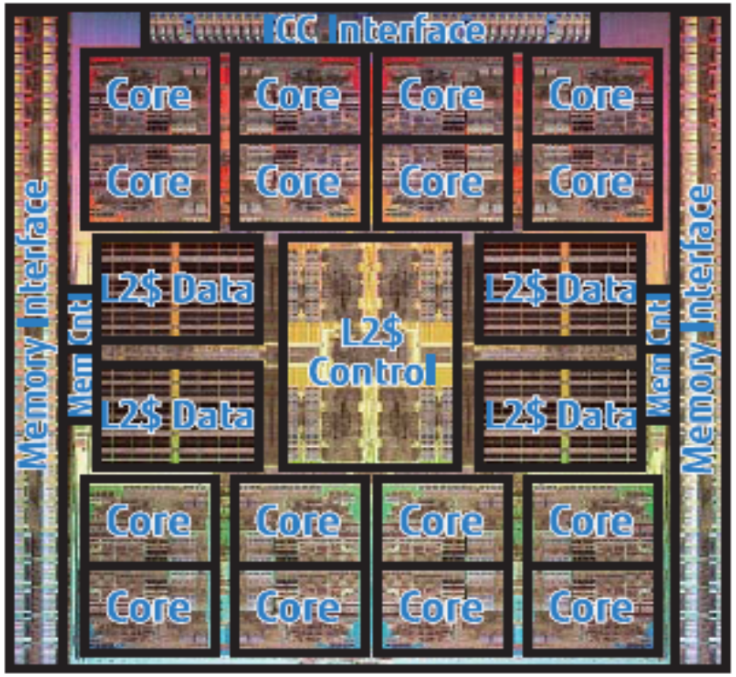
# Performance of Supercomputers

- Performance of CPU: Clock Rate
- FLOPS (Floating Point Operations per Second)
  - Real Number
- Recent Multicore CPU
  - 4-8 FLOPS per Clock
  - (e.g.) Peak performance of a core with 3GHz
    - $3 \times 10^9 \times 4$(or 8)=12(or 24)$\times 10^9$ FLOPS=12(or 24)GFLOPS

    - $10^6$ FLOPS= 1 Mega FLOPS = 1 MFLOPS
    - $10^9$ FLOPS= 1 Giga FLOPS = 1 GFLOPS
    - $10^{12}$ FLOPS= 1 Tera FLOPS = 1 TFLOPS
    - $10^{15}$ FLOPS= 1 Peta FLOPS = 1 PFLOPS
    - $10^{18}$ FLOPS= 1 Exa FLOPS = 1 EFLOPS

# Peak Performance of Oakleaf-FX

## Fujitsu PRIMEHPC FX10 at U.Tokyo



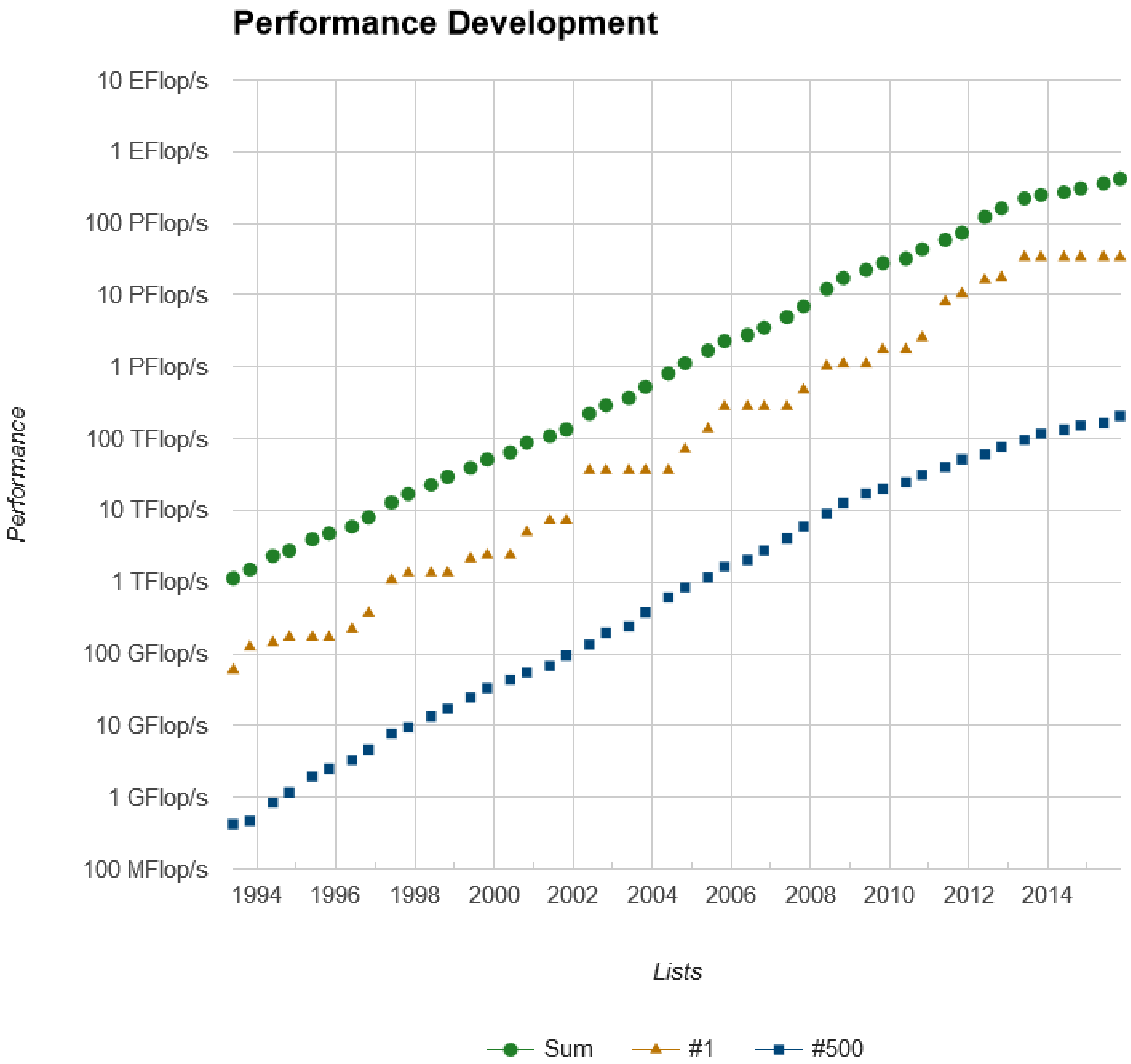



- 1.848 GHz
- 8 FLOP operations per Clock
- Peak Performance (1 core)
  - 1.848×8= 14.78 GFLOPS
- Peak Performance (1 node/16 cores)
  - 236.5 GFLOPS
- **Peak Performance of Entire Performance**
  - **4,800 nodes, 76,800 cores**
  - **1.13 PFLOPS**

# TOP 500 List

http://www.top500.org/

- Ranking list of supercomputers in the world
- Performance (FLOPS rate) is measured by "Linpack" which solves large-scale linear equations.
  - Since 1993
  - Updated twice a year (International Conferences in June and November)
- Linpack
  - iPhone version is available




- PFLOPS: Peta (=$10^{15}$) Floating OPerations per Sec.
- Exa-FLOPS (=$10^{18}$) will be attained after 2020

http://www.top500.org/

# 48th TOP500 List (November, 2016)

| | Site | Computer/Year Vendor | Cores | $R_{max}$ (TFLOPS) | $R_{peak}$ (TFLOPS) | Power (kW) |
|---|---|---|---|---|---|---|
| 1 | National Supercomputing Center in Wuxi, China | **Sunway TaihuLight** , Sunway MPP, Sunway SW26010 260C 1.45GHz, 2016 NRCPC | 10,649,600 | 93,015 (= 93.0 PF) | 125,436 | 15,371 |
| 2 | National Supercomputing Center in Tianjin, China | **Tianhe-2**, Intel Xeon E5-2692, TH Express-2, Xeon Phi, 2013 NUDT | 3,120,000 | 33,863 (= 33.9 PF) | 54,902 | 17,808 |
| 3 | Oak Ridge National Laboratory, USA | **Titan** Cray XK7/NVIDIA K20x, 2012 Cray | 560,640 | 17,590 | 27,113 | 8,209 |
| 4 | Lawrence Livermore National Laboratory, USA | **Sequoia** BlueGene/Q, 2011 IBM | 1,572,864 | 17,173 | 20,133 | 7,890 |
| 5 | DOE/SC/LBNL/NERSC USA | **Cori**, Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries, 2016 Cray | 632,400 | 14,015 | 27,881 | 3,939 |
| 6 | Joint Center for Advanced High Performance Computing, Japan | **Oakforest-PACS**, PRIMERGY CX600 M1, Intel Xeon Phi Processor 7250 68C 1.4GHz, Intel Omni-Path, 2016 Fujitsu | 557,056 | 13,555 | 24,914 | 2,719 |
| 7 | RIKEN AICS, Japan | **K computer**, SPARC64 VIIIfx , 2011 Fujitsu | 705,024 | 10,510 | 11,280 | 12,660 |
| 8 | Swiss Natl. Supercomputer Center, Switzerland | **Piz Daint** Cray XC30/NVIDIA P100, 2013 Cray | 206,720 | 9,779 | 15,988 | 1,312 |
| 9 | Argonne National Laboratory, USA | **Mira** BlueGene/Q, 2012 IBM | 786,432 | 8,587 | 10,066 | 3,945 |
| 10 | DOE/NNSA/LANL/SNL, USA | **Trinity**, Cray XC40, Xeon E5-2698v3 16C 2.3GHz, 2016 Cray | 301,056 | 8,101 | 11,079 | 4,233 |

$R_{max}$: Performance of Linpack (TFLOPS)
$R_{peak}$: Peak Performance (TFLOPS), Power: kW

http://www.top500.org/

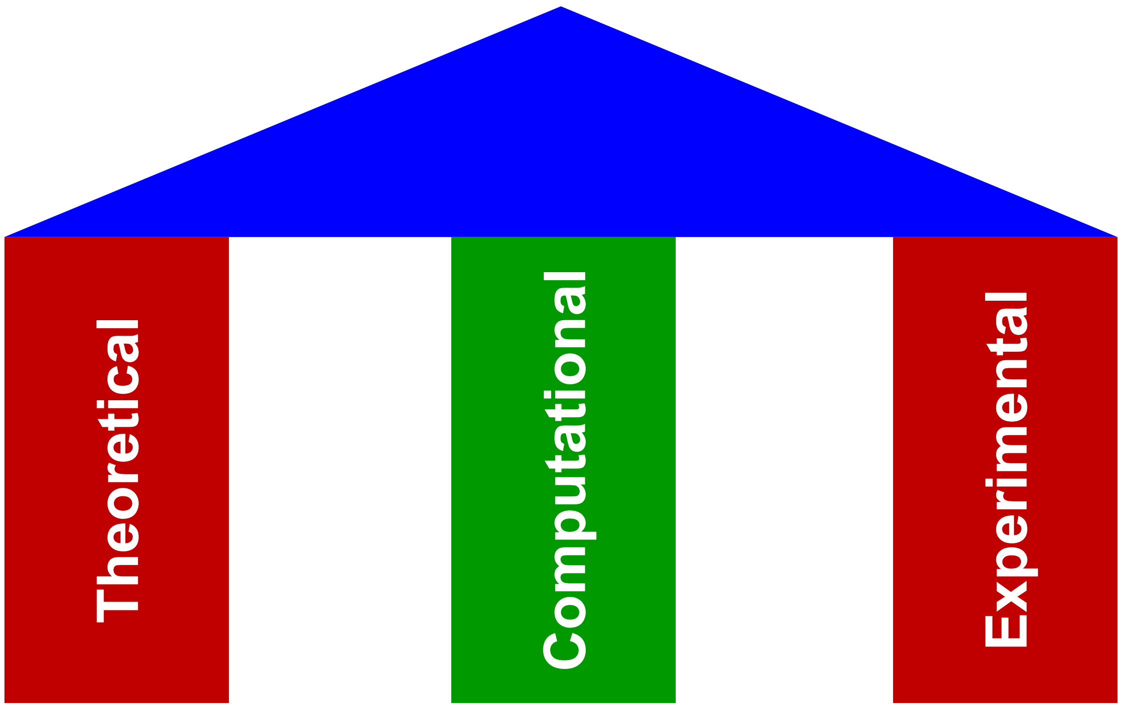
# 48th TOP500 List (November, 2016)

| | Site | Computer/Year Vendor | Cores | $R_{max}$ (TFLOPS) | $R_{peak}$ (TFLOPS) | Power (kW) |
|---|---|---|---|---|---|---|
| 1 | National Supercomputing Center in Wuxi, China | **Sunway TaihuLight** , Sunway MPP, Sunway SW26010 260C 1.45GHz, 2016 NRCPC | 10,649,600 | 93,015 (= 93.0 PF) | 125,436 | 15,371 |
| 2 | National Supercomputing Center in Tianjin, China | **Tianhe-2**, Intel Xeon E5-2692, TH Express-2, Xeon Phi, 2013 NUDT | 3,120,000 | 33,863 (= 33.9 PF) | 54,902 | 17,808 |
| 3 | Oak Ridge National Laboratory, USA | **Titan** Cray XK7/NVIDIA K20x, 2012 Cray | 560,640 | 17,590 | 27,113 | 8,209 |
| 4 | Lawrence Livermore National Laboratory, USA | **Sequoia** BlueGene/Q, 2011 IBM | 1,572,864 | 17,173 | 20,133 | 7,890 |
| 5 | DOE/SC/LBNL/NERSC USA | **Cori**, Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries, 2016 Cray | 632,400 | 14,015 | 27,881 | 3,939 |
| 6 | Joint Center for Advanced High Performance Computing, Japan | **Oakforest-PACS**, PRIMERGY CX600 M1, Intel Xeon Phi Processor 7250 68C 1.4GHz, Intel Omni-Path, 2016 Fujitsu | 557,056 | 13,555 | 24,914 | 2,719 |
| 7 | RIKEN AICS, Japan | **K computer**, SPARC64 VIIIfx , 2011 Fujitsu | 705,024 | 10,510 | 11,280 | 12,660 |
| 8 | Swiss Natl. Supercomputer Center, Switzerland | **Piz Daint** Cray XC30/NVIDIA P100, 2013 Cray | 206,720 | 9,779 | 15,988 | 1,312 |
| 9 | Argonne National Laboratory, USA | **Mira** BlueGene/Q, 2012 IBM | 786,432 | 8,587 | 10,066 | 3,945 |
| 10 | DOE/NNSA/LANL/SNL, USA | **Trinity**, Cray XC40, Xeon E5-2698v3 16C 2.3GHz, 2016 Cray | 301,056 | 8,101 | 11,079 | 4,233 |
| **104** | **ITC/U. Tokyo Japan** | **Oakleaf-FX SPARC64 IXfx, 2012 Fujitsu** | **76800** | **1043** | **1135** | **1177** |

# Computational Science

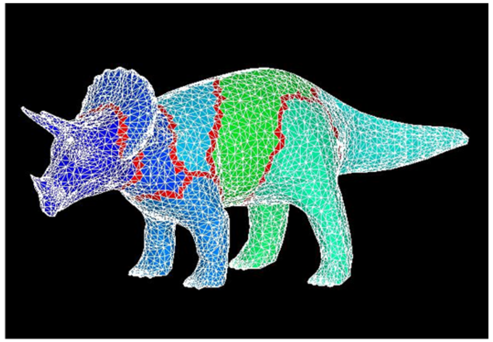
## The 3rd Pillar of Science

- Theoretical & Experimental Science
- Computational Science
  - The 3rd Pillar of Science
  - Simulations using Supercomputers

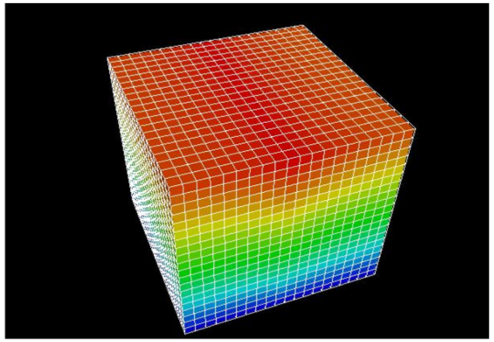


# Methods for Scientific Computing

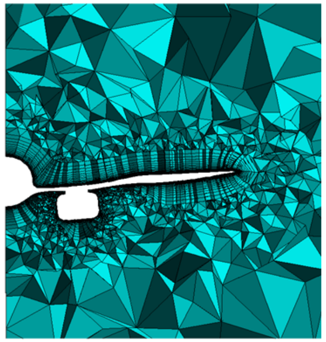
- Numerical solutions of PDE (Partial Diff. Equations)
- Grids, Meshes, Particles
  - Large-Scale Linear Equations
  - Finer meshes provide more accurate solutions



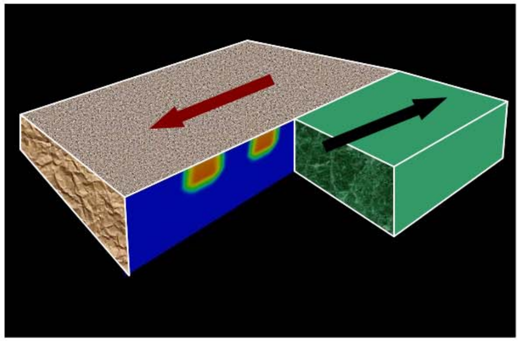
有限要素法
Finite Element Method
FEM



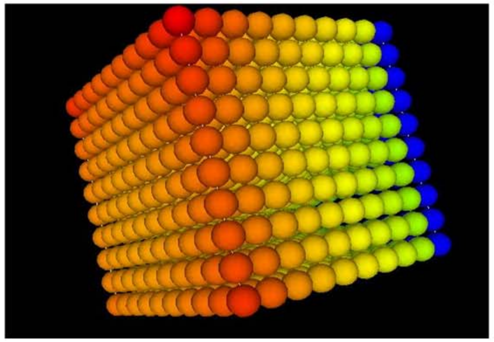
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



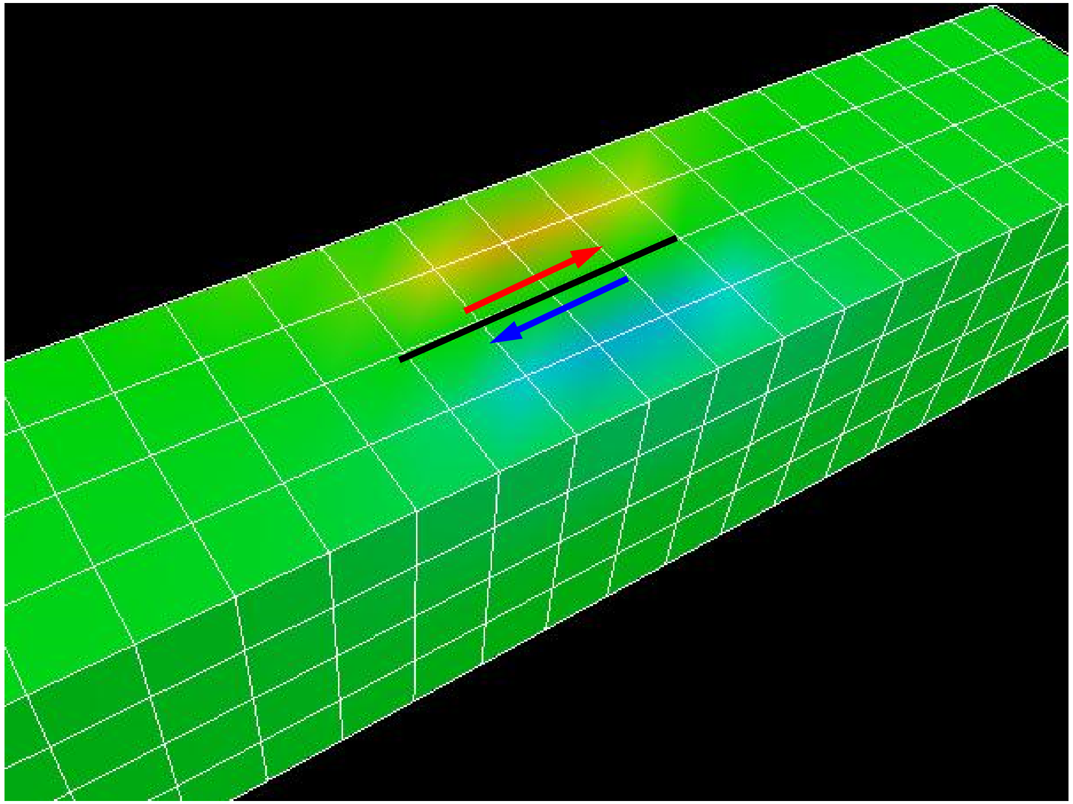
境界要素法
Boundary Element Method
BEM

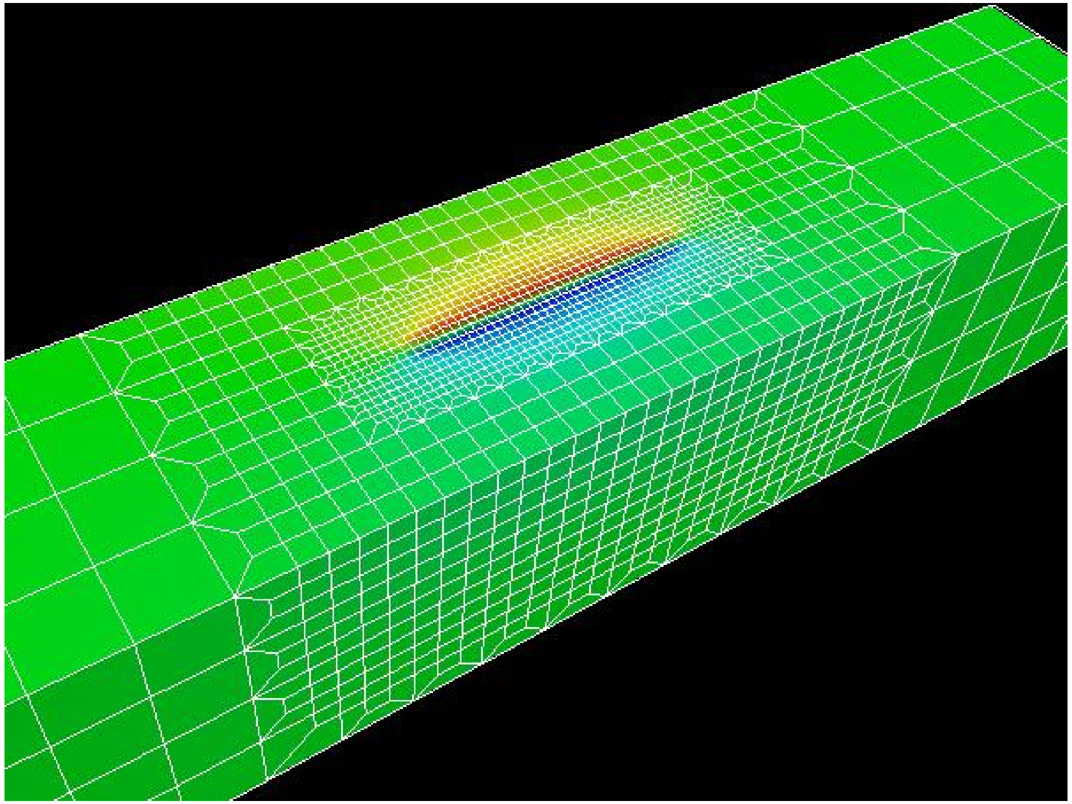


個別要素法
Discrete Element Method
DEM

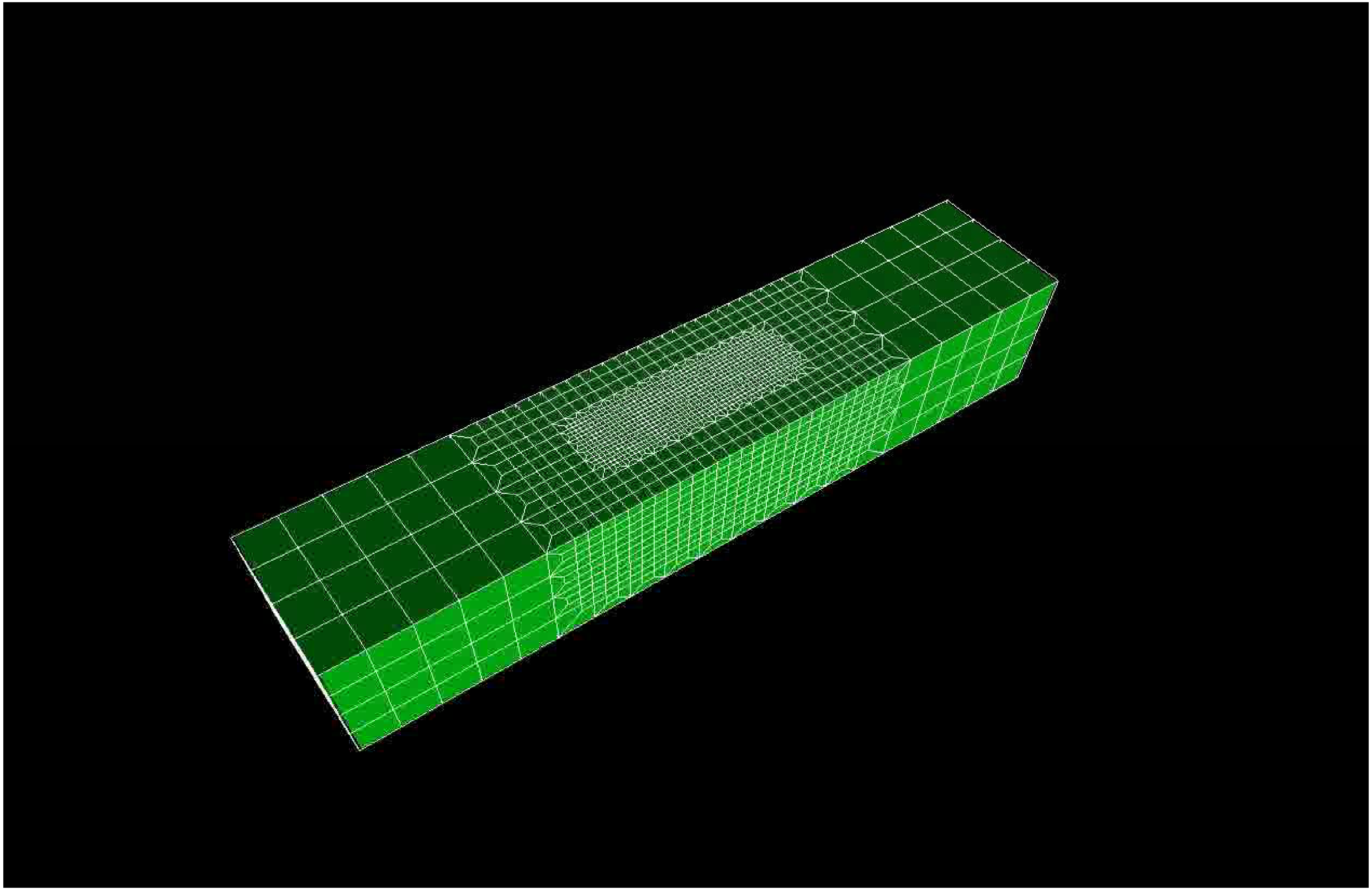
# 3D Simulations for Earthquake Generation Cycle San Andreas Faults, CA, USA

Stress Accumulation at Transcurrent Plate Boundaries

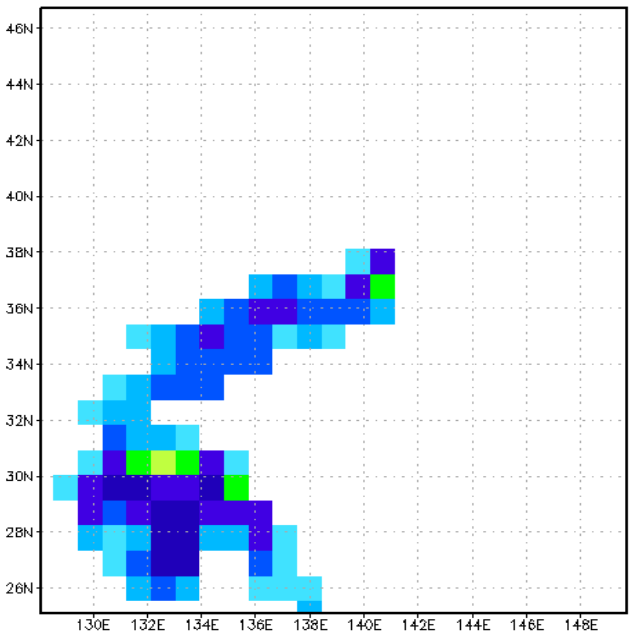




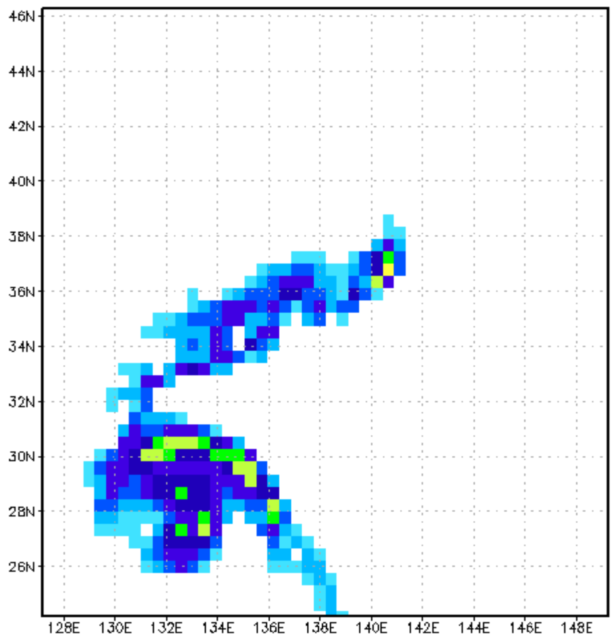
# Adaptive FEM: High-resolution needed at meshes with large deformation (large accumulation)



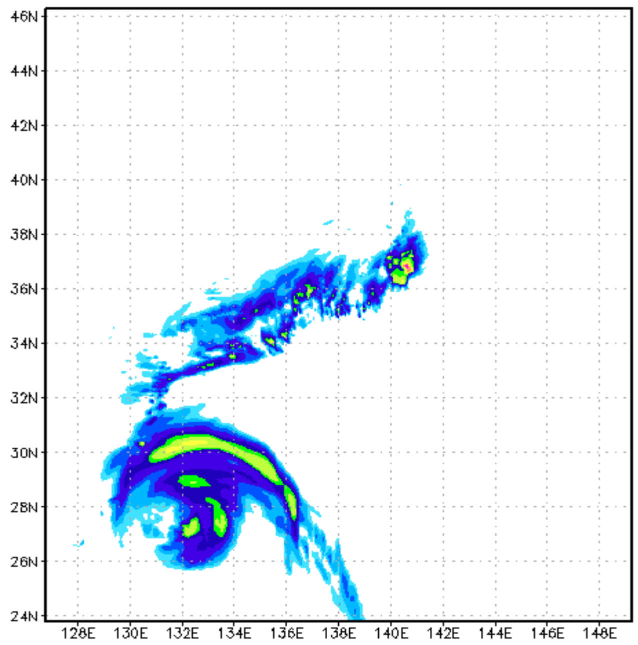
# Typhoon Simulations by FDM
# Effect of Resolution




**Δh=100km**



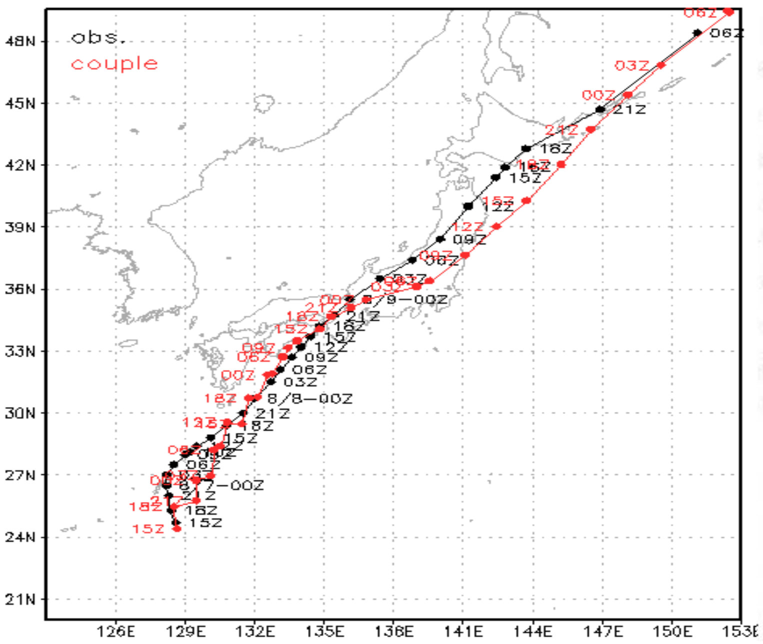

**Δh=50km**

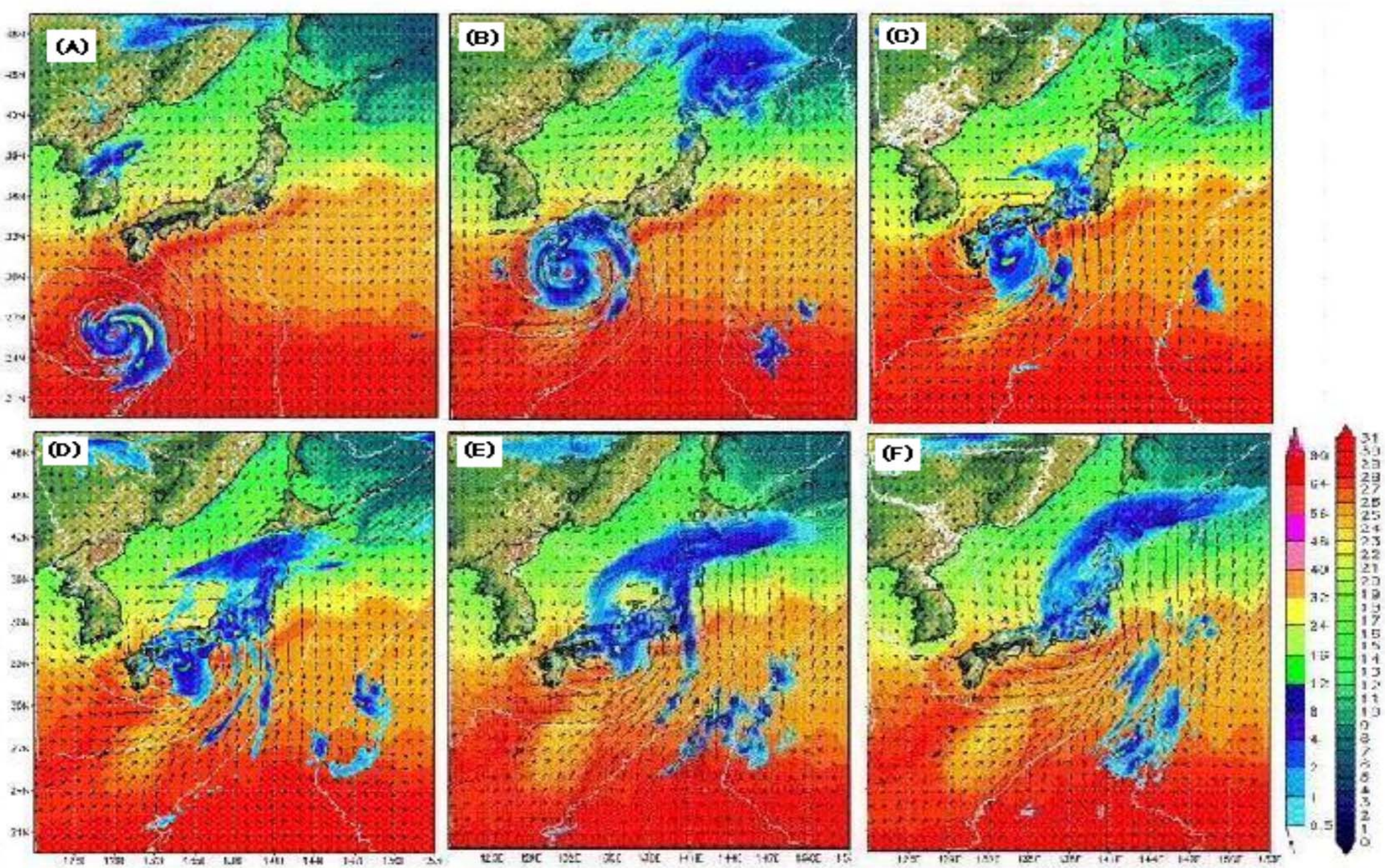



**Δh=5km**

[JAMSTEC]

# Simulation of Typhoon MANGKHUT in 2003 using the Earth Simulator

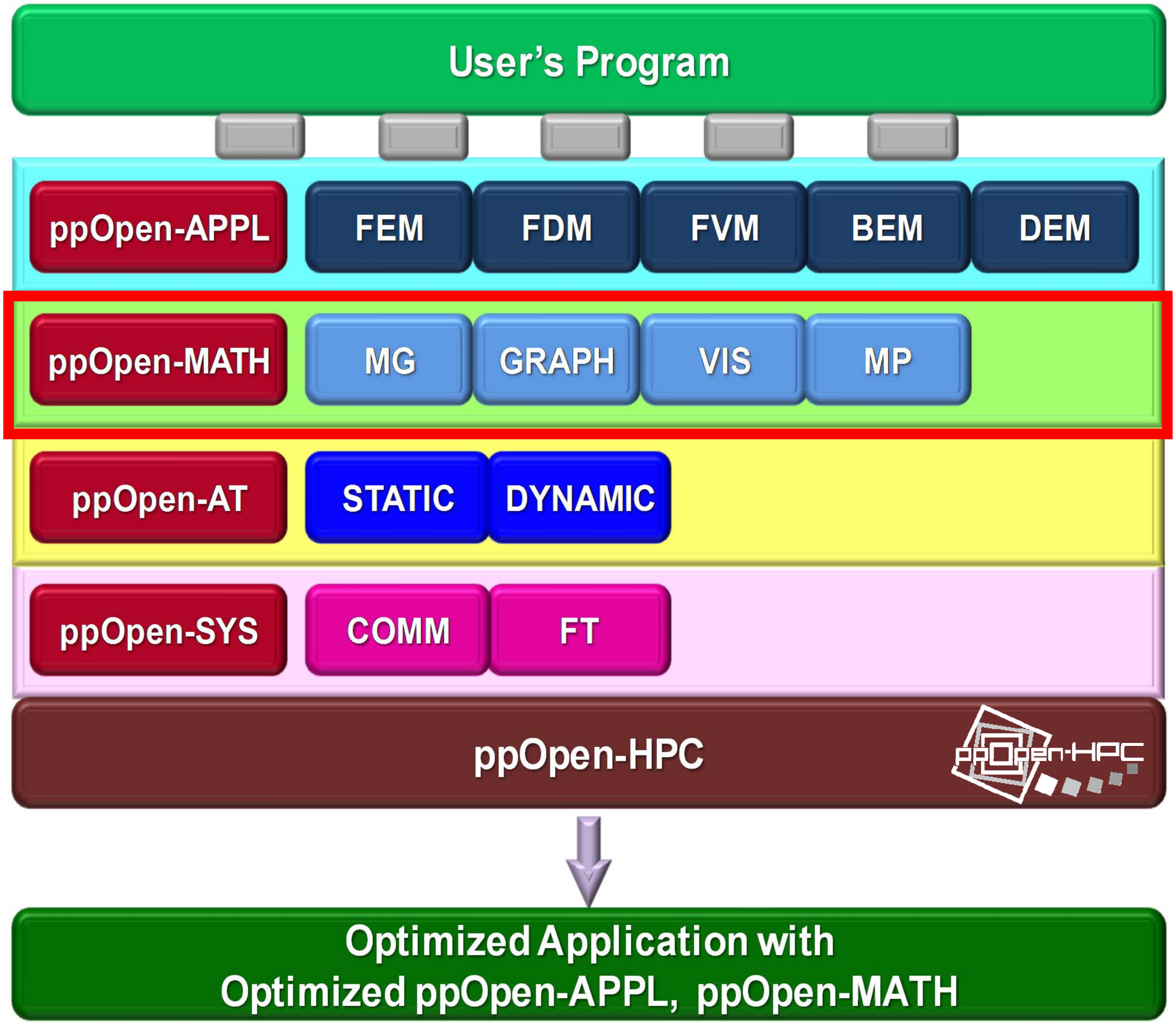



[JAMSTEC]

# ppOpen-HPC

- ppOpen-HPC is an open source infrastructure for development and execution of optimized and reliable simulation code on post-peta-scale (pp) parallel computers based on many-core architectures with automatic tuning (AT), and it consists of various types of libraries, which cover general procedures for scientific computation.
- Software
  - Source Files, English Documents
  - MIT License
  - http://ppopenhpc.cc.u-tokyo.ac.jp/

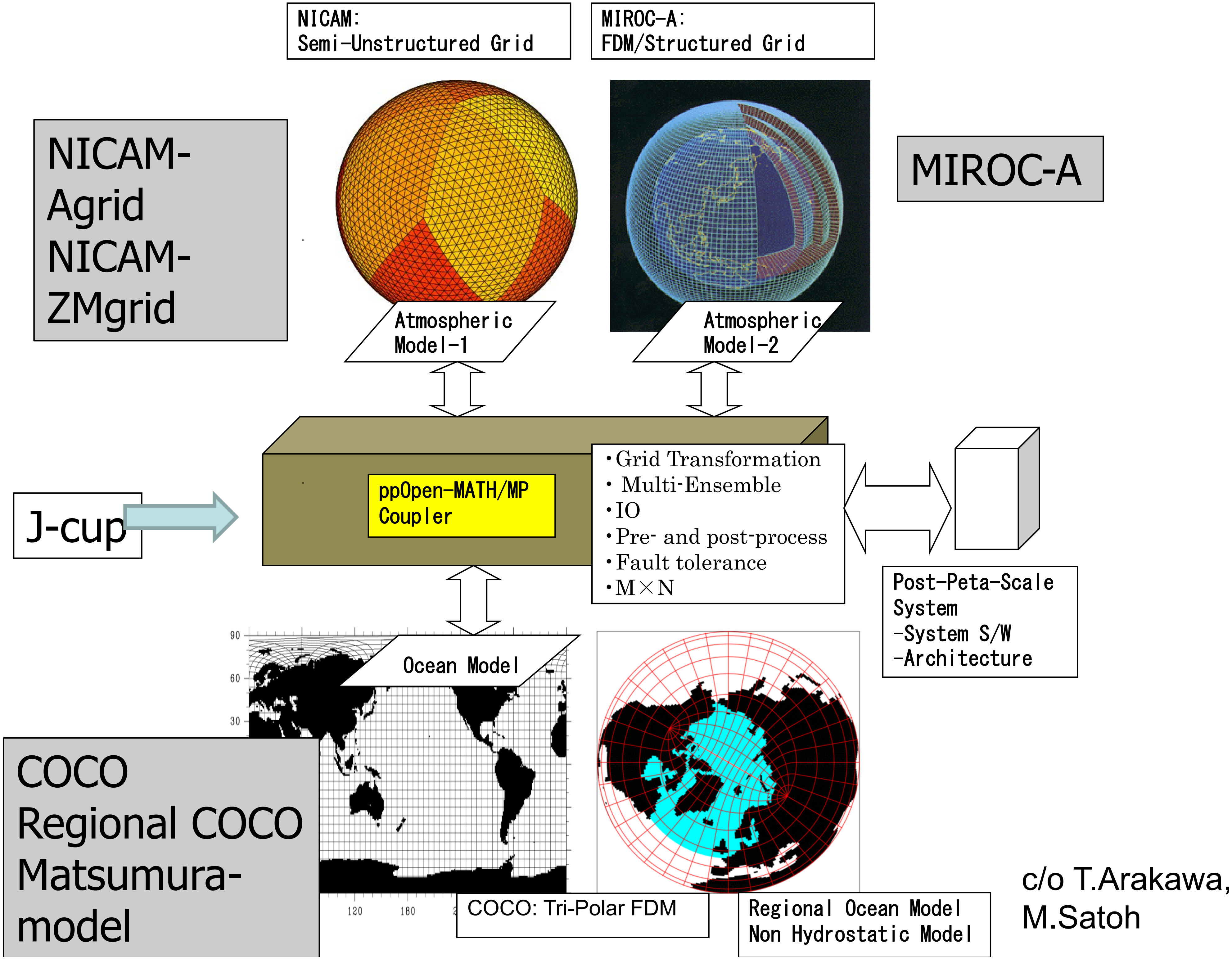

User's Program
ppOpen-APPL
FEM
FDM
FVM
BEM
DEM
ppOpen-MATH
MG
GRAPH
VIS
MP
ppOpen-AT
STATIC
DYNAMIC
ppOpen-SYS
COMM
FT
ppOpen-HPC
ppOpen-HPC
Optimized Application with
Optimized ppOpen-APPL, ppOpen-MATH

# ppOpen-MATH

- A set of common numerical libraries
  - Multigrid solvers (ppOpen-MATH/MG)
  - 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

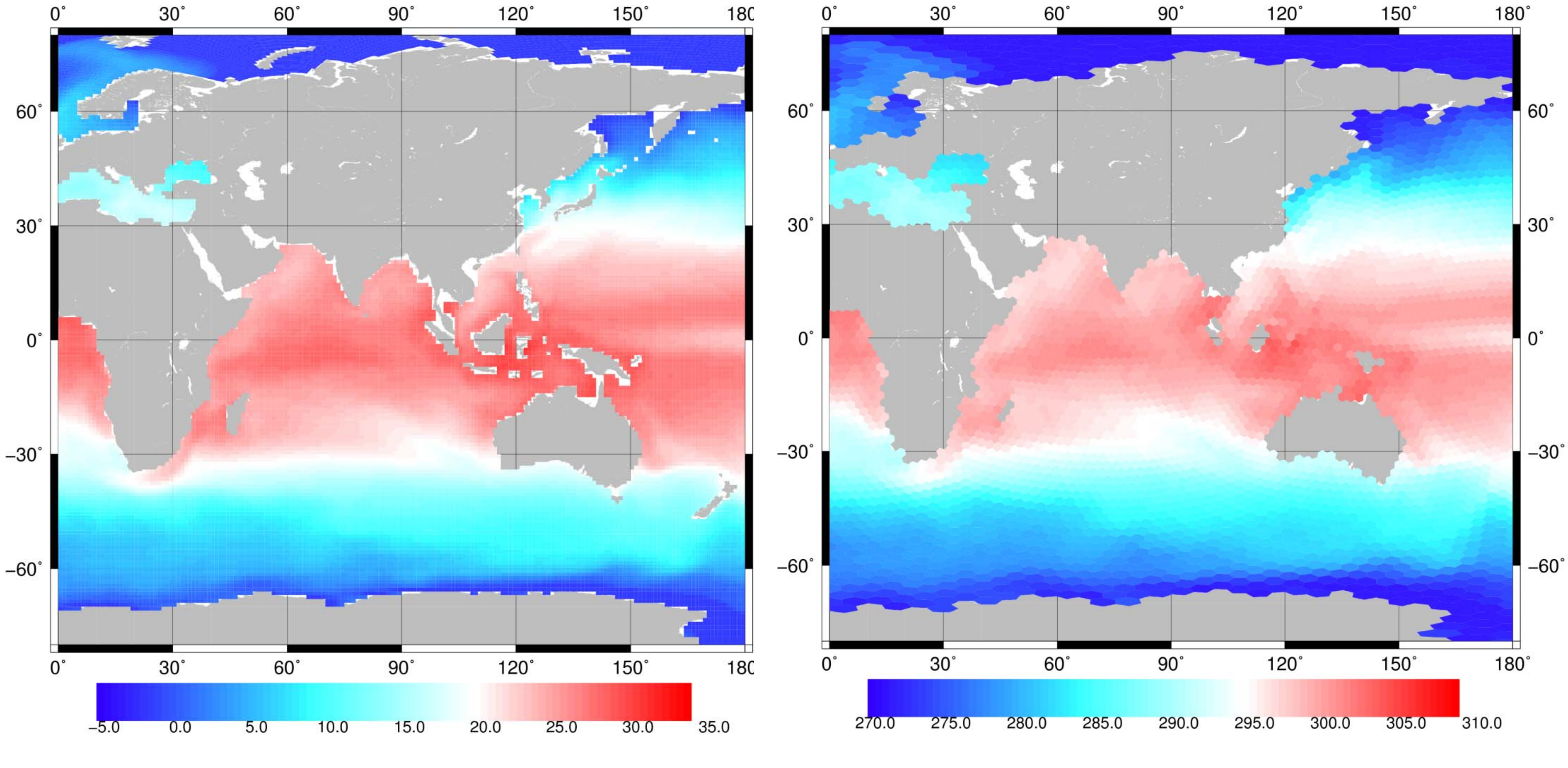
# Motivation for Parallel Computing (again)

- Large-scale parallel computer enables fast computing in large-scale scientific simulations with detailed models. Computational science develops new frontiers of science and engineering.
- Why parallel computing ?
  - faster & larger
  - "larger" is more important from the view point of "new frontiers of science & engineering", but "faster" is also important.
  - + more complicated
    - Coupled Problems (Fluid + Structure)
    - Multiple PDE (Partial Differential Equations)
    - **Actually, most of the real world problems are "coupled" ones**


NICAM:
Semi-Unstructured Grid
MIROC-A:
FDM/Structured Grid
NICAM-
Agrid
NICAM-
ZMgrid
MIROC-A
Atmospheric
Model-1
Atmospheric
Model-2
J-cup
ppOpen-MATH/MP
Coupler
・Grid Transformation
・ Multi-Ensemble
・IO
・Pre- and post-process
・Fault tolerance
・M×N
Post-Peta-Scale
System
-System S/W
-Architecture
90
60
30
Ocean Model
120
180
COCO
Regional COCO
Matsumura-
model
COCO: Tri-Polar FDM
Regional Ocean Model
Non Hydrostatic Model
c/o T.Arakawa,
M.Satoh

c/o T.Arakawa, M.Satoh

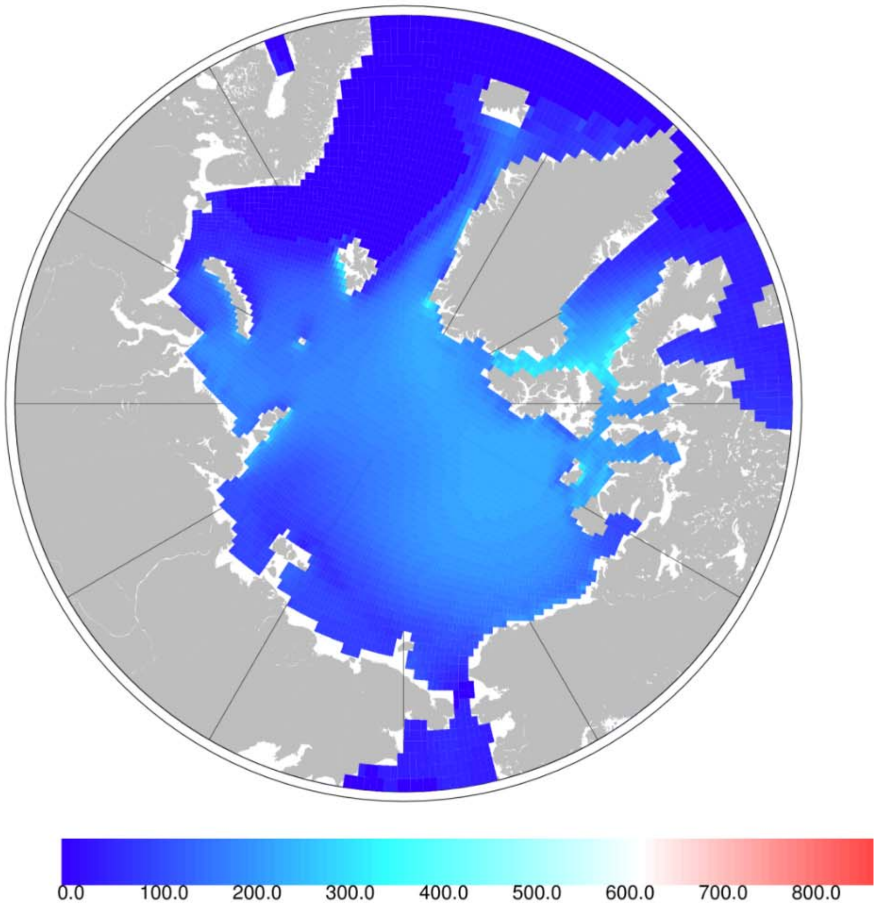
# Sea surface temperature (OSST)

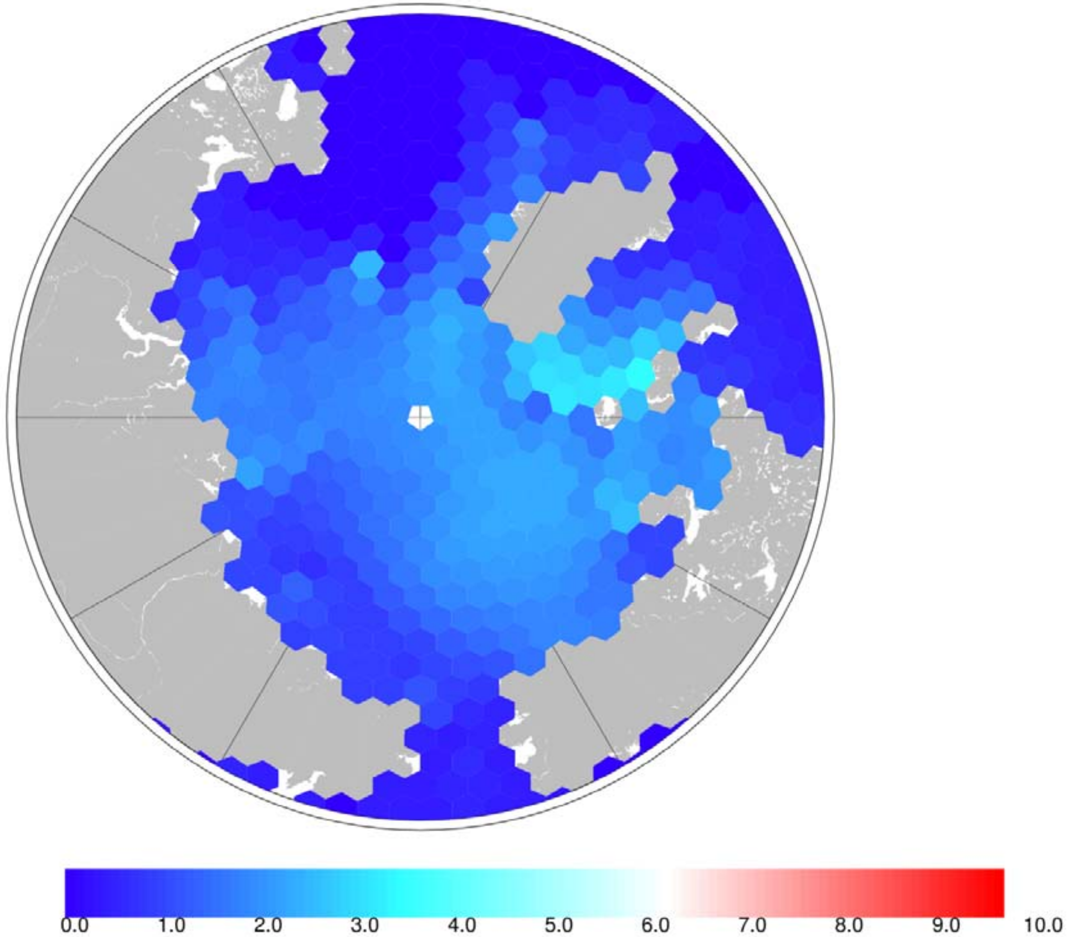



left：COCO (Ocean: Structured), right：NICAM (Atmospheric: Semi-Unst.)

c/o T.Arakawa, M.Satoh

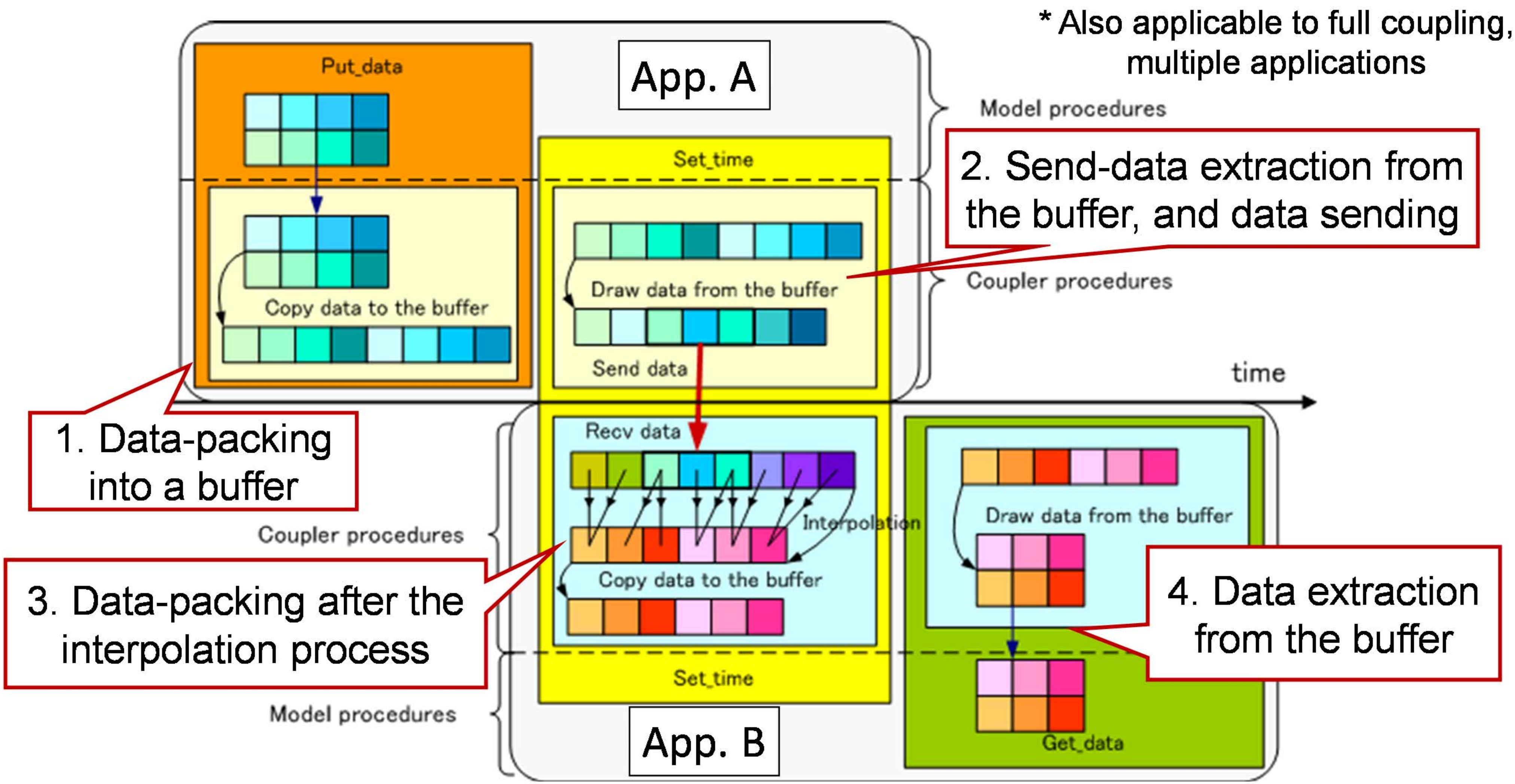
# Thickness of Sea Ice (OHI)




left：COCO (Ocean: Structured), right：NICAM (Atmospheric: Semi-Unst.)

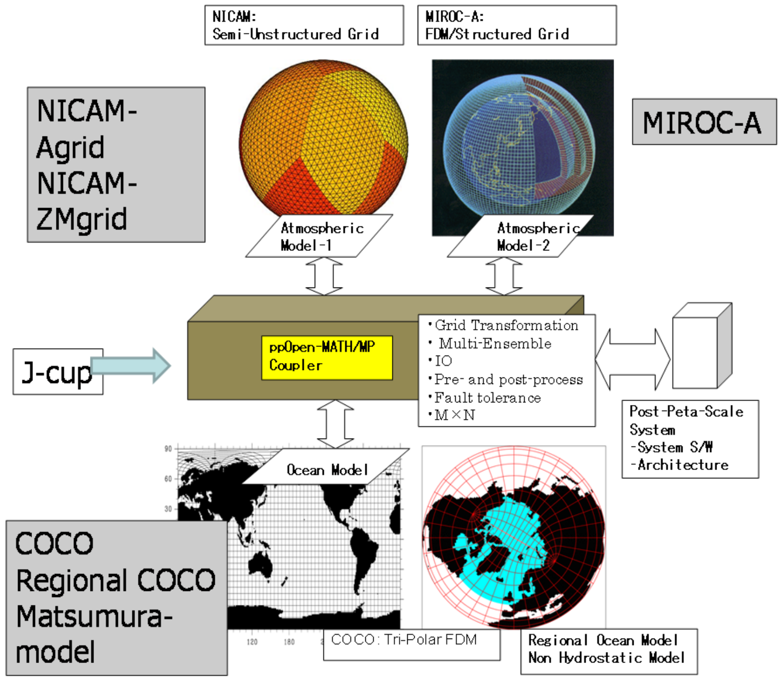
# Dataflow of ppOpen-MATH/MP*

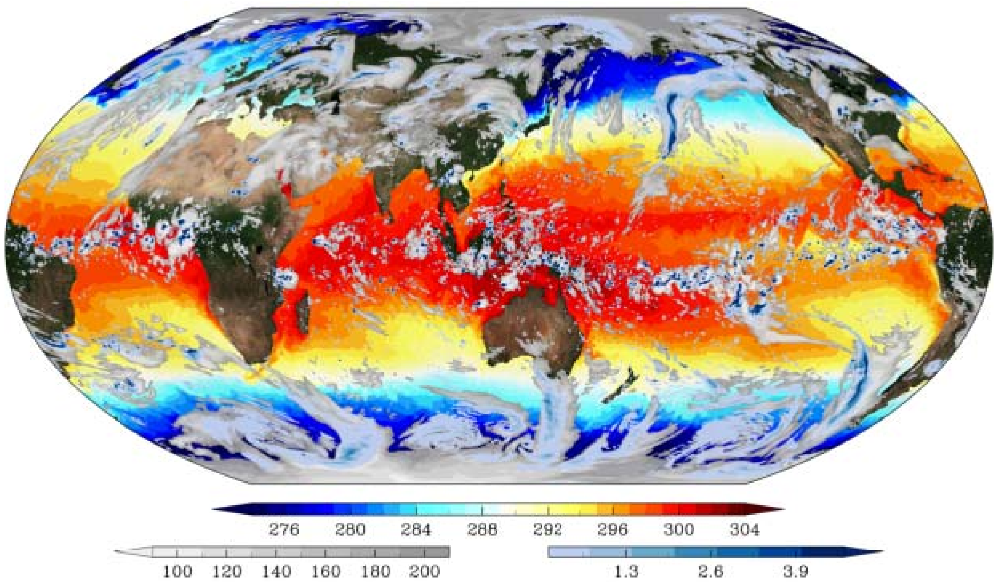


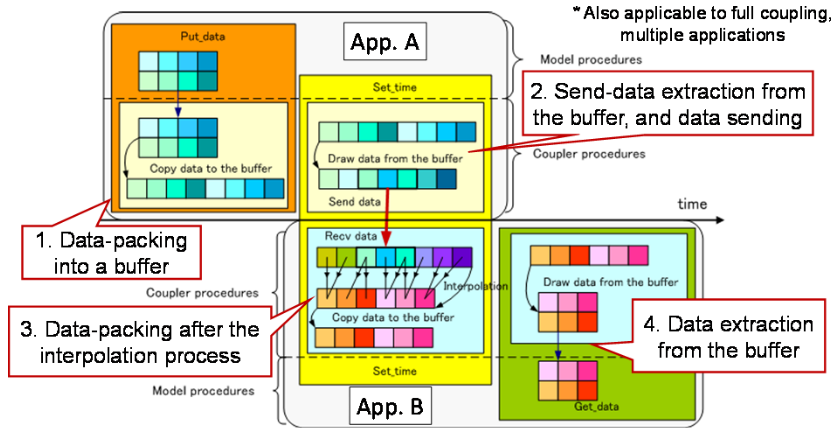
# Atmosphere-Ocean Coupling

## on OFP by NICAM/COCO/ppOpen-MATH/MP

- High-resolution global atmosphere-ocean coupled simulation by NICAM and COCO (Ocean Simulation) through ppOpen-MATH/MP on the K computer is achieved.
  - ppOpen-MATH/MP is a coupling software for the models employing various discretization method.
- An O(km)-mesh NICAM-COCO coupled simulation is planned on the Oakforest-PACS system.
  - A big challenge for optimization of the codes on new Intel Xeon Phi processor
  - New insights for understanding of global climate dynamics

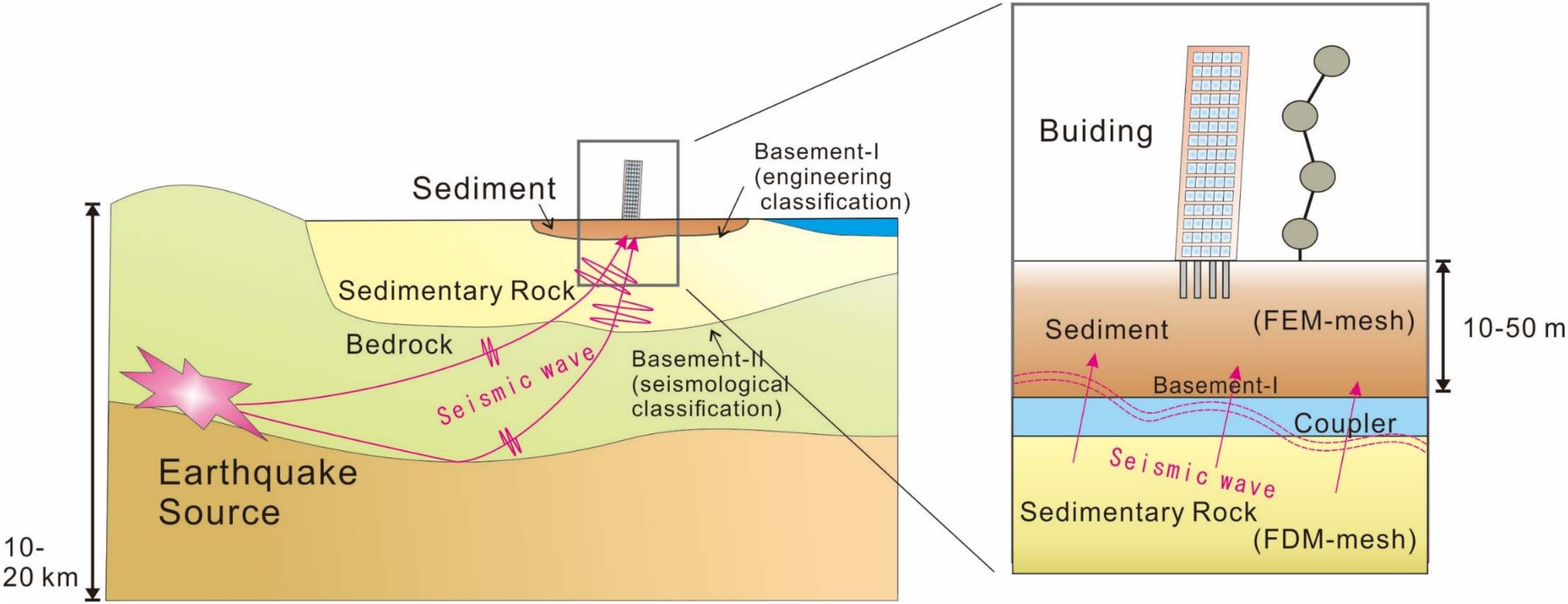


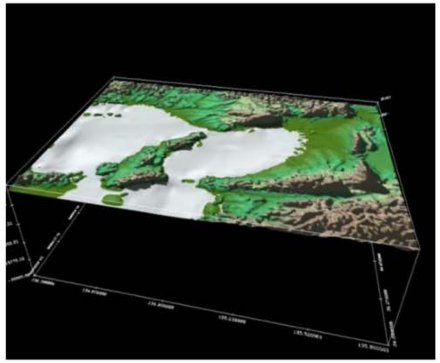

[C/O M. Satoh (AORI/UTokyo)@SC16]

# 1 Coupling Simulation of Seismic Waves and Building Vibrations




The coupling simulation refers to one-way data communication from FDM (seismic wave propagation) to FEM (dynamic structure).

# Numerical Model Description



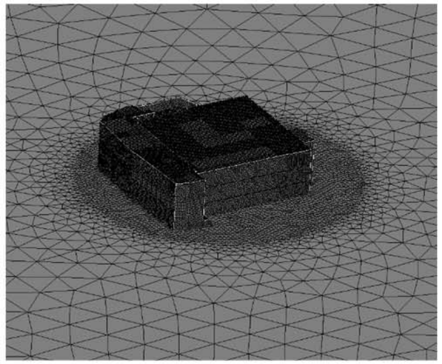
**Seism3D+ (composed of ppOpen-APPL/FDM)**

Explicit FDM Application for Seismic Wave Analysis

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

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

- $v$: velocity
- $\sigma$: stress
- $f$: external force
- $\lambda$, $\mu$: Lame's constant



**FrontISTR++ (composed of ppOpen-APPL/FEM)**

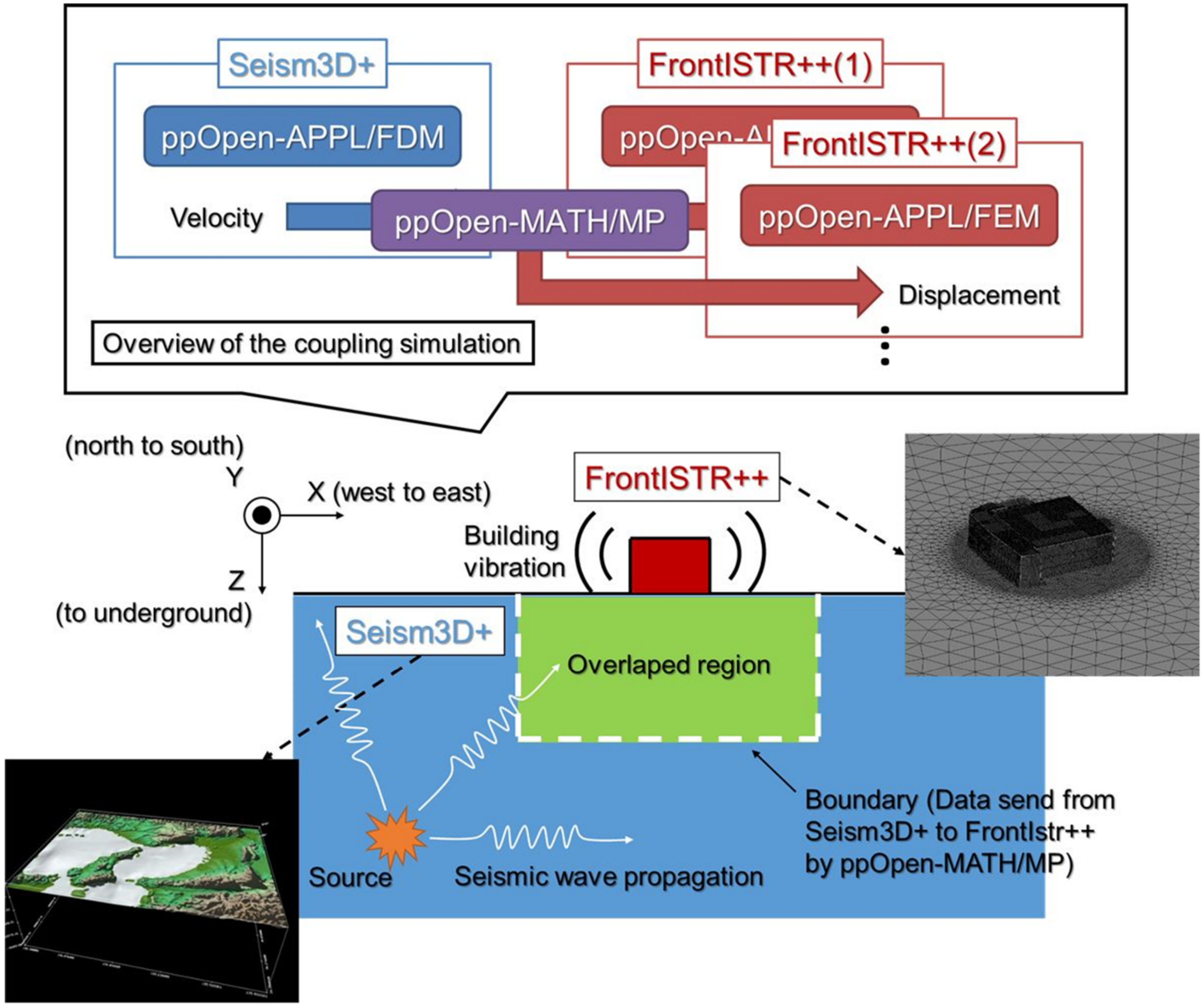
Implicit FEM Application for Structural Analysis

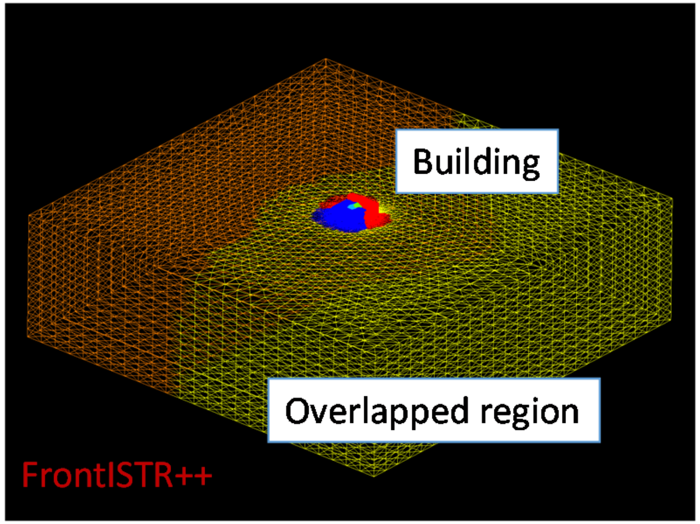
$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}$$

- $\mathbf{M}$: mass matrix
- $\mathbf{C}$: damping matrix
- $\mathbf{K}$: stiffness matrix
- $\mathbf{F}$: nodal load vector
- $\mathbf{d}$: nodal displacement vector

Computational load: Seism3D+ $<$ FrontISTR++

# Implementation of the Coupling Simulation



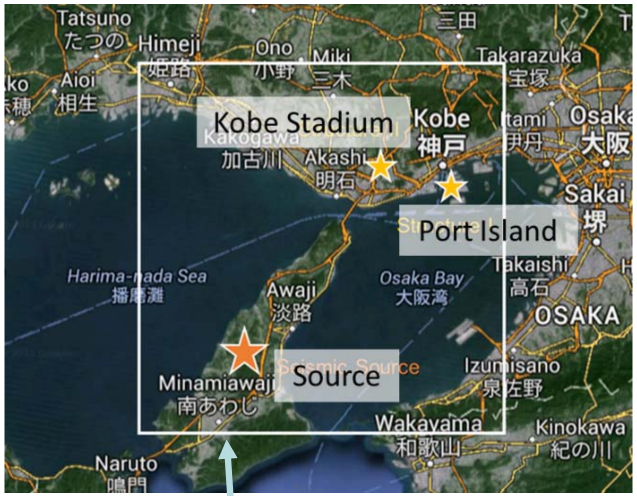




Unstructured mesh used in FrontISTR++
Colors: MPI process (64 processes)
(Partitioning by METIS)

# Practical Simulation on Oakleaf-FX

The simulation target is the earthquake that occurred at Awaji Island on 13 April, 2013. The computational domain of Seism3D+ is 60 km$^2$ from Awaji Island and that of FrontISTR++ is the actual building of RIKEN Advanced Institute for Computational Science (AICS), Port Island, Kobe, modeled by an unstructured mesh.

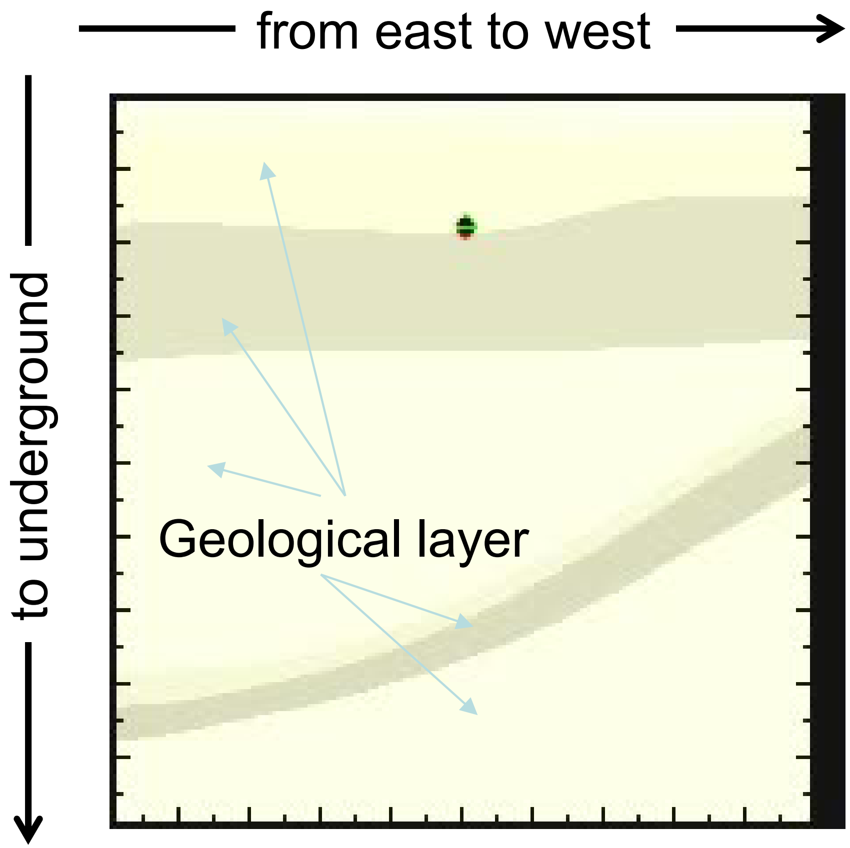
| | | |
|---|---|---|
| **Seism3D+** | Grid Points | （x, y, z）＝（1536, 1536, 1600） |
| | Parallelization | 2560 processes/16 threads |
| **FrontISTR++** | Grid Points | 600 million（AICS building） |
| | Parallelization | 1000 processes/16 threads（@Port Island）<br>1000 processes/16 threads（@Kobe Stadium） |



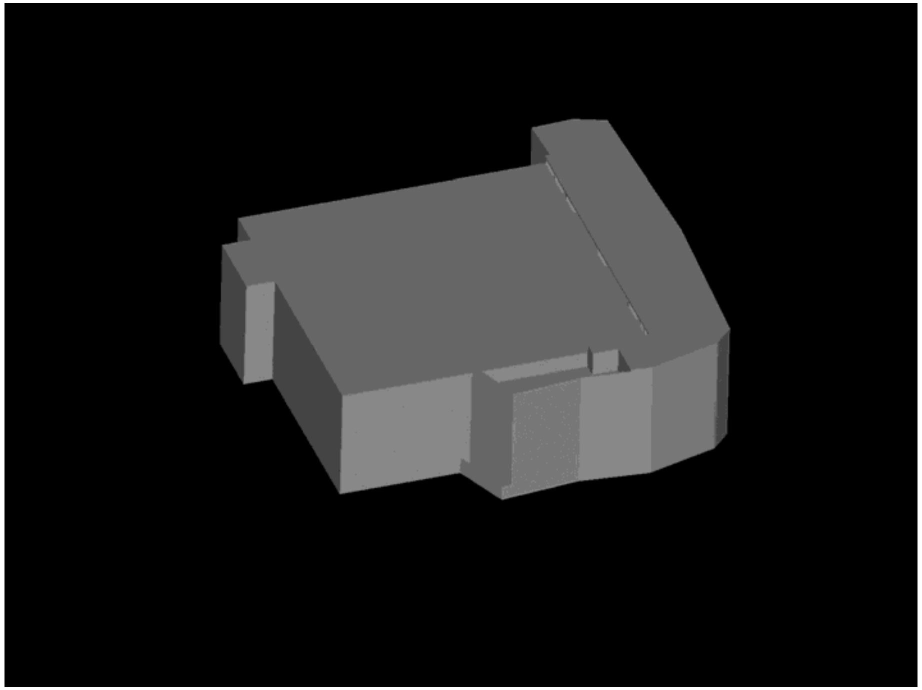

Computational domain of Seism3D+

Total 4560 nodes on Oakleaf-FX
（Seism3D+: 2560 nodes、FrontISTR++: 2000 nodes）

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




Seismic wave propagation by Seism3D+
（Red：P-wave, Green：S-wave）



Building vibration by FrontISTR++

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

- ✓ Coupling simulation was executed on large-scale computational resources of Oakleaf-FX supercomputer system.
- ✓ Seismic wave propagations (Seism3D+) for the simulation time of 90 sec., and building vibrations (FrontISTR++) for the simulation time of 20 sec. were calculated.

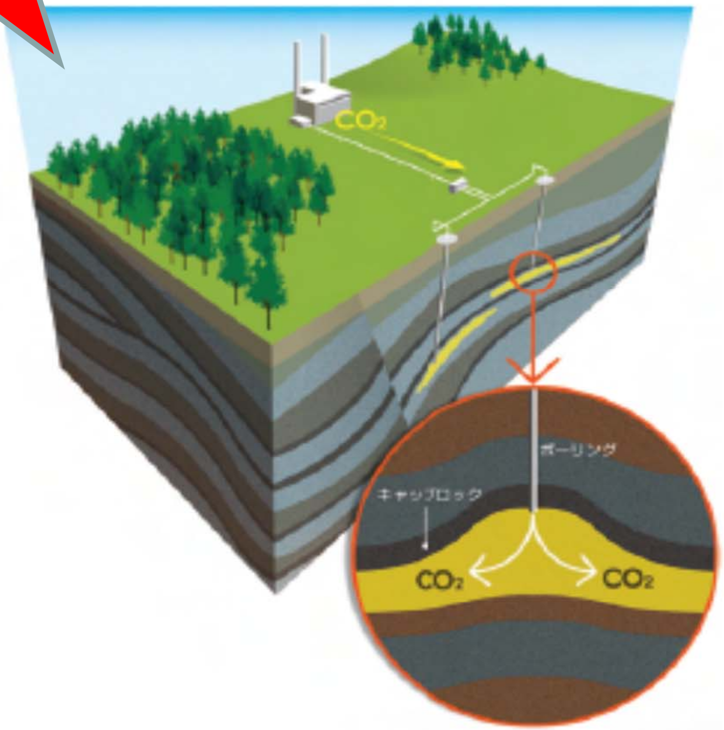
Comparison between sim. time and exe. time

| | Sim. Time | Exe. Time |
|---|---|---|
| Seism3D+ | 90 sec. | 6 hours |
| FrontISTR++ | 20 sec. | 16 hours |

- ✓ It was revealed that the manner in which memory allocation occurs in the coupler has some problem when such a large-scale simulation is performed.

2

# Simulation of Geologic $CO_2$ Storage



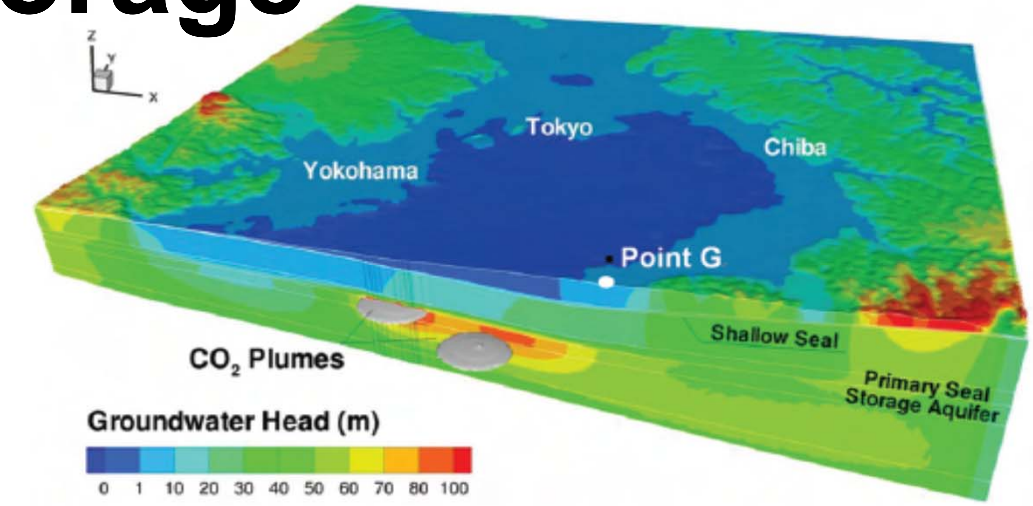
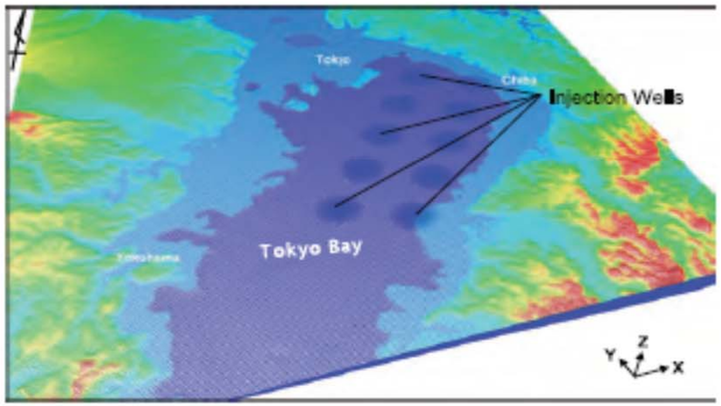
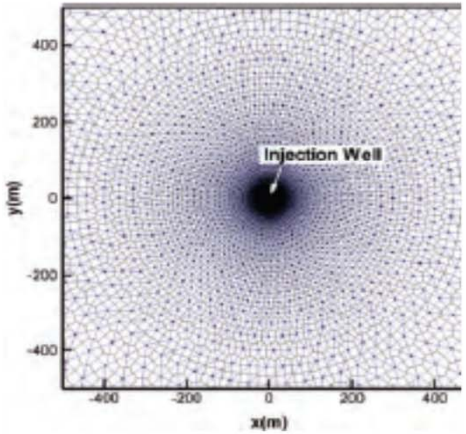
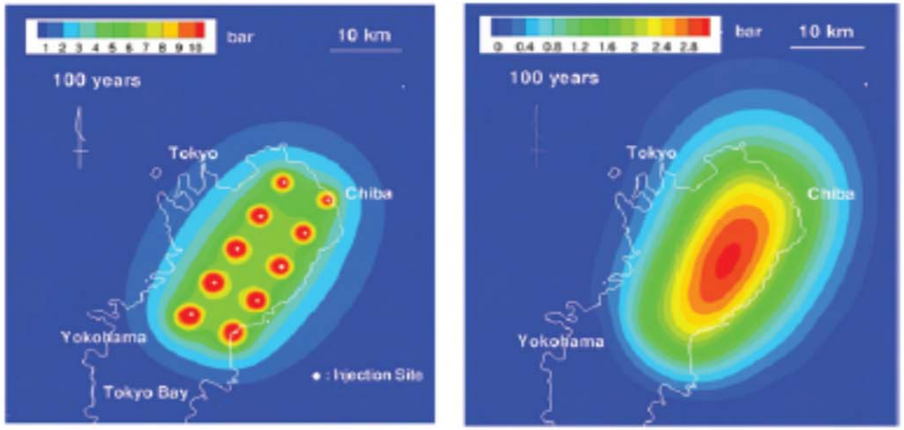




図 -4　$CO_2$ 圧入後の地下水圧（全水頭換算）の分布（100 年後）




(a) 深部遮蔽層下面　　(b) 浅部遮蔽層下面

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

[Dr. Hajime Yamamoto, Taisei]

# Simulation of Geologic $CO_2$ Storage

- International/Interdisciplinary Collaborations
  - Taisei (Science, Modeling)
  - Lawrence Berkeley National Laboratory, USA (Modeling)
  - Information Technology Center, the University of Tokyo (Algorithm, Software)
  - JAMSTEC (Earth Simulator Center) (Software, Hardware)
  - NEC (Software, Hardware)
- 2010 Japan Geotechnical Society (JGS) Award

**Science**

**Modeling**

**Algorithm**

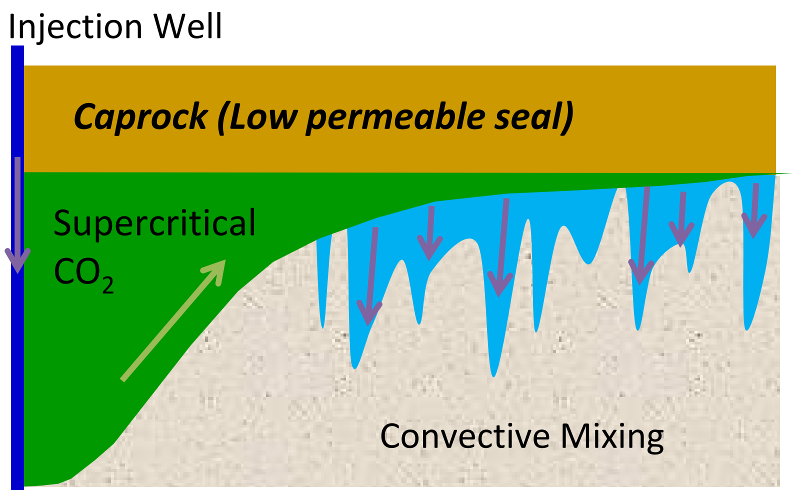
**Software**

**Hardware**

# Simulation of Geologic $CO_2$ Storage

- Science
  - Behavior of $CO_2$ in supercritical state at deep reservoir
- PDE's
  - 3D Multiphase Flow (Liquid/Gas) + 3D Mass Transfer
- Method for Computation
  - TOUGH2 code based on FVM, and developed by Lawrence Berkeley National Laboratory, USA
    - More than 90% of computation time is spent for solving large-scale linear equations with more than $10^7$ unknowns
- Numerical Algorithm
  - Fast algorithm for large-scale linear equations developed by Information Technology Center, the University of Tokyo
- Supercomputer
  - Earth Simulator II (NEX SX9, JAMSTEC, 130 TFLOPS)
  - Oakleaf-FX (Fujitsu PRIMEHP FX10, U.Tokyo, 1.13 PFLOPS

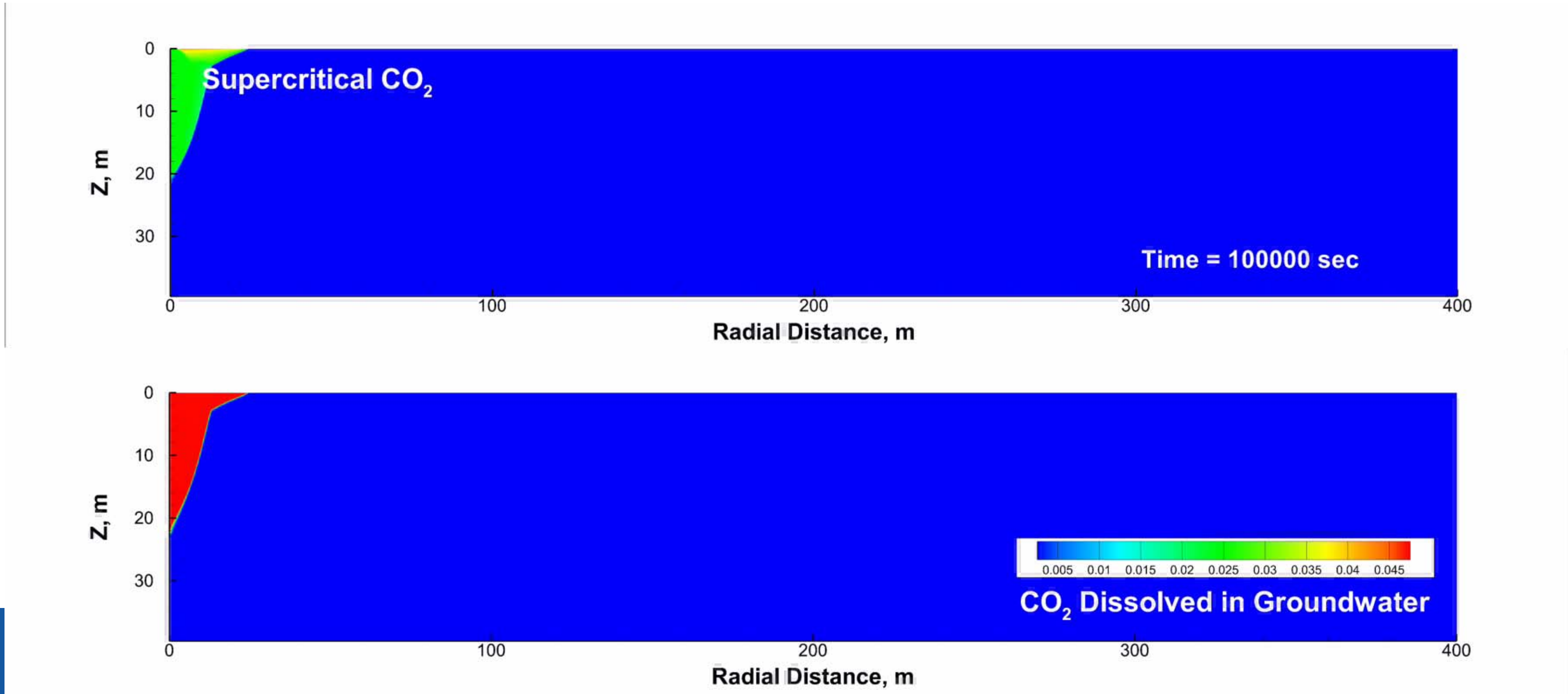
# Diffusion-Dissolution-Convection Process




- Buoyant $scCO_2$ overrides onto groundwater
- Dissolution of $CO_2$ increases water density
- Denser fluid laid on lighter fluid
- Rayleigh-Taylor instability invokes convective mixing of groundwater

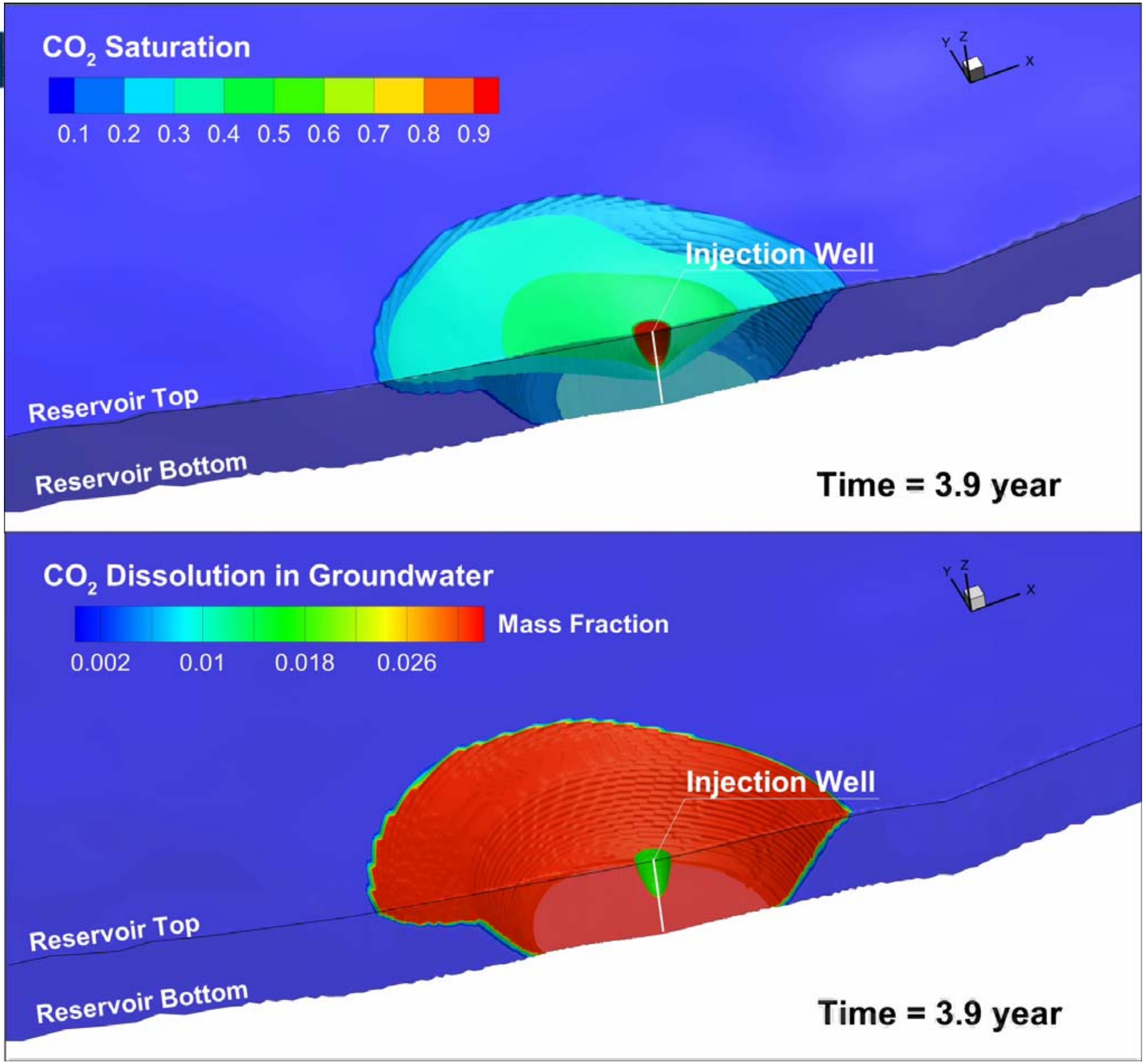
The mixing significantly enhances the $CO_2$ dissolution into groundwater, resulting in more stable storage

Preliminary 2D simulation (Yamamoto et al., GHGT11) [Dr. Hajime Yamamoto, Taisei]



# Density convections for 1,000 years:

# Flow Model




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

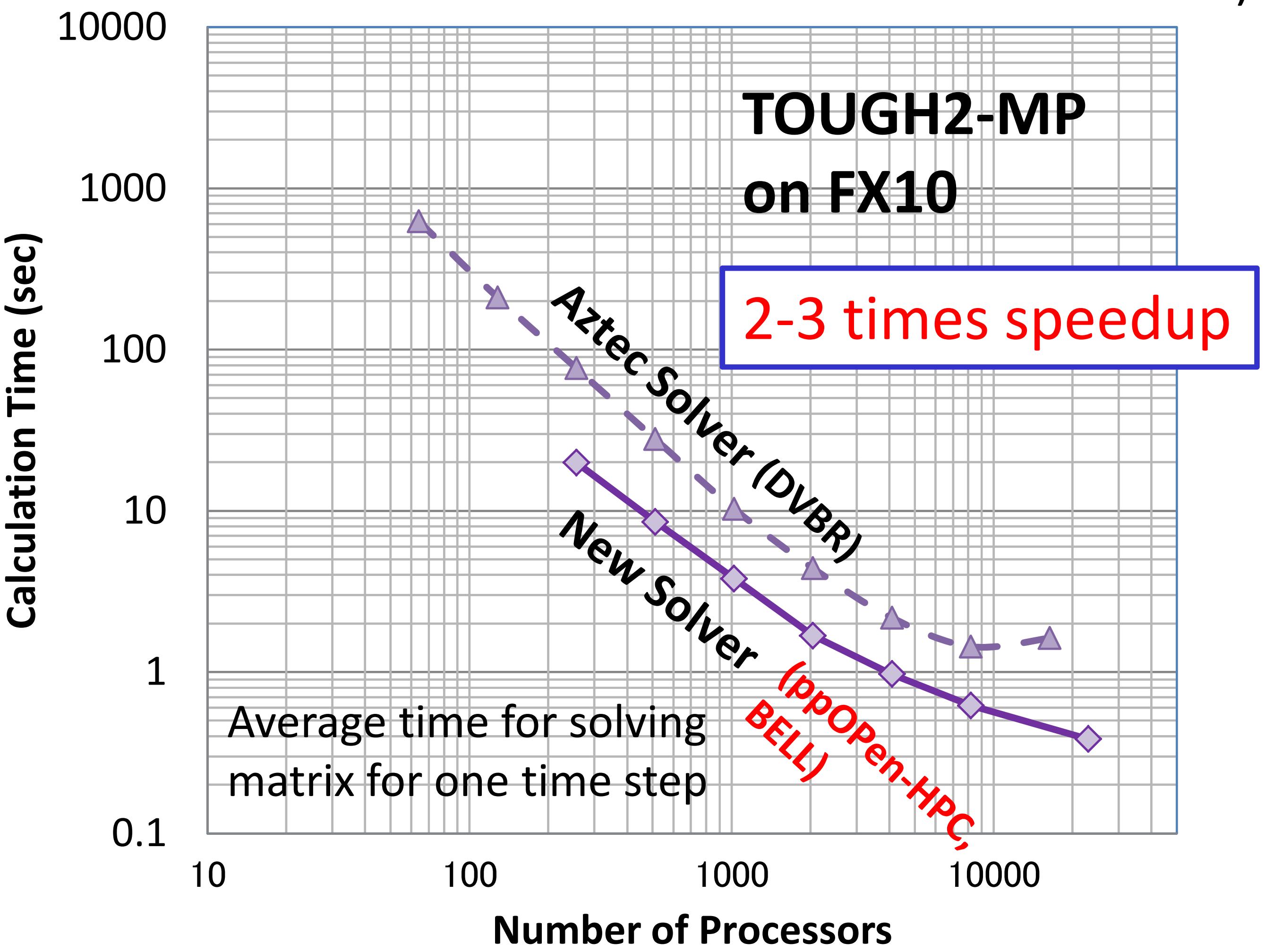
[Dr. Hajime Yamamoto, Taisei]

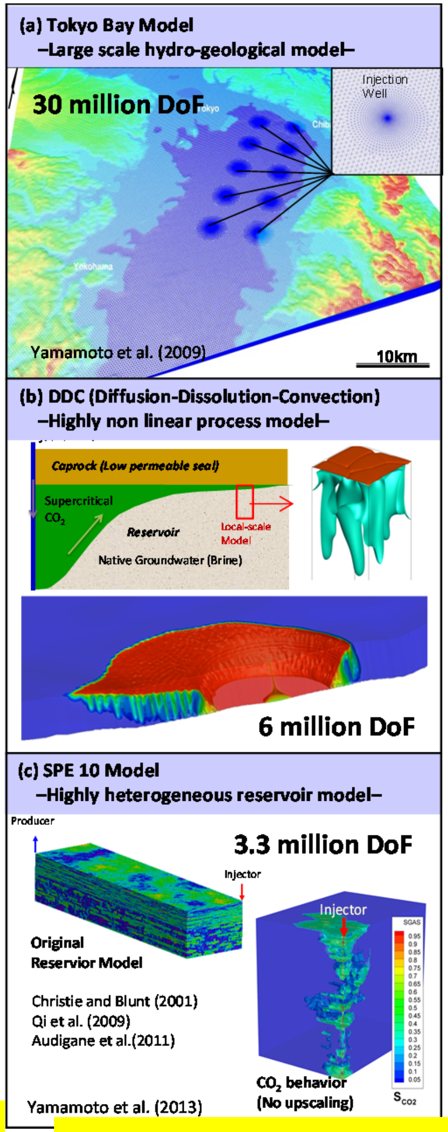
- The meter-scale fingers gradually developed to larger ones in the field-scale model
- Huge number of time steps (> $10^5$) were required to complete the 1,000-yrs simulation
- Onset time (10-20 yrs) is comparable to theoretical (linear stability analysis, 15.5yrs)

TAISEI

# Simulation of Geologic $CO_2$ Storage

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




[Dr. Hajime Yamamoto, Taisei]

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

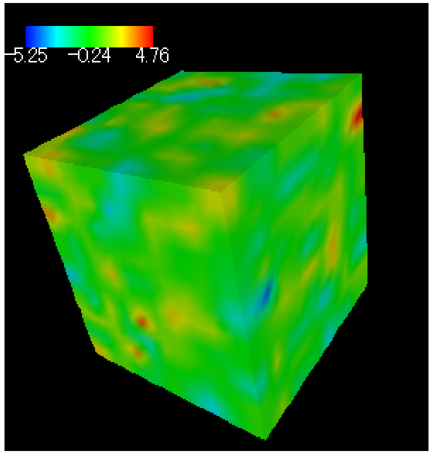
3D Multiphase Flow (Liquid/Gas) + 3D Mass Transfer

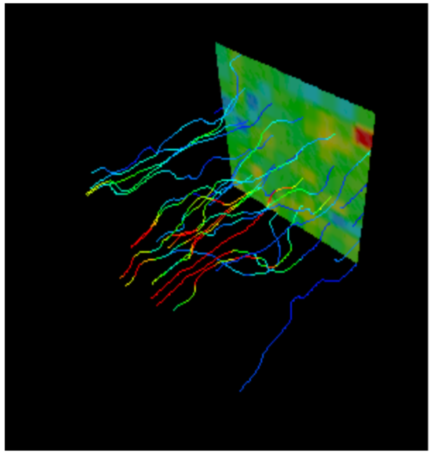
3

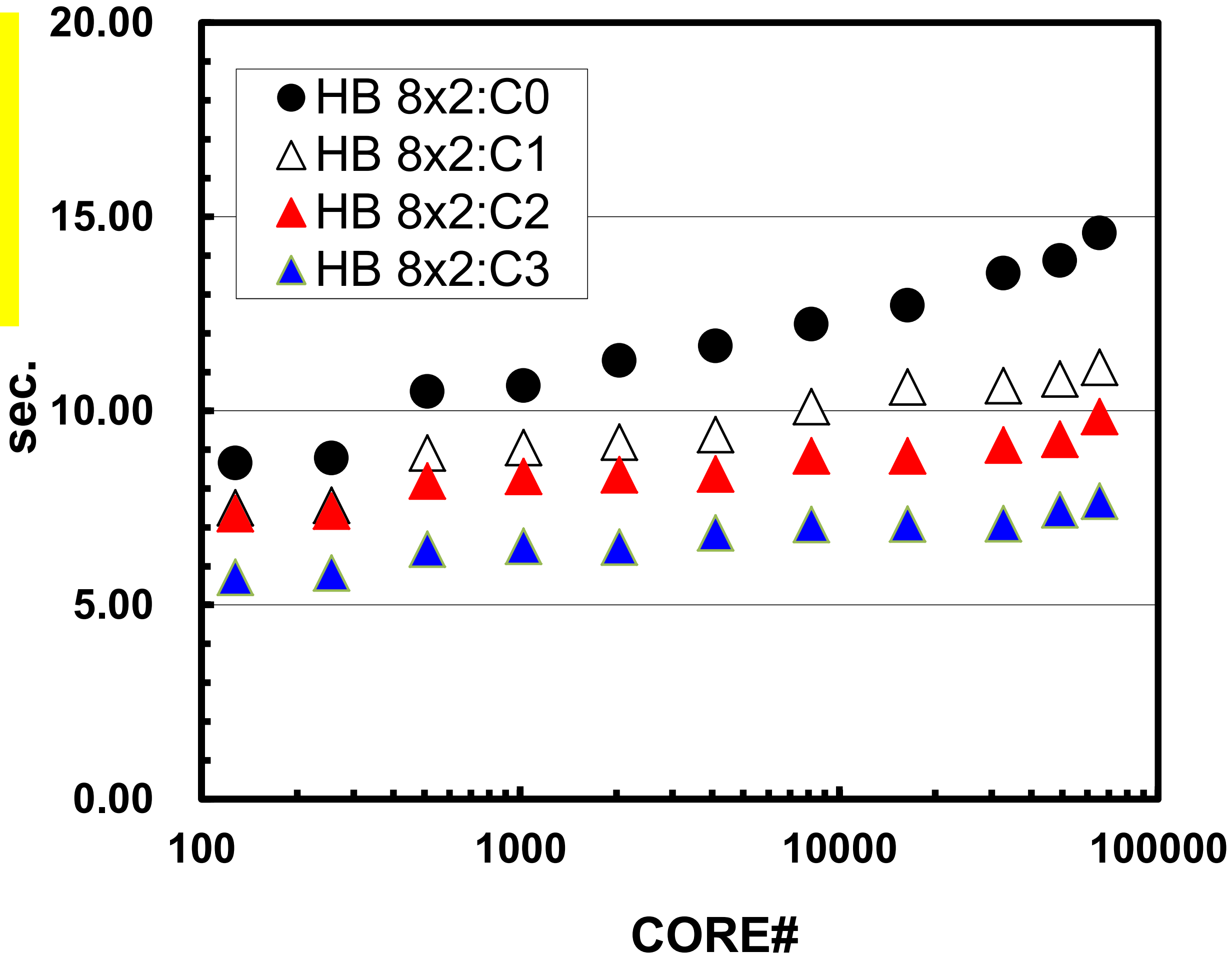
# 3D Ground Water Flow Simulation with up to 4,096 nodes on Fujitsu FX10 (GMG-CG)

up to 17,179,869,184 meshes ($64^3$ meshes/core)

**Linear equations with 17B unknowns can be solved in 7 sec.**





# Motivation for Parallel Computing again

- **Large-scale parallel computer enables fast computing in large-scale scientific simulations with detailed models. Computational science develops new frontiers of science and engineering.**
- Why parallel computing ?
  - faster & larger
  - "larger" is more important from the view point of "new frontiers of science & engineering", but "faster" is also important.
  - + more complicated
  - Ideal: Scalable
    - Solving $N^x$ scale problem using $N^x$ computational resources during same computation time (weak scaling)
    - Solving a fix-sized problem using $N^x$ computational resources in 1/N computation time (strong scaling)

- Supercomputers and Computational Science
- **Overview of the Class**

# Goal of SC4SC 2017

- **If you want to do something on supercomputers, you have to learn "parallel programming" !!**
- Introduction to MPI & OpenMP
  - MPI: Message Passing Interface
    - "grab" the idea of **SPMD (Single-Program Multiple-Data)**
  - OpenMP: Multithreading
- Parallel Application
  - Finite-Volume Method (FVM): We start at this part
  - Data Structure for Parallel Computing
  - Parallel FVM by OpenMP
  - Parallel FVM by OpenMP/MPI Hybrid

# Initial Structure of this Short Course

| | |
|---|---|
| Feb. 21 (T) | **Part-I (Katagiri):** Introduction to OpenMP & MPI |
| Feb. 22 (W) | |
| Feb. 23 (Th) | **Part-II (Nakajima):** Parallel FVM |
| Feb. 24 (F) | |

# New Structure of this Short Course

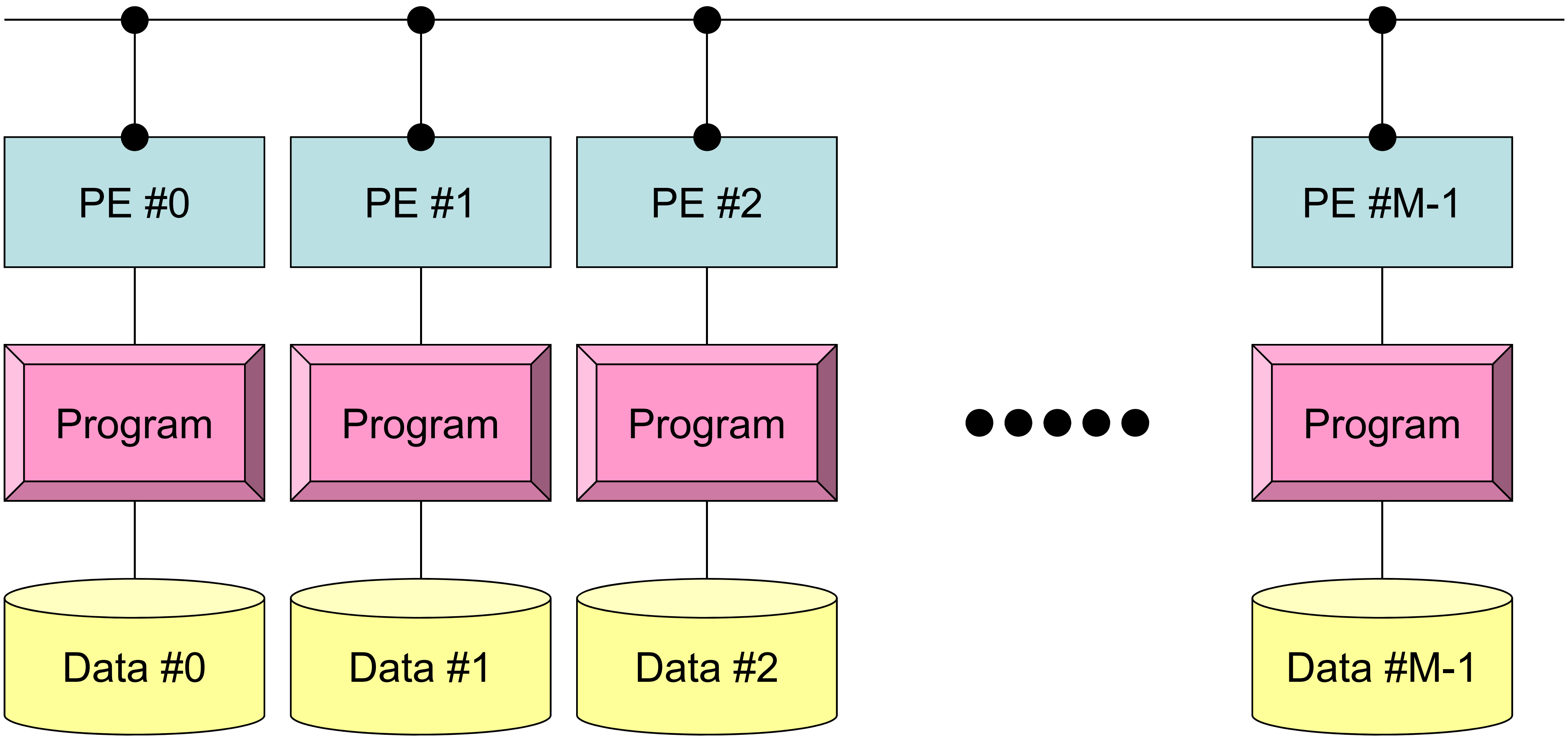
| | |
|---|---|
| Feb. 21 (T) | Part-II-1 (Nakajima): Introduction to FVM |
| Feb. 21 (T), Feb. 22 (W), Feb. 23 (Th) | Part-I (Katagiri): Introduction to OpenMP & MPI |
| Feb. 23 (Th), Feb. 24 (F) | Part-II-2 (Nakajima): Parallel FVM<br>FX10 Supercomputer is not available on Feb.24 |

PE: Processing Element
Processor, Domain, Process

# SPMD

You understand 90% MPI, if you understand this figure.

`mpirun -np M <Program>`



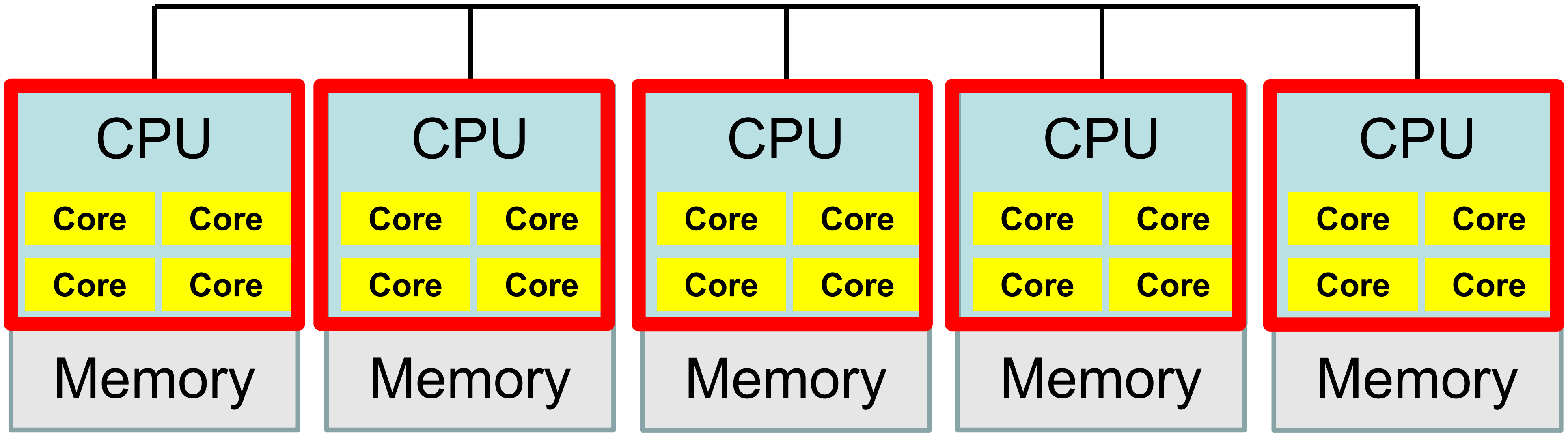

Each process does same operation for different data

Large-scale data is decomposed, and each part is computed by each process

It is ideal that parallel program is not different from serial one except communication.

# Our Current Target: Multicore Cluster

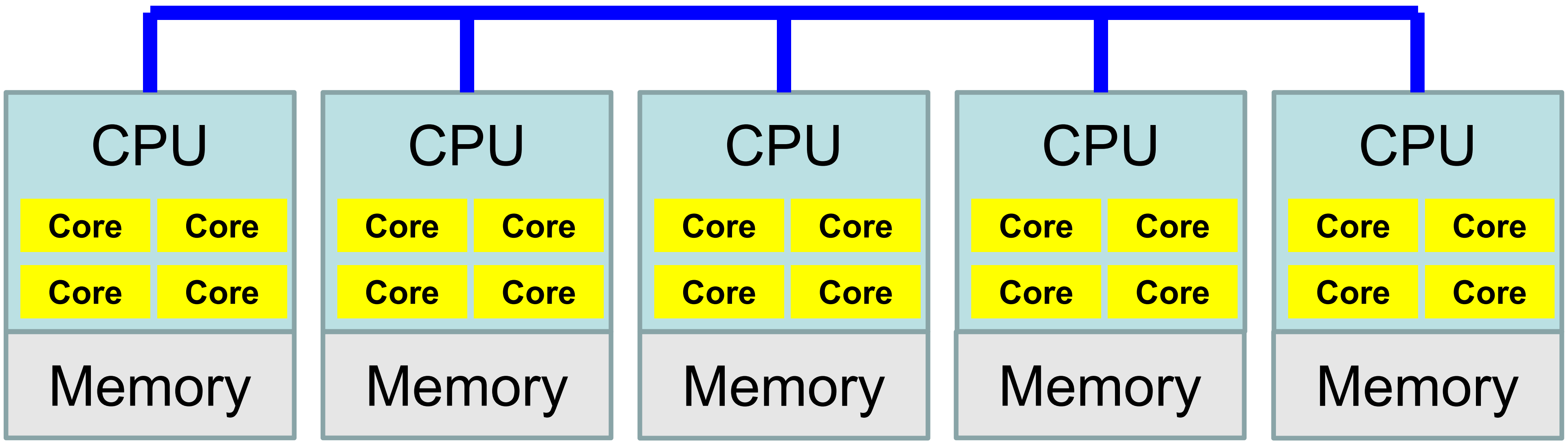
Multicore CPU's are connected through network




- OpenMP
  - ✓ Multithreading
  - ✓ Intra Node (Intra CPU)
  - ✓ Shared Memory
- MPI
  - ✓ Message Passing
  - ✓ Inter Node (Inter CPU)
  - ✓ Distributed Memory

# Our Current Target: Multicore Cluster

Multicore CPU's are connected through network

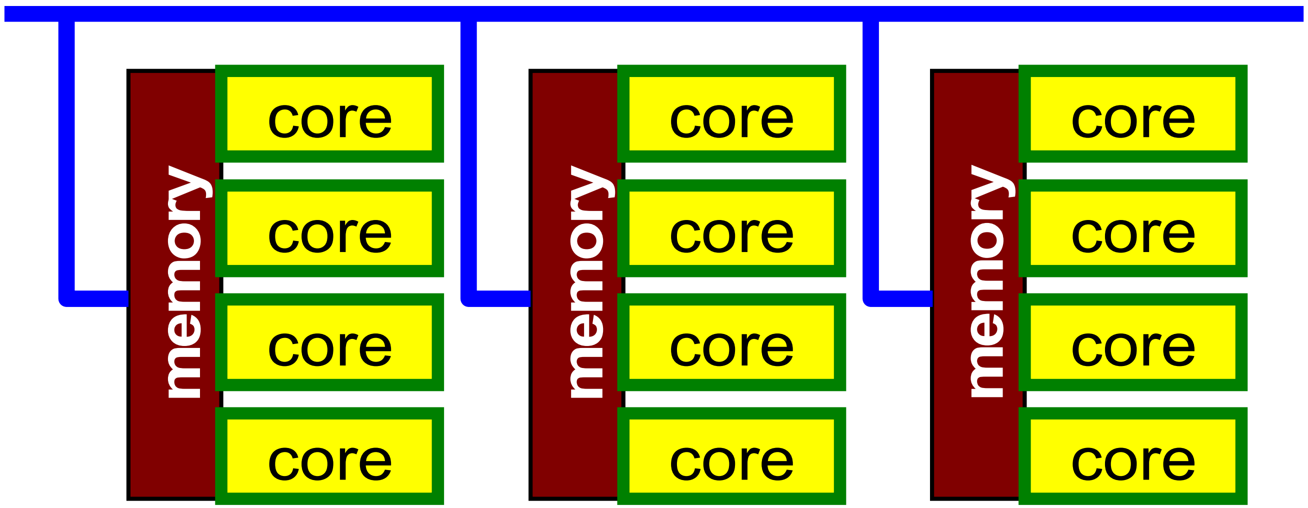



- OpenMP
  - ✓ Multithreading
  - ✓ Intra Node (Intra CPU)
  - ✓ Shared Memory
- MPI
  - ✓ Message Passing
  - ✓ Inter Node (Inter CPU)
  - ✓ Distributed Memory

# Flat MPI vs. Hybrid

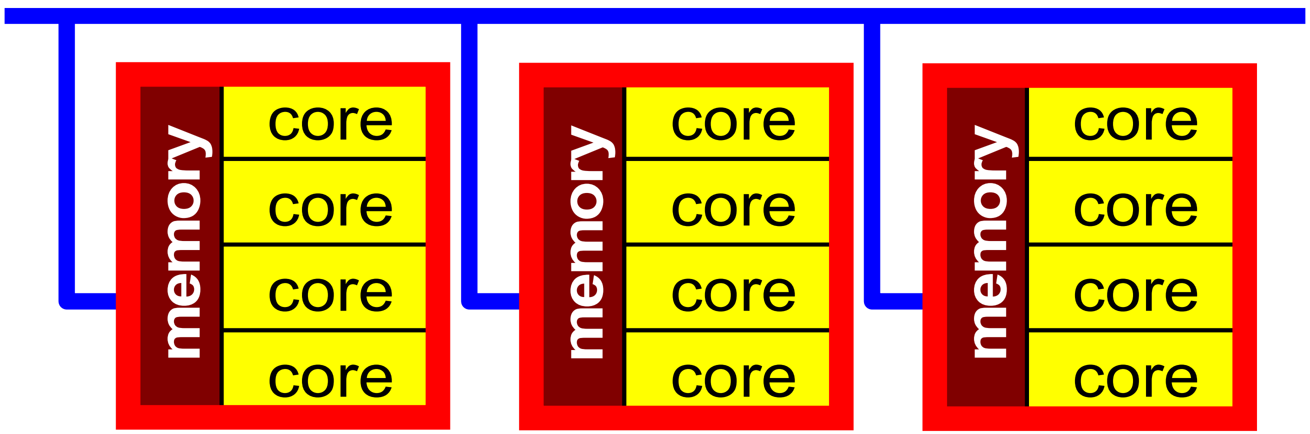
## Flat-MPI：Each Core -> Independent

- MPI only
- Intra/Inter Node




## Hybrid：Hierarchal Structure

- OpenMP
- MPI



# Example of OpnMP/MPI Hybrid

## Sending Messages to Neighboring Processes

MPI: Message Passing, OpenMP: Threading with Directives

```
!C
!C- SEND

      do neib= 1, NEIBPETOT
        II= (LEVEL-1)*NEIBPETOT
        istart= STACK_EXPORT(II+neib-1)
        inum  = STACK_EXPORT(II+neib  ) - istart
!$omp parallel do
        do k= istart+1, istart+inum
          WS(k-NE0)= X(NOD_EXPORT(k))
        enddo

        call MPI_Isend (WS(istart+1-NE0), inum, MPI_DOUBLE_PRECISION,   &
     &                  NEIBPE(neib), 0, MPI_COMM_WORLD,                &
     &                  req1(neib), ierr)
      enddo
```

# Prerequisites

- Experiences in Unix/Linux
- Experiences of programming (Fortran or C/C++)
- Fundamental numerical algorithms (Gaussian Elimination, LU Factorization, Jacobi/Gauss-Seidel/SOR Iterative Solvers)
- Experiences in SSH Public Key Authentication Method

# Preparation

- Windows
  - Cygwin: Please install gcc (C) or gfortran (Fortran) and SSH !!
  - ParaView
- MacOS, UNIX/Linux
  - ParaView

- Cygwin: https://www.cygwin.com/
- ParaView: http://www.paraview.org/