Parallel Finite Element Method
Introduction
Overview of the Class

Kengo Nakajima
Information Technology Center
The University of Tokyo

- Target: Parallel FEM
- Supercomputers and Computational Science
- Overview of the Class
- Future Issues

This 5-day intensive course provides introduction to large-scale scientific computing using the most advanced massively parallel supercomputers. Topics cover:

- Finite-Element Method (FEM)
- Message Passing Interface (MPI)
- Parallel FEM using MPI and OpenMP
- Parallel Numerical Algorithms for Iterative Linear Solvers

Several sample programs will be provided and participants can review the contents of lectures through hands-on-exercise/practices using **Oakridge-CX System with Intel® Xeon® Platinum 8280 (Cascade Lake, CLX)** at the University of Tokyo.

Finite-Element Method is widely-used for solving various types of real-world scientific and engineering problems, such as structural analysis, fluid dynamics, electromagnetics, and etc. This lecture course provides brief introduction to procedures of FEM for 1D/3D steady-state heat

conduction problems with iterative linear solvers and to parallel FEM. **Lectures for parallel FEM will be focused on design of data structure for distributed local mesh files, which is the key issue for efficient parallel FEM.** Introduction to MPI (Message Passing Interface), which is widely used method as "de facto standard" of parallel programming, is also provided.

Solving large-scale linear equations with sparse coefficient matrices is the most expensive and important part of FEM and other methods for scientific computing, such as Finite-Difference Method (FDM) and Finite-Volume Method (FVM). Recently, families of Krylov iterative solvers are widely used for this process. In this course, details of implementations of parallel Krylov iterative methods are provided along with parallel FEM.

Moreover, lectures on programming for multicore architectures will be also given along with brief introduction to OpenMP and OpenMP/MPI Hybrid Parallel Programming Model.

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
    - Weak Scaling, Strong Scaling

Scalable, Scaling, Scalability

- Solving N^x scale problem using N^x computational resources during same computation time
  - for large-scale problems: **Weak Scaling, Weak Scalability**
  - e.g. CG solver: more iterations needed for larger problems
- Solving a problem using N^x computational resources during 1/N computation time
  - for faster computation: **Strong Scaling, Strong Scalability**

Kengo Nakajima (1/2) 中島研吾

- Current Position
  - Professor, Supercomputing Research Division, Information Technology Center 情報基盤センター
  - Professor, Department of Mathematical Informatics, Graduate School of Information Science & Engineering 数理情報学専攻
  - Professor, Department of Electrical Engineering & Information Systems, Graduate School of Engineering 電気系工学専攻
  - Deputy Director, RIKEN Center for Computational Science (R-CCS) (2018 April -)
- Research Interest
  - High-Performance Computing
  - Parallel Numerical Linear Algebra (Preconditioning)
  - Parallel Programming Model
  - Computational Mechanics, Computational Fluid Dynamics
  - Adaptive Mesh Refinement, Parallel Visualization

Kengo Nakajima (2/2)

- Education
  - B.Eng (Aeronautics, The University of Tokyo, 1985)
  - M.S. (Aerospace Engineering, University of Texas, 1993)
  - Ph.D. (Quantum Engineering & System Sciences, The University of Tokyo, 2003)
- Professional Background
  - Mitsubishi Research Institute, Inc. (1985-1999)
  - Research Organization for Information Science & Technology (1999-2004)
  - The University of Tokyo
    - Department Earth & Planetary Science (2004-2008)
    - Information Technology Center (2008-)
  - JAMSTEC (2008-2011), part-time
  - RIKEN (2009-), part-time

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): Your Department ?

This Class ...

Science

Modeling

Algorithm

Software

Hardware

- Parallel FEM using MPI and OpenMP

- Science: Heat Conduction
- Modeling: FEM
- Algorithm: Iterative Solvers etc.

- You have to know many components to learn FEM, although you have already learned each of these in undergraduate and high-school classes.

Road to Programming for “Parallel” Scientific Computing

Programming for Parallel Scientific Computing (e.g. Parallel FEM/FDM)

Programming for Real World Scientific Computing (e.g. FEM, FDM)

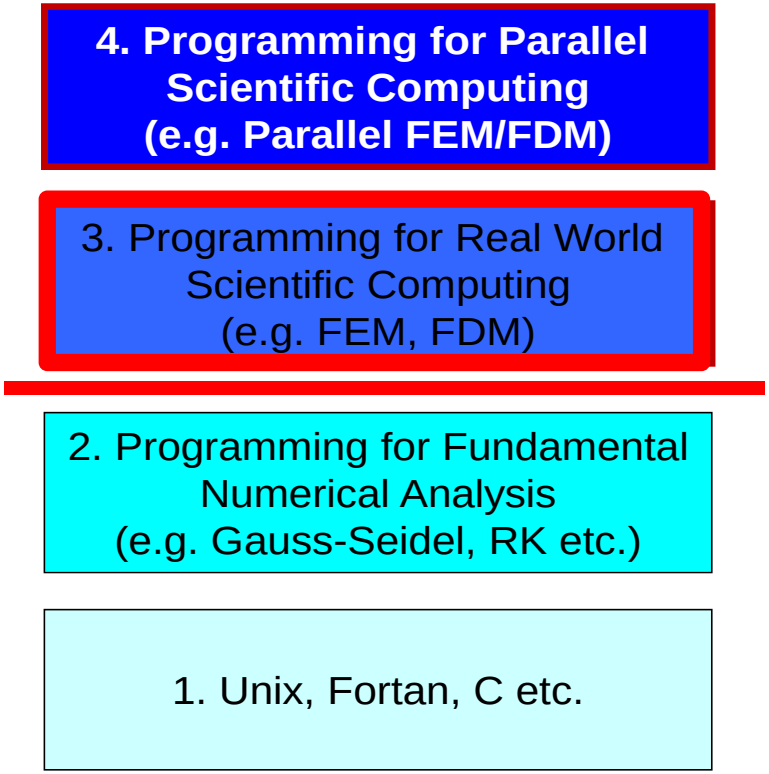
Big gap here !!

Programming for Fundamental Numerical Analysis (e.g. Gauss-Seidel, RK etc.)

Unix, Fortan, C etc.

The third step is important !

- How to parallelize applications ?
  - How to extract parallelism ?
  - If you understand methods, algorithms, and implementations of the original code, it's easy.
  - "Data-structure" is important



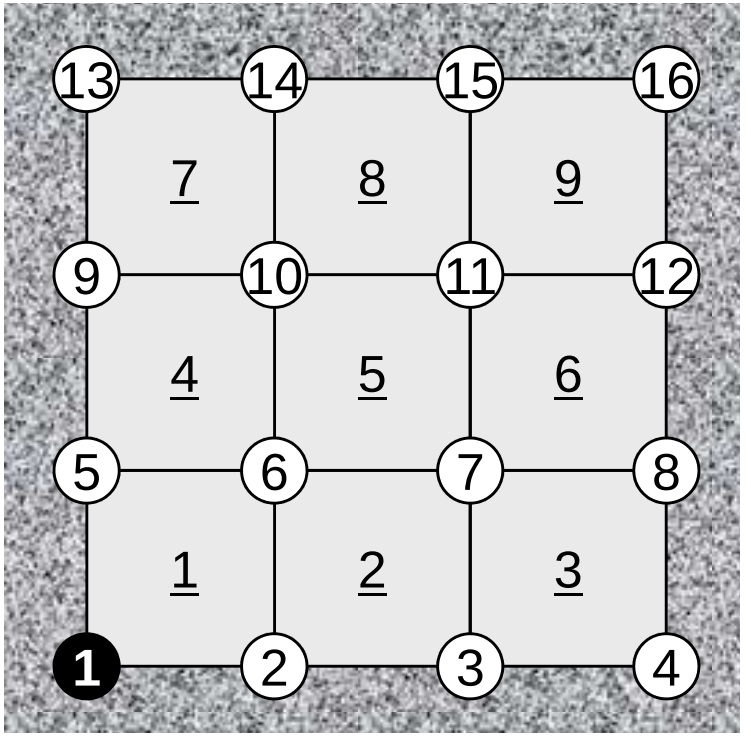
- How to understand the code ?
  - Reading the application code !!
  - It seems primitive, but very effective.
  - In this class, "reading the source code" is encouraged.

Finite-Element Method (FEM)

- One of the most popular numerical methods for solving PDE's.
  - elements (meshes) & nodes (vertices)
- Consider the following 2D heat transfer problem:

$$\lambda\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+Q=0$$

  - 16 nodes, 9 bi-linear elements
  - uniform thermal conductivity (λ=1)
  - uniform volume heat flux (Q=1)
  - T=0 at node 1
  - Insulated boundaries



Galerkin FEM procedures

- Apply Galerkin procedures to each element:

$$\int_V [N]^T \left\{ \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q \right\} dV = 0$$

where $T = [N]\{\phi\}$ in each elem.

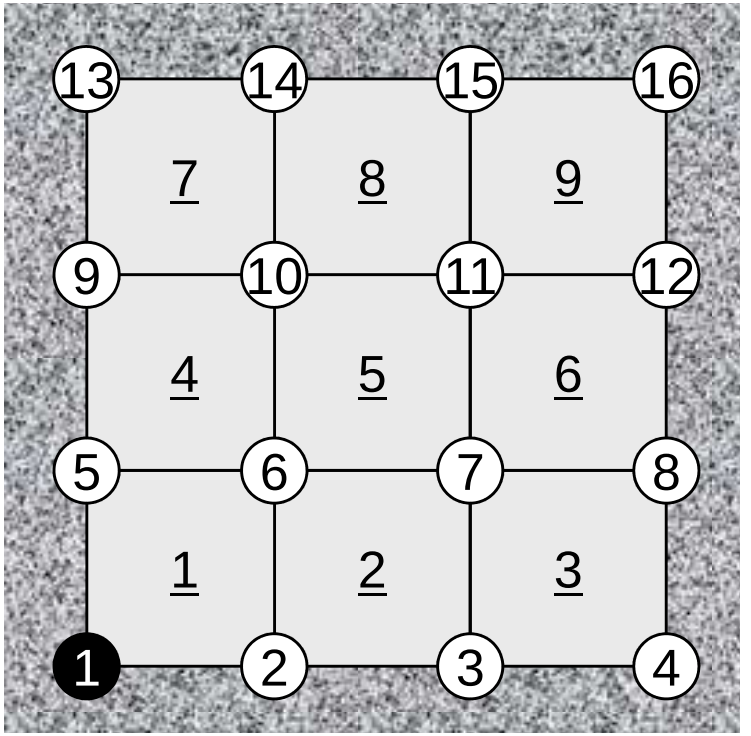
$\{\phi\}$: T at each vertex

$[N]$: Shape function (Interpolation function)

- Introduce the following "weak form" of original PDE using Green's theorem:

$$-\int_V \lambda \left(\frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial [N]^T}{\partial y} \frac{\partial [N]}{\partial y} \right) dV \cdot \{\phi\}$$

$$+\int_V Q[N]^T dV = 0$$

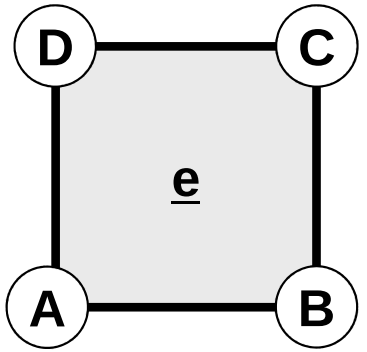


Element Matrix

- Apply the integration to each element and form “element” matrix.

$$-\int_V \lambda\left(\frac{\partial[N]^T}{\partial x}\frac{\partial[N]}{\partial x}+\frac{\partial[N]^T}{\partial y}\frac{\partial[N]}{\partial y}\right)dV\cdot\{\phi\}$$

$$+\int_V Q[N]^T dV = 0$$

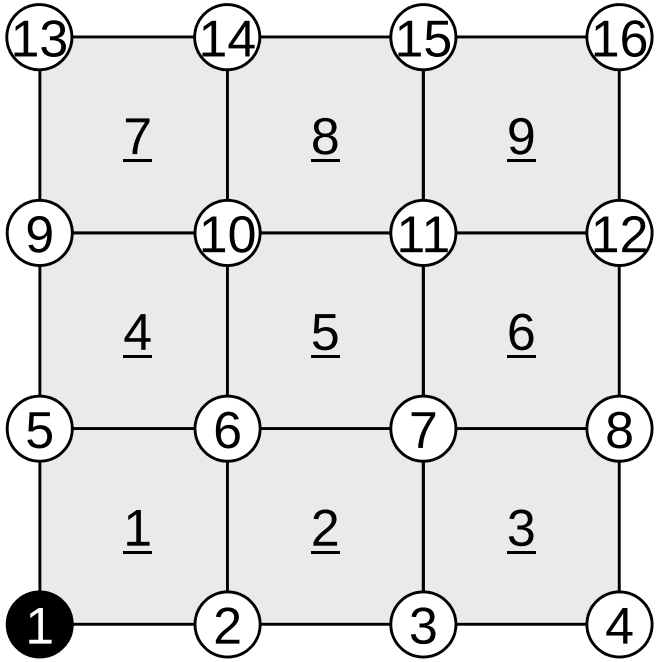


$$[k^{(e)}]\{\phi^{(e)}\}=\{f^{(e)}\}$$

$$\begin{bmatrix} k_{AA}^{(e)} & k_{AB}^{(e)} & k_{AC}^{(e)} & k_{AD}^{(e)} \\ k_{BA}^{(e)} & k_{BB}^{(e)} & k_{BC}^{(e)} & k_{BD}^{(e)} \\ k_{CA}^{(e)} & k_{CB}^{(e)} & k_{CC}^{(e)} & k_{CD}^{(e)} \\ k_{DA}^{(e)} & k_{DB}^{(e)} & k_{DC}^{(e)} & k_{DD}^{(e)} \end{bmatrix} \begin{Bmatrix} \phi_A^{(e)} \\ \phi_B^{(e)} \\ \phi_C^{(e)} \\ \phi_D^{(e)} \end{Bmatrix} = \begin{Bmatrix} f_A^{(e)} \\ f_B^{(e)} \\ f_C^{(e)} \\ f_D^{(e)} \end{Bmatrix}$$

Global (Overall) Matrix

Accumulate each element matrix to “global” matrix.

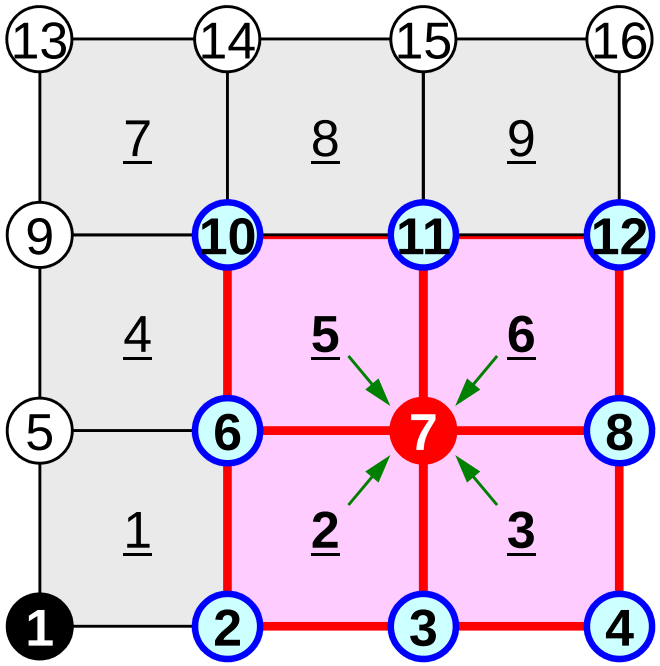


$$[K]\{\Phi\} = \{F\}$$

$$\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
 & X & D & X & & X & X & X & & & & & & & & \\
 & & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
 & X & X & X & & X & D & X & & X & X & X & & & & \\
 & & X & X & & & X & D & & & X & X & & & & \\
 & & & & X & X & & & D & X & & & X & X & & \\
 & & & & X & X & X & & X & D & X & & X & X & X & \\
 & & & & & X & X & X & & X & D & X & & X & X & X \\
 & & & & & & X & X & & & X & D & & & X & X \\
 & & & & & & & & X & X & & & D & X & & \\
 & & & & & & & & X & X & X & & X & D & X & \\
 & & & & & & & & & X & X & X & & X & D & X \\
 & & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16} \end{Bmatrix}
=
\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16} \end{Bmatrix}$$

To each node …

Effect of surrounding elem's/nodes are accumulated.



$$[K]\{\Phi\} = \{F\}$$

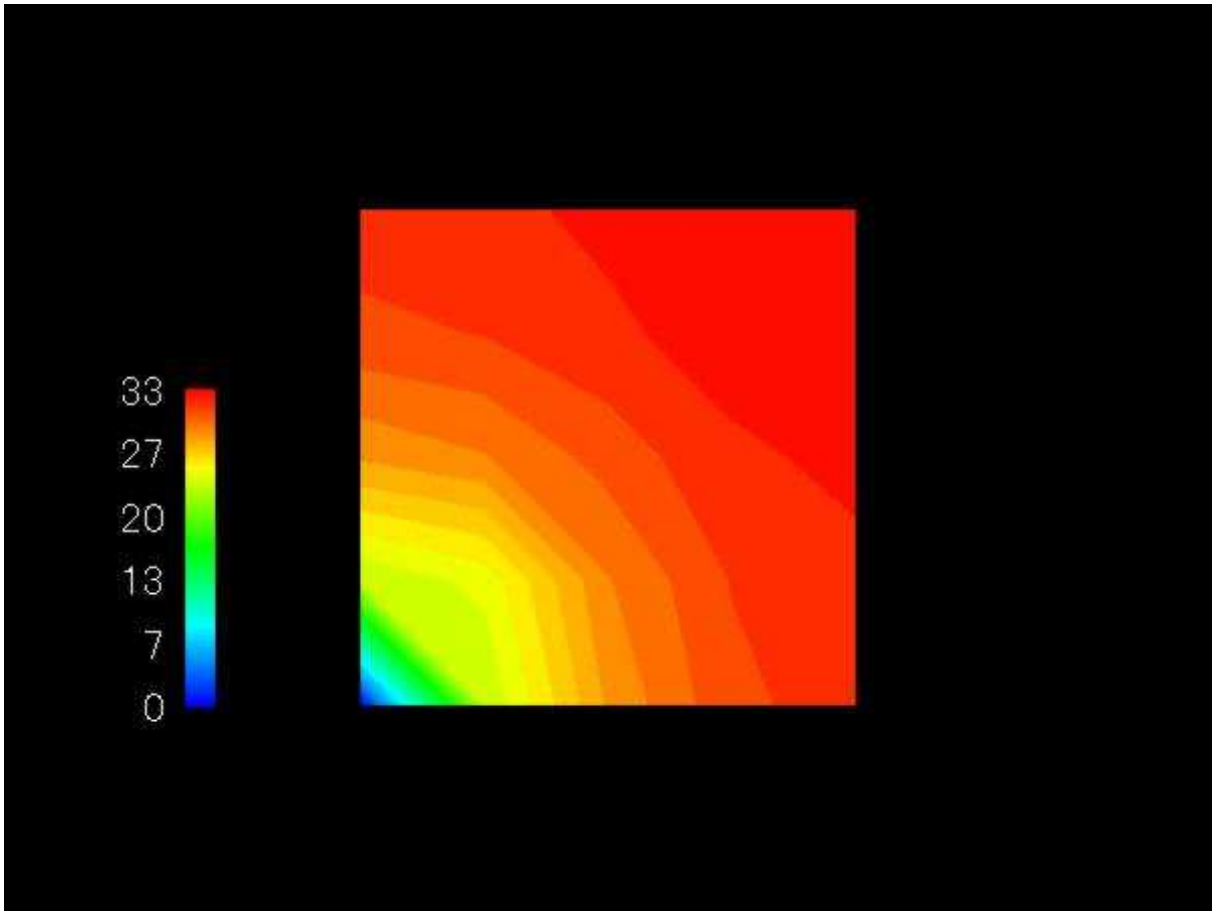
$$\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
 & X & D & X & & X & X & X & & & & & & & & \\
 & & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
 & X & X & X & & X & D & X & & X & X & X & & & & \\
 & & X & X & & & X & D & & & X & X & & & & \\
 & & & & X & X & & & D & X & & & X & X & & \\
 & & & & X & X & X & & X & D & X & & X & X & X & \\
 & & & & & X & X & X & & X & D & X & & X & X & X \\
 & & & & & & X & X & & & X & D & & & X & X \\
 & & & & & & & & X & X & & & D & X & & \\
 & & & & & & & & X & X & X & & X & D & X & \\
 & & & & & & & & & X & X & X & & X & D & X \\
 & & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16} \end{Bmatrix}
=
\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16} \end{Bmatrix}$$

Solve the obtained global eqn's

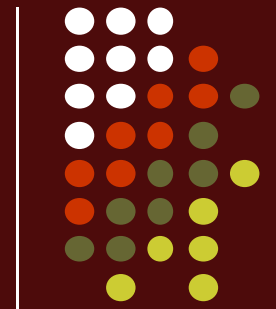
under certain boundary conditions
(Φ_1=0 in this case)

$$
\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
 & X & D & X & & X & X & X & & & & & & & & \\
 & & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
 & X & X & X & & X & D & X & & X & X & X & & & & \\
 & & X & X & & & X & D & & & X & X & & & & \\
 & & & & X & X & & & D & X & & & X & X & & \\
 & & & & X & X & X & & X & D & X & & X & X & X & \\
 & & & & & X & X & X & & X & D & X & & X & X & X \\
 & & & & & & X & X & & & X & D & & & X & X \\
 & & & & & & & & X & X & & & D & X & & \\
 & & & & & & & & X & X & X & & X & D & X & \\
 & & & & & & & & & X & X & X & & X & D & X \\
 & & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16} \end{Bmatrix}
=
\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16} \end{Bmatrix}
$$

Result …



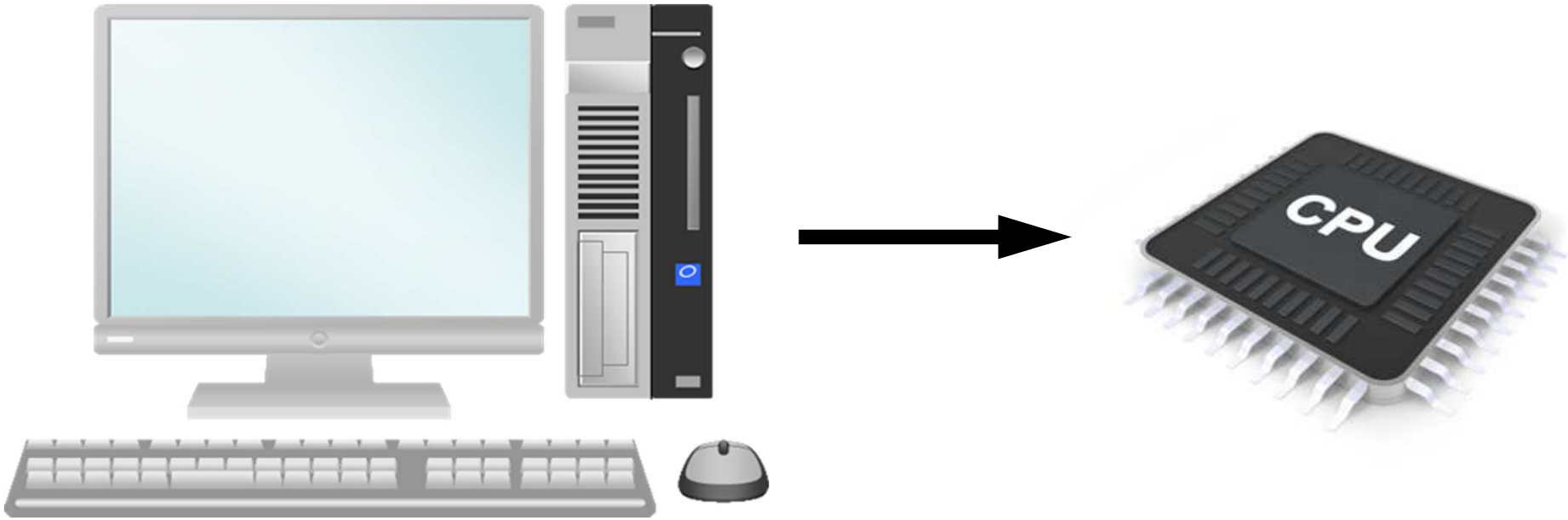
Features of FEM applications



- Typical Procedures for FEM Computations
  - Input/Output
  - Matrix Assembling
  - Linear Solvers for Large-scale Sparse Matrices
  - Most of the computation time is spent for matrix assembling/formation and solving linear equations.
- **HUGE** “indirect” accesses
  - memory intensive
- Local “element-by-element” operations
  - sparse coefficient matrices
  - suitable for parallel computing
- Excellent modularity of each procedure

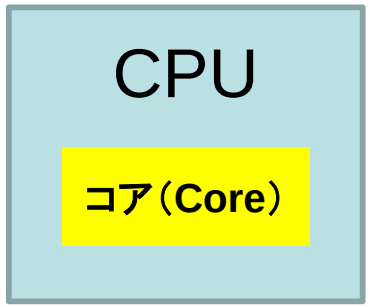
- Target: Parallel FEM
- **Supercomputers and Computational Science**
- Overview of the Class
- Future Issues

Computer & CPU

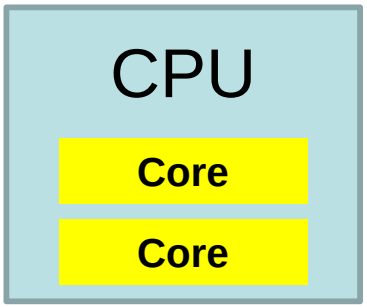


- Central Processing Unit（中央処理装置）:CPU
- CPU’s used in PC and Supercomputers are based on same architecture
- GHz: Clock Rate
  - Frequency: Number of operations by CPU per second
    - GHz -> 10^9 operations/sec
  - Simultaneous 4-8 instructions per clock

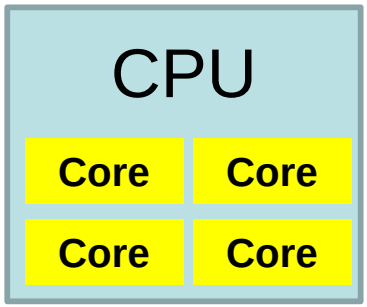
Multicore CPU



Single Core
1 cores/CPU

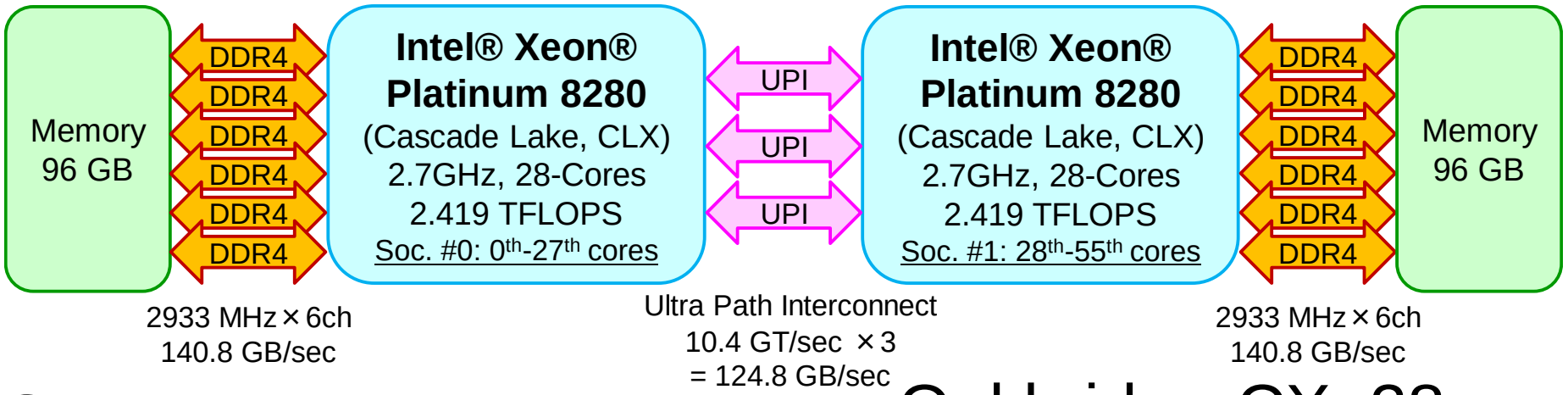


Dual Core
2 cores/CPU



Quad Core
4 cores/CPU

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



2933 MHz × 6ch
140.8 GB/sec

Ultra Path Interconnect
10.4 GT/sec × 3
= 124.8 GB/sec

2933 MHz × 6ch
140.8 GB/sec

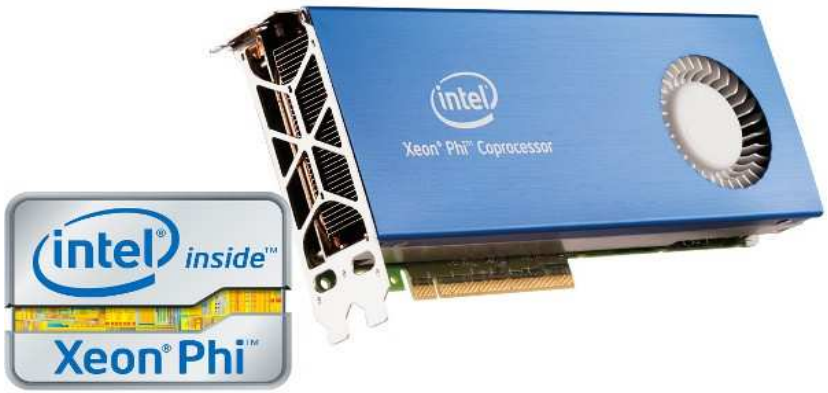
- GPU: Manycore
  - $O(10^1)$-$O(10^2)$ cores
- More and more cores
  - Parallel computing

- Oakbridge-CX: 28 cores x 2
  - Intel Xeon Platinum 8280
  - Cascade Lake, CLX
  - Intel Xeon SP
    - Scalable Processor

GPU/Manycores

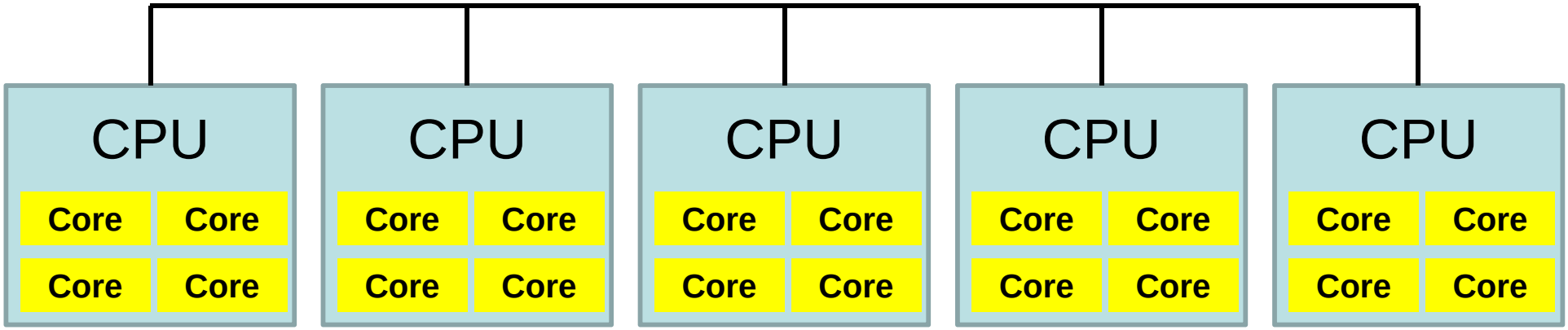
- GPU : Graphic Processing Unit
  - GPGPU: General Purpose GPU
  - O(10^2) cores
  - High Memory Bandwidth
  - Cheap
  - NO stand-alone operations
    - Host CPU needed
  - Programming: CUDA, OpenACC

- Intel Xeon/Phi: Manycore CPU
  - 60+ cores
  - High Memory Bandwidth
  - Unix, Fortran, C compiler
  - Host CPU needed in the 1st generation
    - Stand-alone is possible now (Knights Landing, KNL)

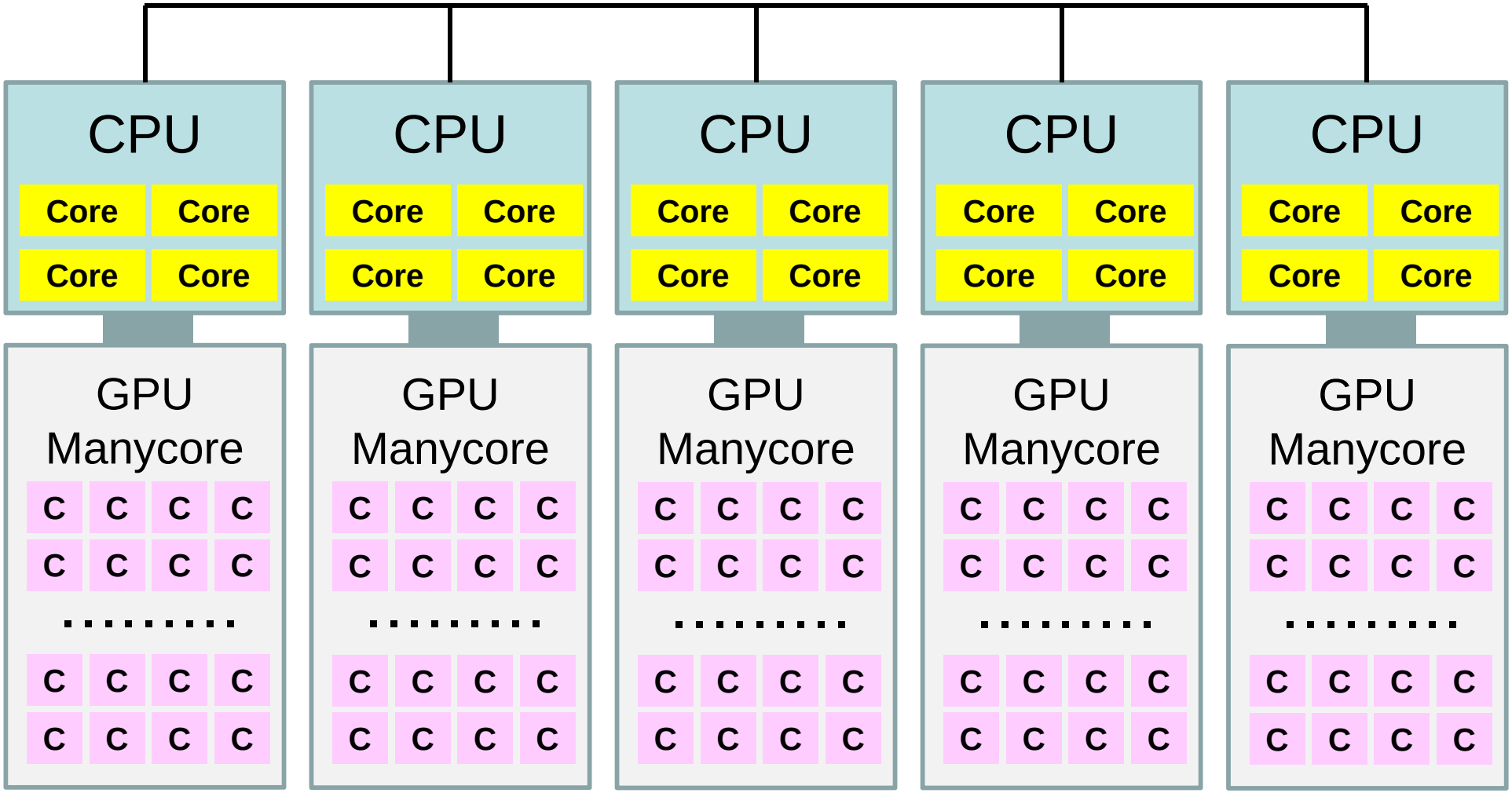


Parallel Supercomputers

Multicore CPU’s are connected through network



Supercomputers with Heterogeneous/Hybrid Nodes



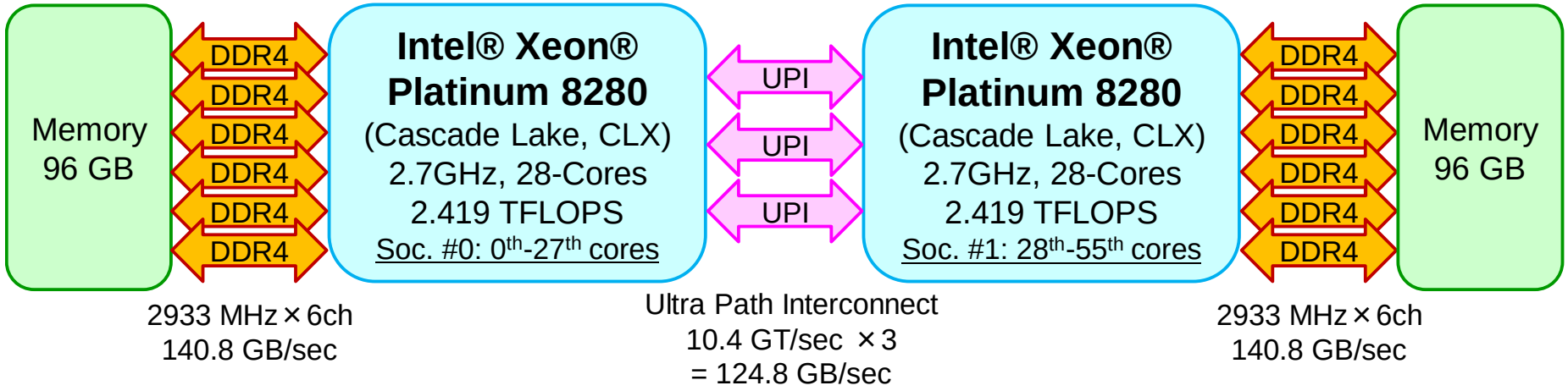
Performance of Supercomputers

- 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
    - $3 \times 10^9 \times 4(\text{or } 8) = 12(\text{or } 24) \times 10^9$ 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 Oakbridge-CX

Intel Xeon Plarinum 8280 (Cascade Lake, Intel Xeon SP)



- 2.7 GHz
  - 32 DP (Double Precision) FLOP operations per Clock
- Peak Performance (1 core)
  - 2.7 × 32= 86.4 GFLOPS
- Peak Performance
  - 1-Socket, 28 cores: 2,419.2 GFLOPS= 2.419 TFLOPS
  - 2-Sockets, 56 cores: 4,838.4 GFLOPS= 4.838 TFLOPS 1-Node

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
- Oakbridge-CX (OBCX) is 50th in the TOP 500 (November 2019)

- PFLOPS: Peta (=10^{15}) Floating OPerations per Sec.
- Exa-FLOPS (=10^{18}) will be attained in 2021 …

http://www.top500.org/

Benchmarks

- TOP 500 (Linpack, HPL(High Performance Linpack))
  - Direct Linear Solvers, FLOPS rate
  - Regular Dense Matrices, Continuous Memory Access
  - Computing Performance
- HPCG
  - Preconditioned Iterative Solvers, FLOPS rate
  - Irregular Sparse Matrices derived from FEM Applications with Many “0” Components
    - Irregular/Random Memory Access,
    - Closer to “Real” Applications than HPL
  - Performance of Memory, Communications
- Green 500
  - FLOPS/W rate for HPL (TOP500)

54th TOP500 List (Nov., 2019) Oakbridge-CX (OBCX) is 50th

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

http://www.top500.org/

	Site	Computer/Year Vendor	Cores	R_{max} (TFLOPS)	R_{peak} (TFLOPS)	Power (kW)
1	**Summit, 2018, USA** DOE/SC/Oak Ridge National Laboratory	IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	2,414,592	148,600 (= 148.6 PF)	200,795	**10,096**
2	**Sieera, 2018, USA** DOE/NNSA/LLNL	IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	1,572,480	94,640	125,712	**7,438**
3	**Sunway TaihuLight, 2016, China** National Supercomputing Center in Wuxi	Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	10,649,600	93,015	125,436	**15,371**
4	**Tianhe-2A, 2018, China** National Super Computer Center in Guangzhou	TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000	4,981,760	61,445	100,679	**18,482**
5	**Frontera, 2019, USA** Texas Advanced Computing Center	Dell C6420, Xeon Platinum 8280 28c 2.7GHz, Mellanox Infiniband HDR	448,448	23,516	38,746	
6	**Piz Daint, 2017, Switzerland** Swiss National Supercomputing Centre (CSCS)	Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100	387,872	21,230	27,154	**2,384**
7	**Trinity, 2017, USA** DOE/NNSA/LANL/SNL	Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect	979,072	20,159	41,461	**7,578**
8	**ABCI (AI Bridging Cloud Infrastructure), 2018, Japan** National Institute of Advanced Industrial Science and Technology (AIST)	PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR	391,680	19,880	32,577	**1,649**
9	**SuperMUC-NG, 2018, Germany** Leibniz Rechenzentrum	Lenovo, ThinkSystem SD650, Xeon Platinum 8174 24C 3.1GHz, Intel Omni-Path	305,856	19,477	26,874	
10	**Lassen, 2019, USA** DOE/NNSA/LLNL	IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta V100, Dual-rail Mellanox EDR Infiniband	288,288	18,200	23,047	
15	**Oakforest-PACS, 2016, Japan** Joint Center for Advanced High Performance Computing	PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path	556,104	13,556	24,913	2,719

HPCG Ranking (November, 2019)

	Computer	Cores	HPL Rmax (Pflop/s)	TOP500 Rank	HPCG (Pflop/s)
1	**Summit**	2,414,592	148,600	1	2.926
2	**Sierra**	1,572,480	94.640	2	1.796
3	**Trinity**	979,072	20,159	7	0.546
4	**ABCI**	391,680	19,880	8	0.509
5	**Piz Daint**	387,872	21.230	6	0.497
6	**Sunway TaihuLight**	10,649,600	93.015	3	0.481
7	**Nurion (KISTI, Korea)**	570,020	13.929	14	0.391
8	**Oakforest-PACS**	556,104	13.555	15	0.385
9	**Cori (NERSC/LBNL, USA)**	632,400	14.015	13	0.355
10	**Tera-100-2 (CEA, France)**	622,336	11.965	17	0.334

http://www.hpcg-benchmark.org/

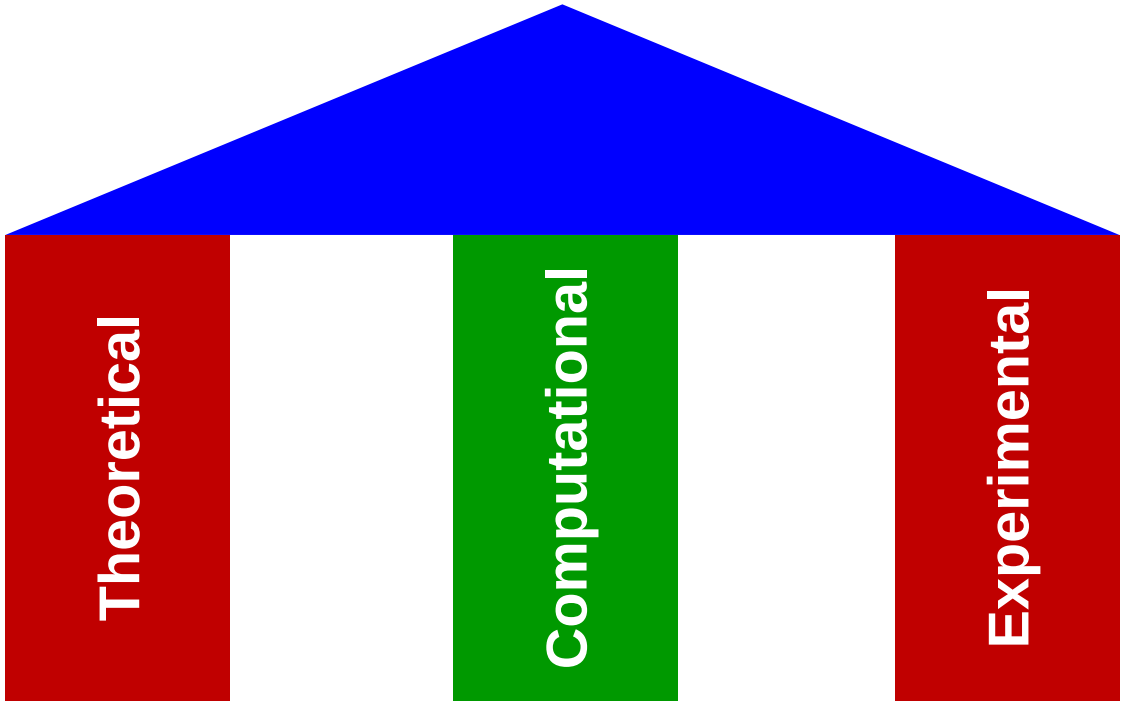
Green 500 Ranking (November, 2019)

http://www.top500.org/

	TOP 500 Rank	System	Cores	HPL Rmax (Pflop/s)	Power (MW)	GFLOPS/W
1	159	A64FX Prototype (Fugaku), Japan	36,864	1,999.5	118	16.876
2	420	NA-1, PEZY, Japan	1,271,040	1,303.2	80	16.256
3	24	AiMOS, IBM, USA	130,000	8,045.0	510	15.771
4	373	Satori, IBM, USA	23,040	1,464.0	94	15.574
5	1	Summit, USA	2,414,592	148,600	10,096	14.719
6	8	ABCI, Japan	391,680	19,880	1,649	14.423
7	494	MareNostrum P9 CTE, Spain	18,360	1,145	81	14.131
8	23	TSUBAME 3.0, Japan	135,828	8,125	792	13.704
9	11	PANGEA III, France	291,024	17,860	1,367	13.065
10	2	Sierra, USA	1,572,480	94,640	7,438	12.723
13	June'18	Reedbush-L, U.Tokyo, Japan	16,640	806	79	10.167
19		Reedbush-H, U.Tokyo, Japan	17,760	802	94	8.576

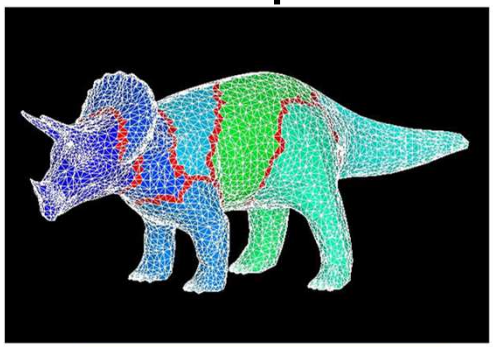
Computational Science
The 3rd Pillar of Science

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

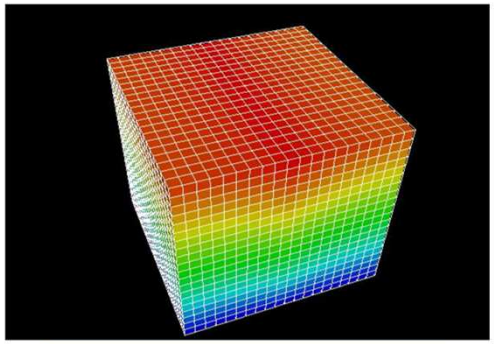


Methods for Scientific Computing

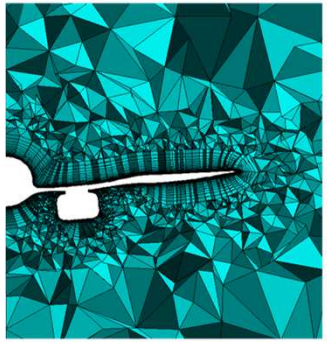
- Numerical solutions of PDE (Partial Diff. Equations)
- Grids, Meshes, Particles
  - Large-Scale Linear Equations
  - Finer meshes provide more accurate solutions



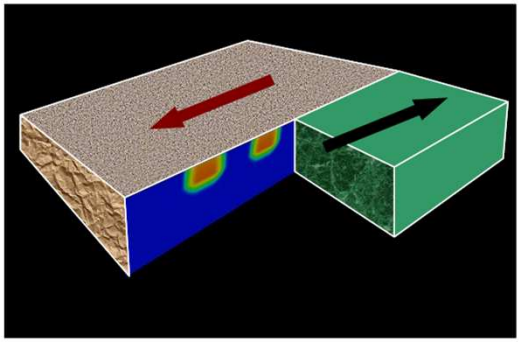
有限要素法
Finite Element Method
FEM



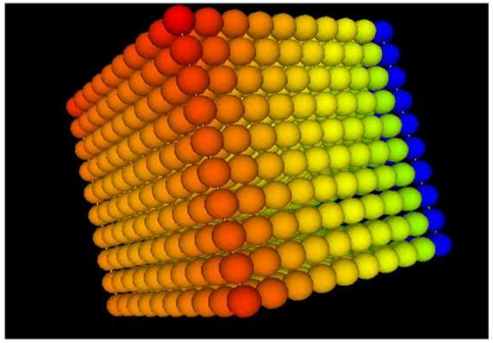
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



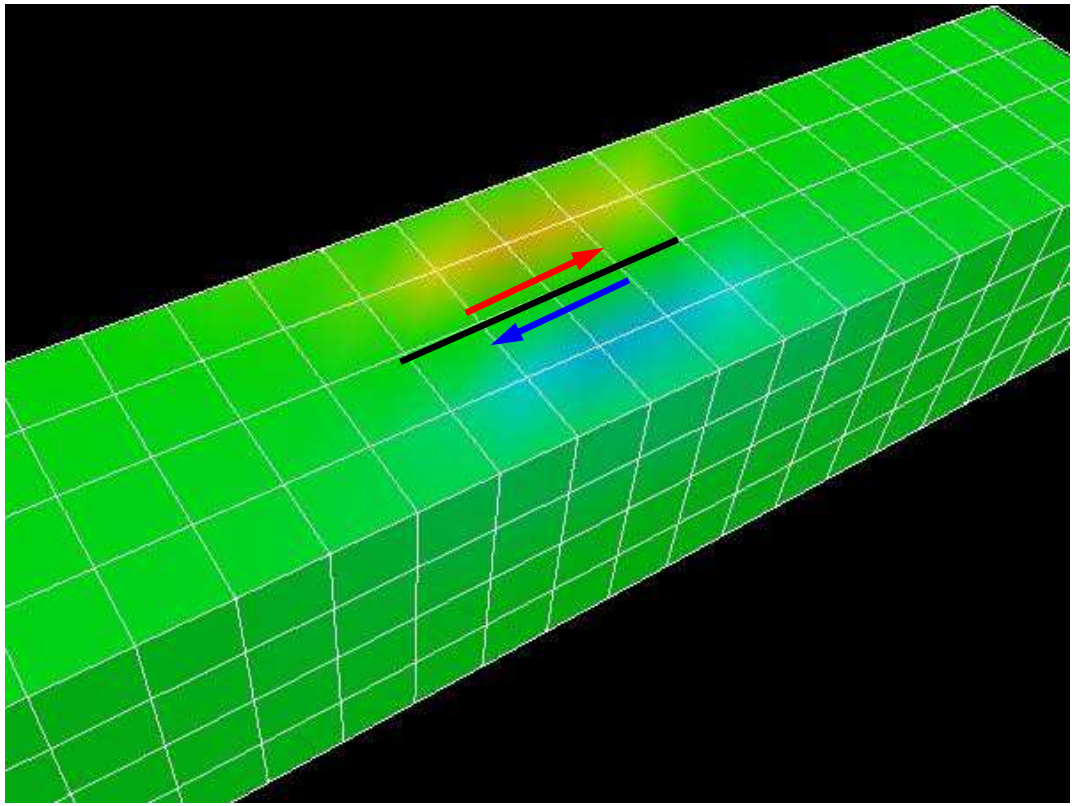
境界要素法
Boundary Element Method
BEM

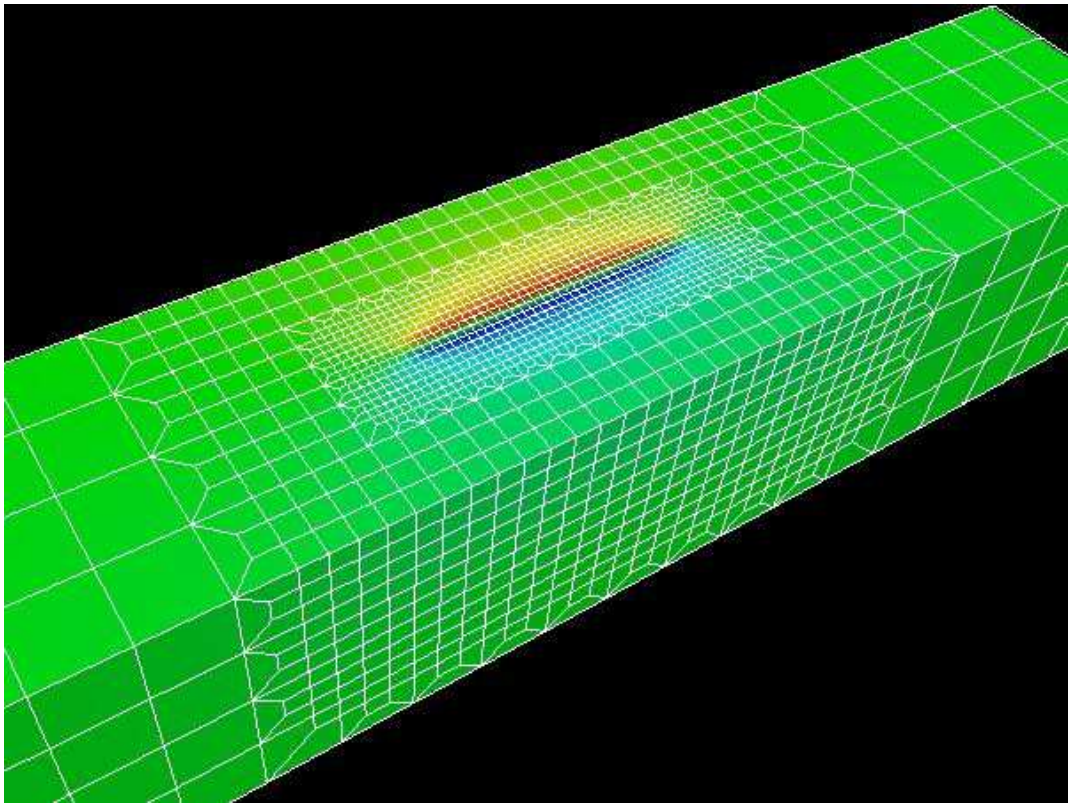


個別要素法
Discrete Element Method
DEM

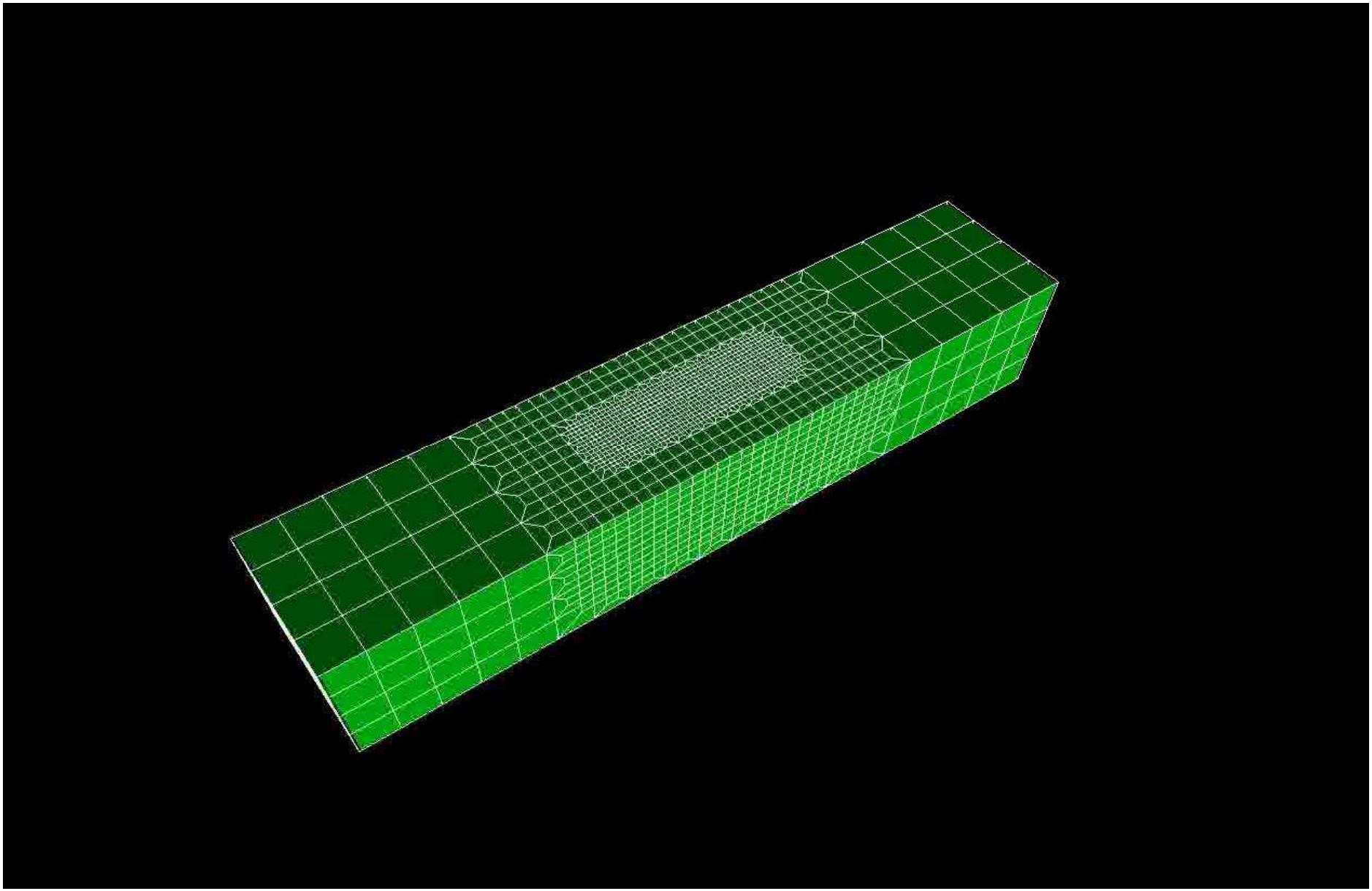
3D Simulations for Earthquake Generation Cycle San Andreas Faults, CA, USA

Stress Accumulation at Transcurrent Plate Boundaries

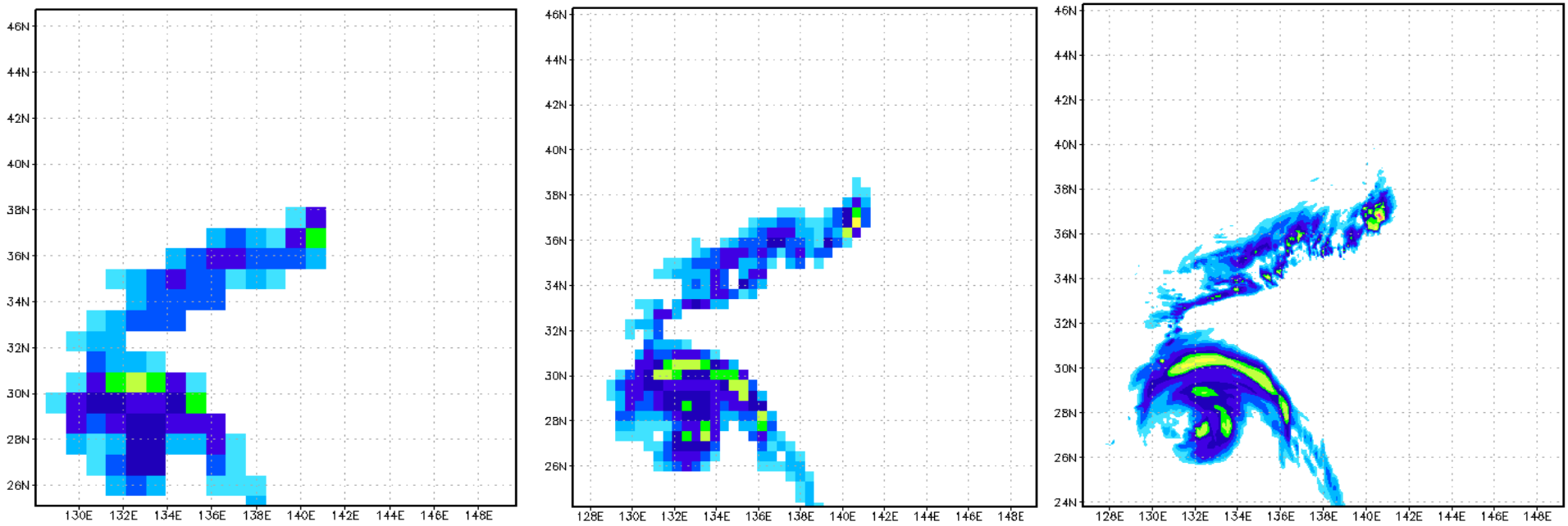




Adaptive FEM: High-resolution needed at meshes with large deformation (large accumulation)



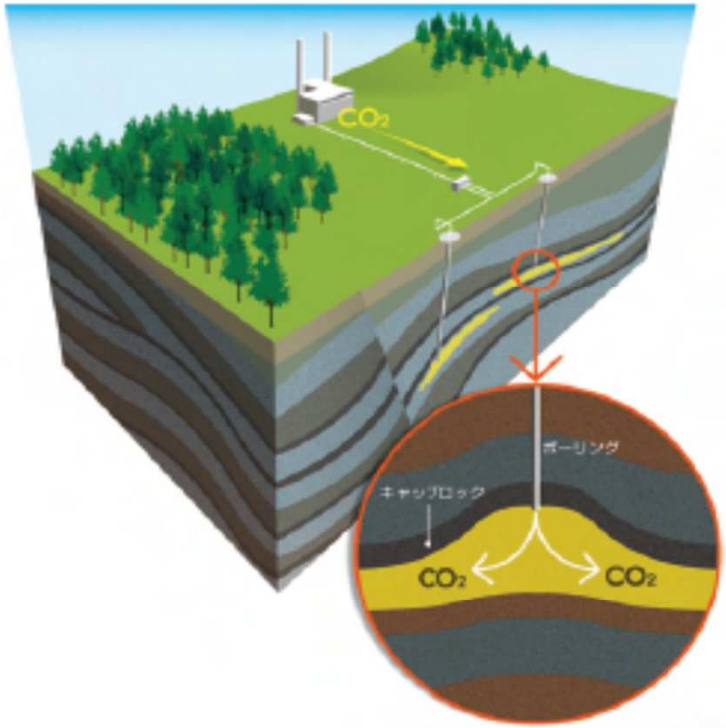
Typhoon Simulations by FDM
Effect of Resolution



Δh=100km **Δh=50km** **Δh=5km**

[JAMSTEC]

Simulation of Geologic CO_2 Storage



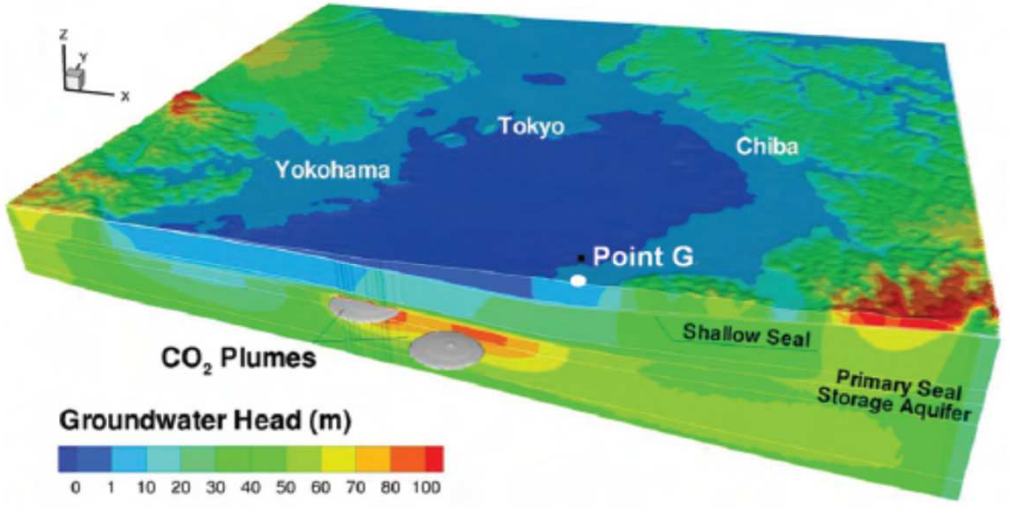
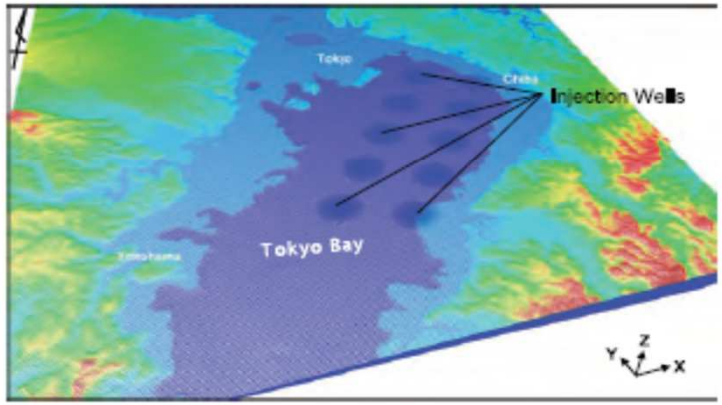
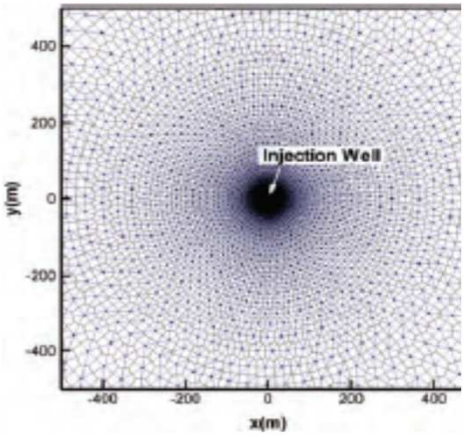
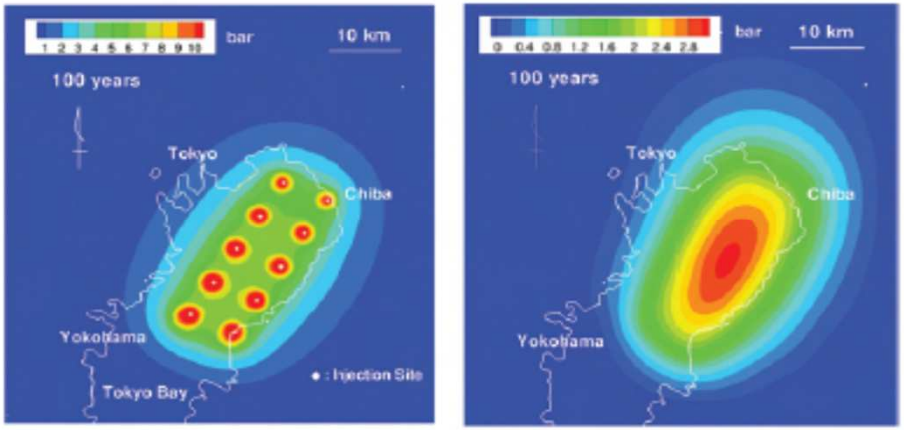


図 -4　CO_2 圧入後の地下水圧（全水頭換算）の分布（100 年後）







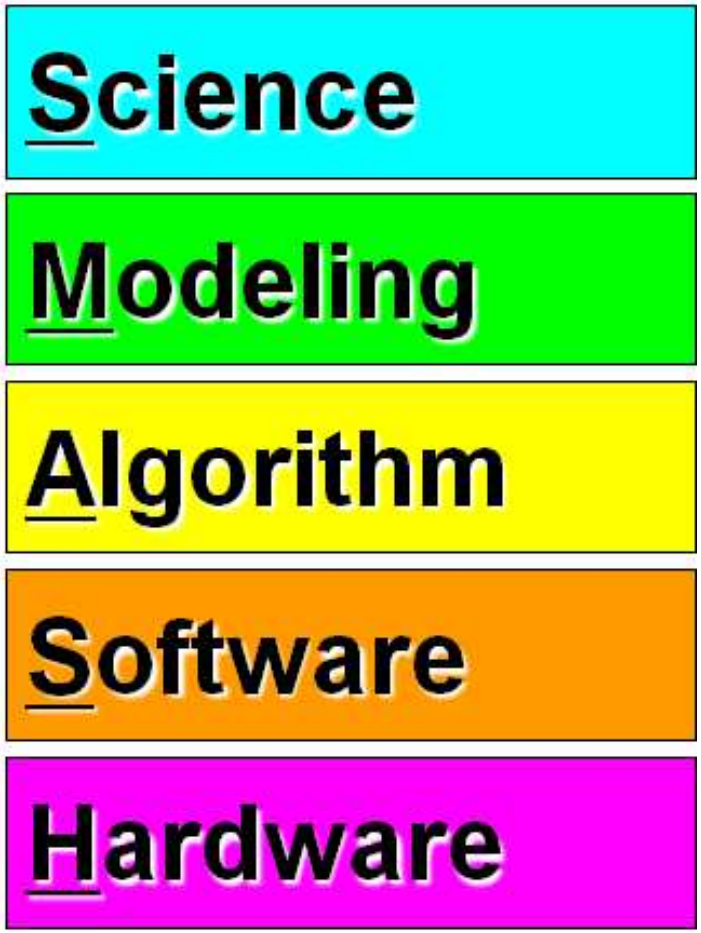
(a) 深部遮蔽層下面　　(b) 浅部遮蔽層下面

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

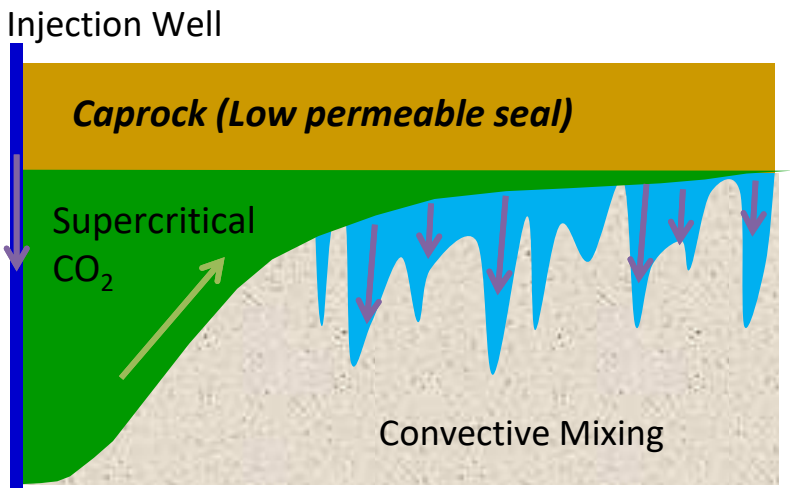
- International/Interdisciplinary Collaborations
  - Taisei (Science, Modeling)
  - Lawrence Berkeley National Laboratory, USA (Modeling)
  - Information Technology Center, the University of Tokyo (Algorithm, Software)
  - JAMSTEC (Earth Simulator Center) (Software, Hardware)
  - NEC (Software, Hardware)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



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

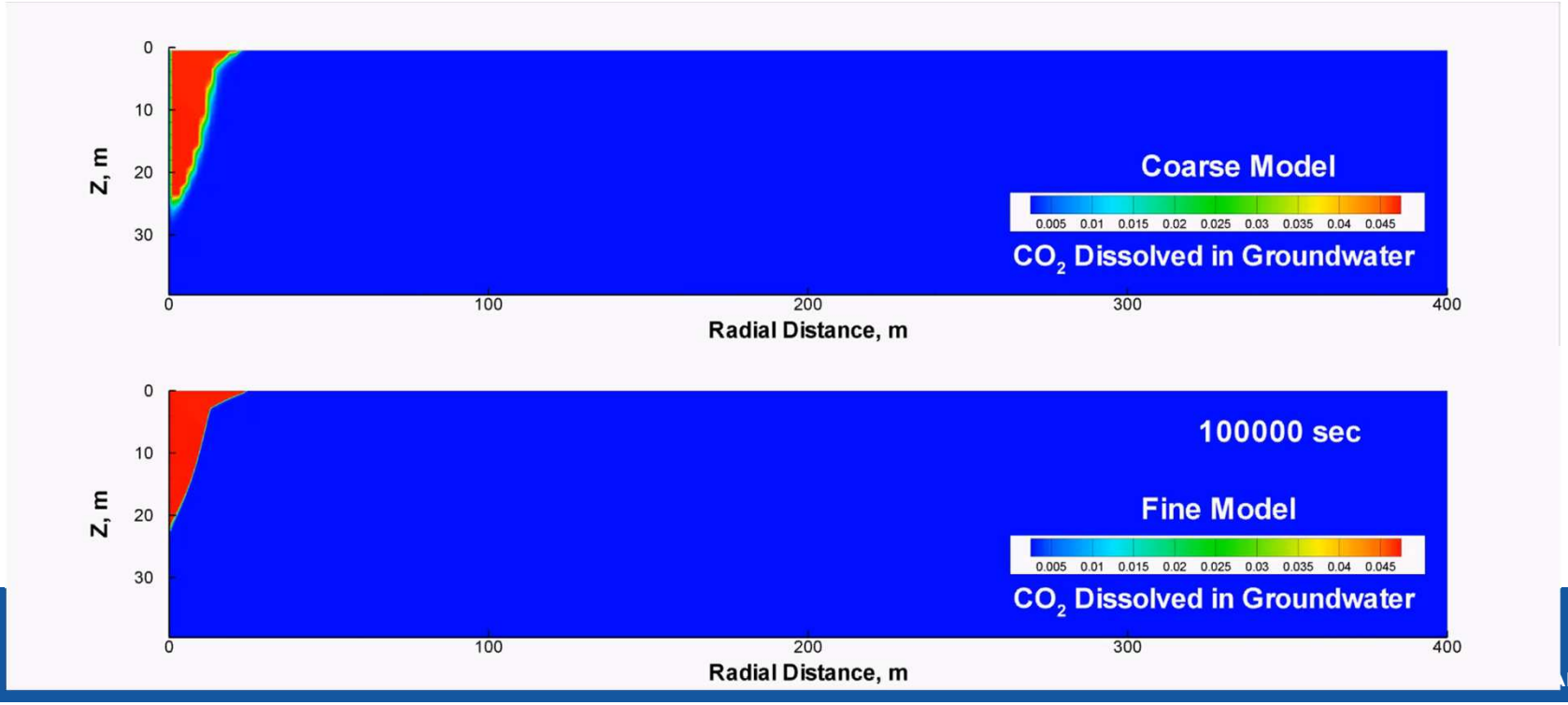
Diffusion-Dissolution-Convection Process

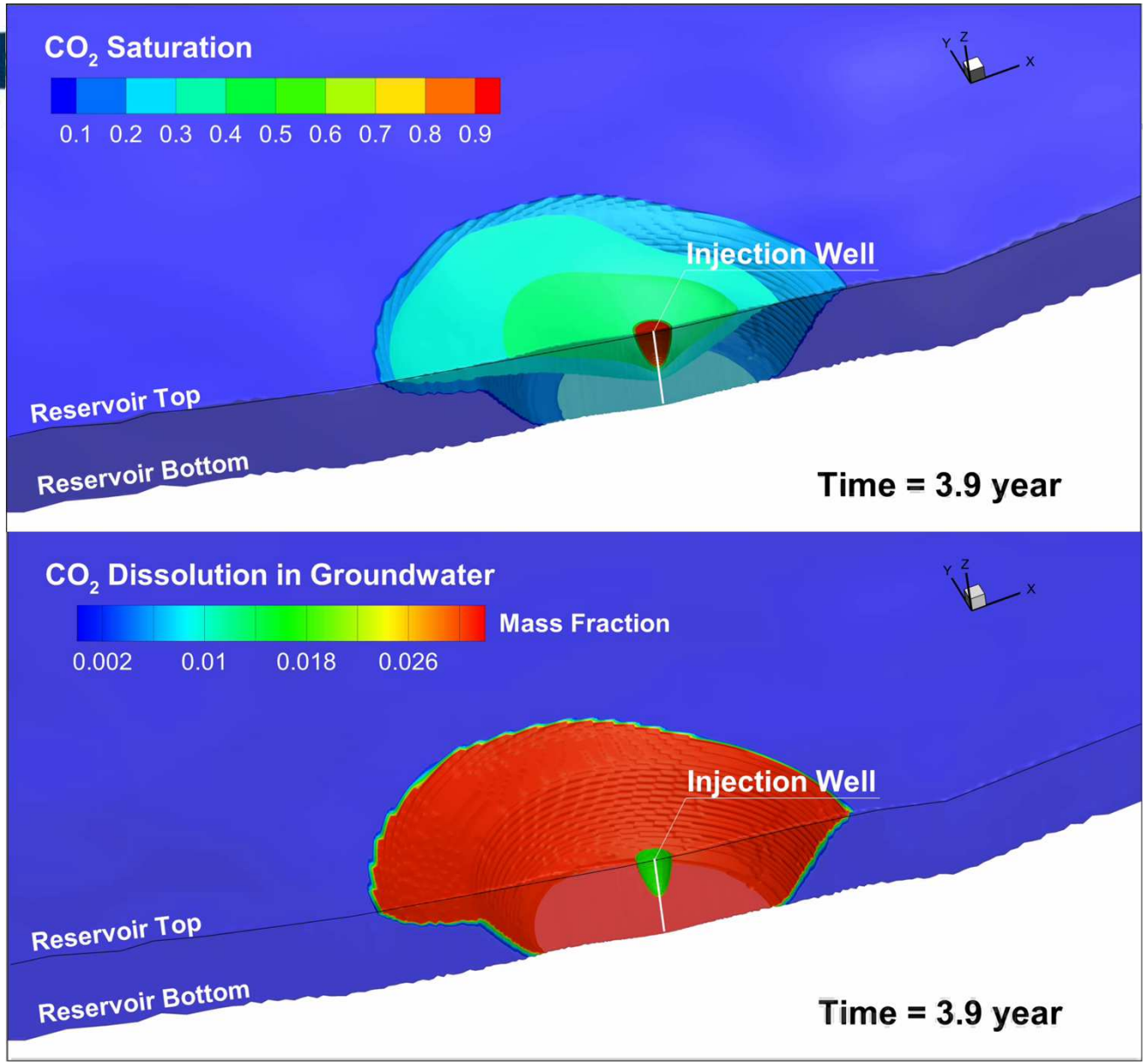


- Buoyant scCO_2 overrides onto groundwater
- Dissolution of CO_2 increases water density
- Denser fluid laid on lighter fluid
- Rayleigh-Taylor instability invokes convective mixing of groundwater

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.

Reservoir Condition

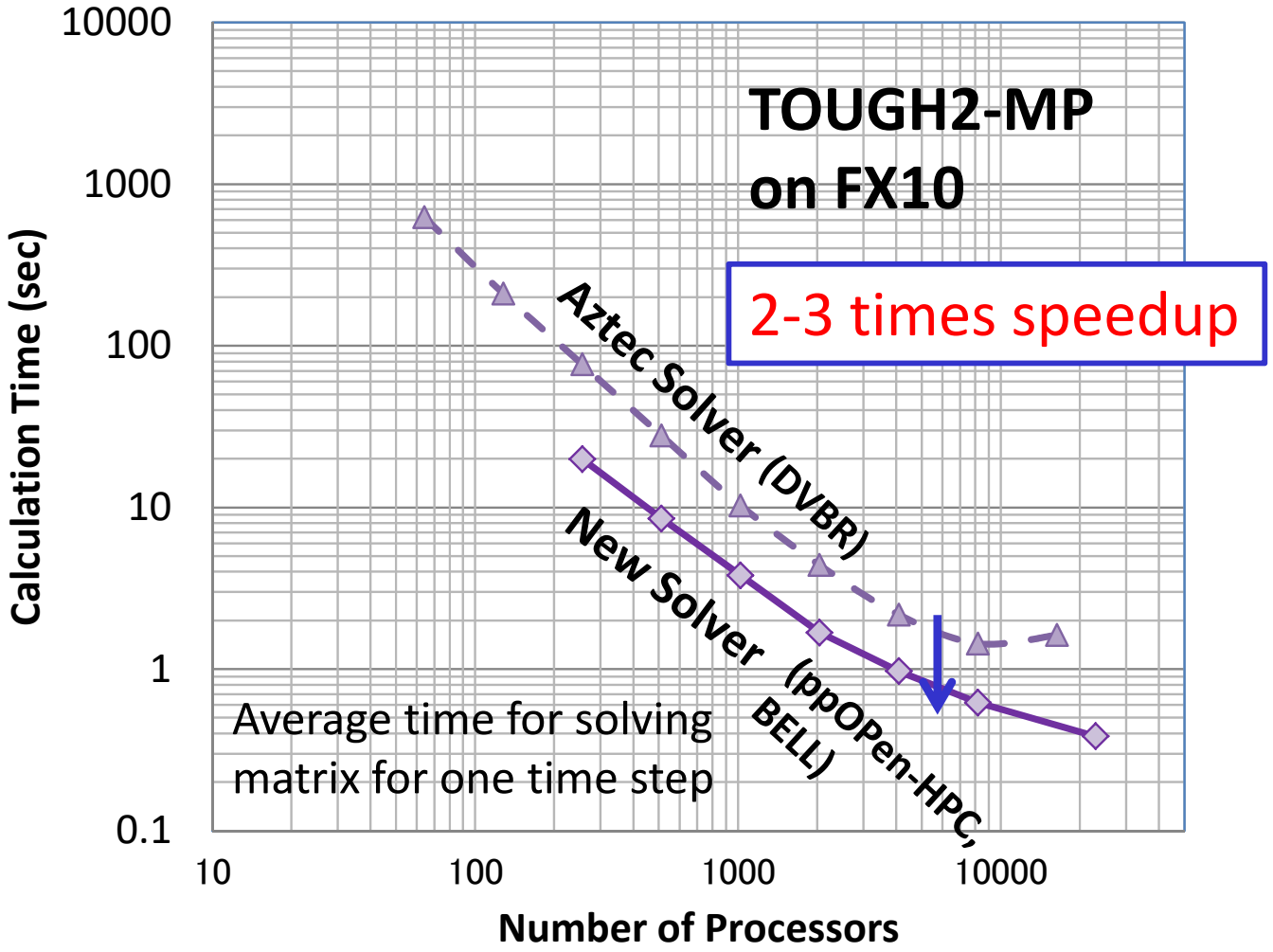
- Permeability: 100 md
- Porosity: 20%
- Pressure: 3MPa
- Temperature: 100°C
- Salinity: 15wt%

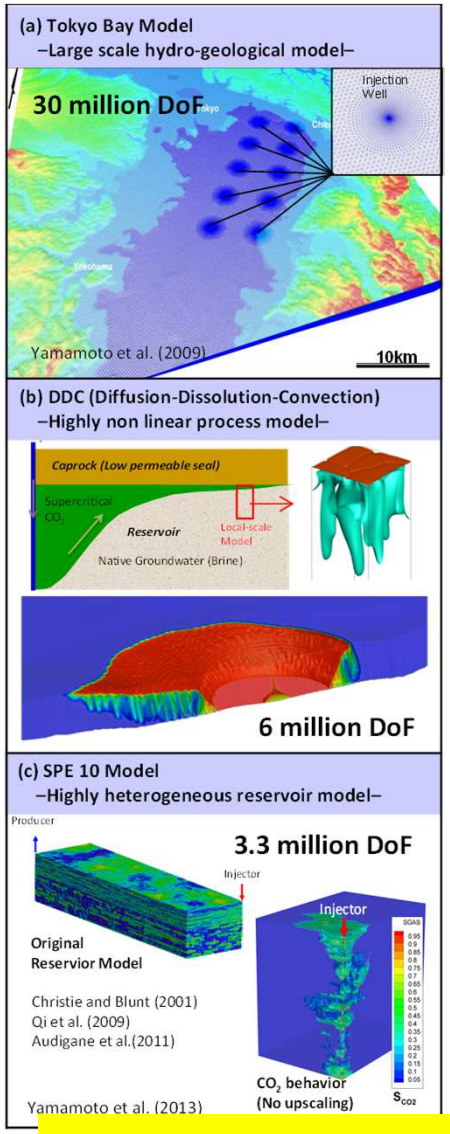
[Dr. Hajime Yamamoto, Taisei]

- 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)

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

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
    - Weak Scaling, Strong Scaling

- Target: Parallel FEM
- Supercomputers and Computational Science
- **Overview of the Class**
- Future Issues

Information of this Class

- Instructor
  - Kengo Nakajima (The University of Tokyo)
    - e-mail: nakajima(at)cc.u-tokyo.ac.jp
- Schedule
  - February 21-25, 2020
  - 7-Slots Each Day
  - **http://nkl.cc.u-tokyo.ac.jp/NTU2020/**
- Facilities
  - Oakbridge-CX (OBCX) System in ITC/U.Tokyo
  - Note PC by Each Student
    - Win, Mac, Linux
    - Paraview
    - Cygwin (for Win)

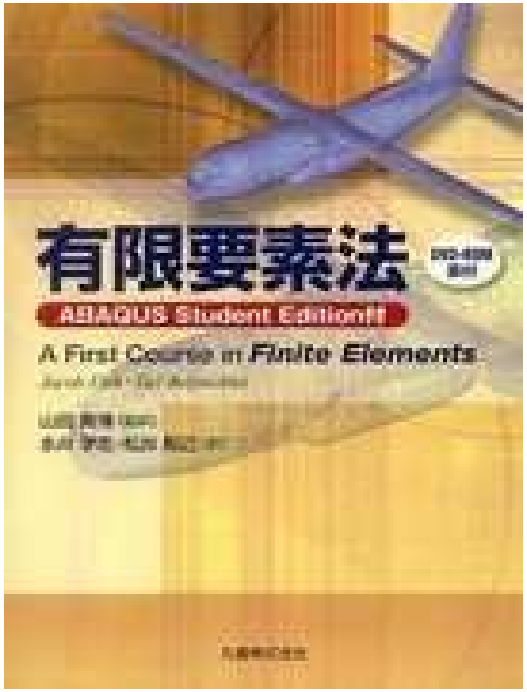
Prerequisites

- Knowledge and experiences in fundamental methods for numerical analysis (e.g. Gaussian elimination, SOR)
- Knowledge and experiences in UNIX/Linux
  - emacs, vi
- Experiences in programming using FORTRAN or C
- Fundamental Issues of FEM
  - http://nkl.cc.u-tokyo.ac.jp/19w/02-FEM/FEMintro.pdf

Data	Hour	Content
Feb.21 (F)	09:10-10:00	Introduction (1/2)
	10:10-11:00	Introduction (2/2)
	11:10-12:00	FEM (1/6)
	13:10-14:00	FEM (2/6)
	14:10-15:00	FEM (3/6)
	15:10-16:00	FEM (4/6)
	16:10-17:00	Exercise (Optional)
Feb.22 (Sa)	09:10-10:00	FEM (5/6)
	10:10-11:00	FEM (6/6)
	11:10-12:00	Exercise
	13:10-14:00	Parallel FEM
	14:10-15:00	Login to OBCX
	15:10-16:00	MPI (1/6)
	16:10-17:00	Exercise (Optional)
Feb.23 (Su)	09:10-10:00	MPI (2/6)
	10:10-11:00	MPI (3/6)
	11:10-12:00	Exercise
	13:10-14:00	MPI Practice (1/3)
	14:10-15:00	MPI (4/6)
	15:10-16:00	MPI (5/6)
	16:10-17:00	Exercise (Optional)

Date	Hour	Content
Feb.24 (M)	09:10-10:00	MPI (6/6)
	10:10-11:00	Exercise
	11:10-12:00	Exercise
	13:10-14:00	MPI Practice (2/3)
	14:10-15:00	MPI Practice (3/3)
	15:10-16:00	Exercise
	16:10-17:00	Parallel FEM (1/4)
Feb.25 (T)	09:10-10:00	Parallel FEM (2/4)
	10:10-11:00	Parallel FEM (3/4)
	11:10-12:00	Parallel FEM (4/4)
	13:10-14:00	Exercise
	14:10-15:00	OpenMP/MPI Hybrid (1/2)
	15:10-16:00	OpenMP/MPI Hybrid (2/2)
	16:10-17:00	Exercise (Optional)

References



- Fish, Belytschko, A First Course in Finite Elements, Wiley, 2007
  - Japanese version is also available
  - “ABAQUS Student Edition” included
- Smith et al., Programming the Finite Element Method (4th edition), Wiley, 2004
  - Parallel FEM included
- Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover, 2000

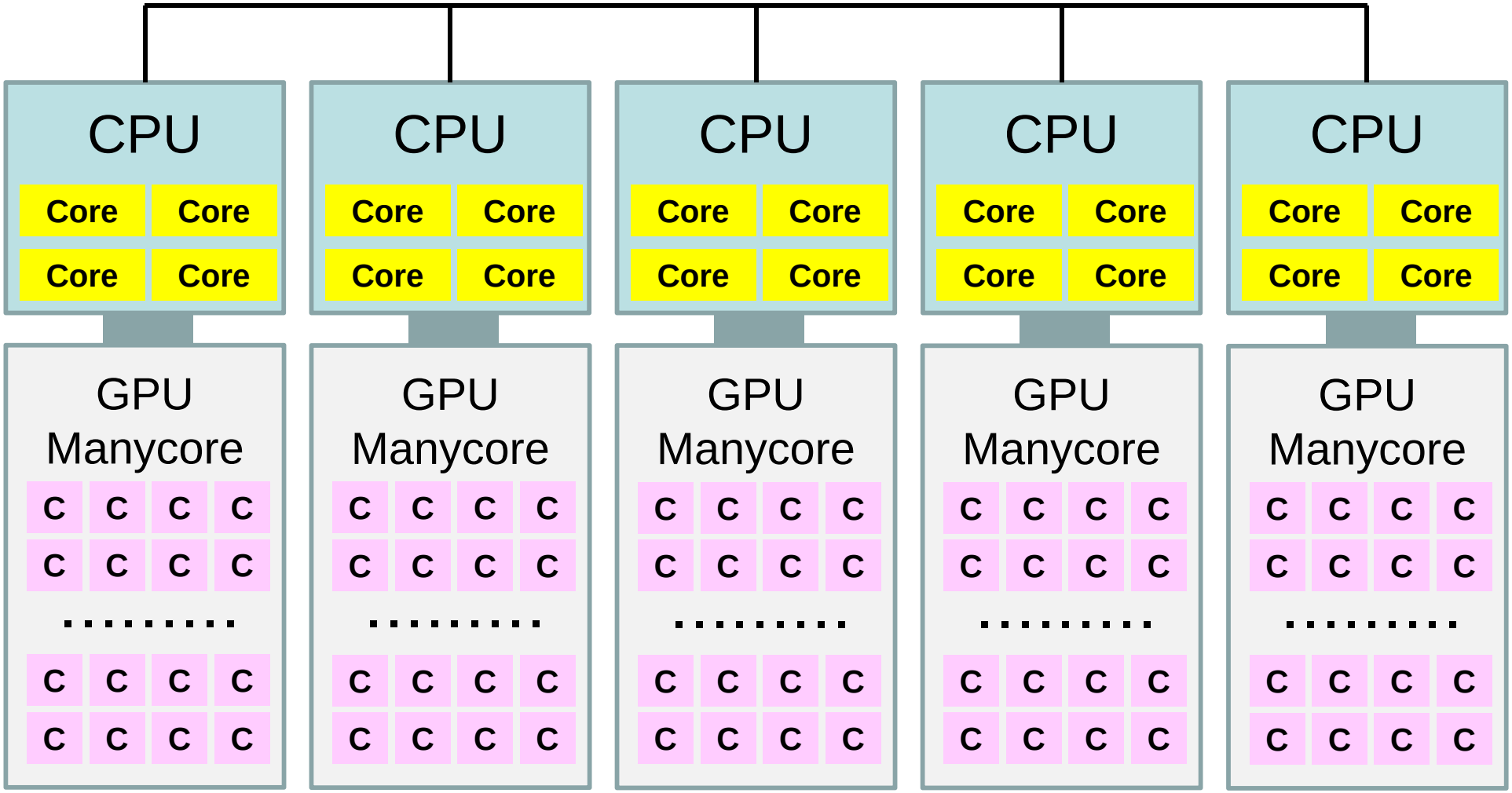
- Target: Parallel FEM
- Supercomputers and Computational Science
- Overview of the Class
- **Future Issues**

Key-Issues towards Appl./Algorithms on Exa-Scale Systems

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

- Hybrid/Heterogeneous Architecture
  - Multicore + GPU/Manycores (Intel MIC/Xeon Phi)
    - Data Movement, Hierarchy of Memory
- Communication/Synchronization Reducing Algorithms
- Mixed Precision Computation
- Auto-Tuning/Self-Adapting
- Fault Resilient Algorithms
- Reproducibility of Results

Supercomputers with Heterogeneous/Hybrid Nodes

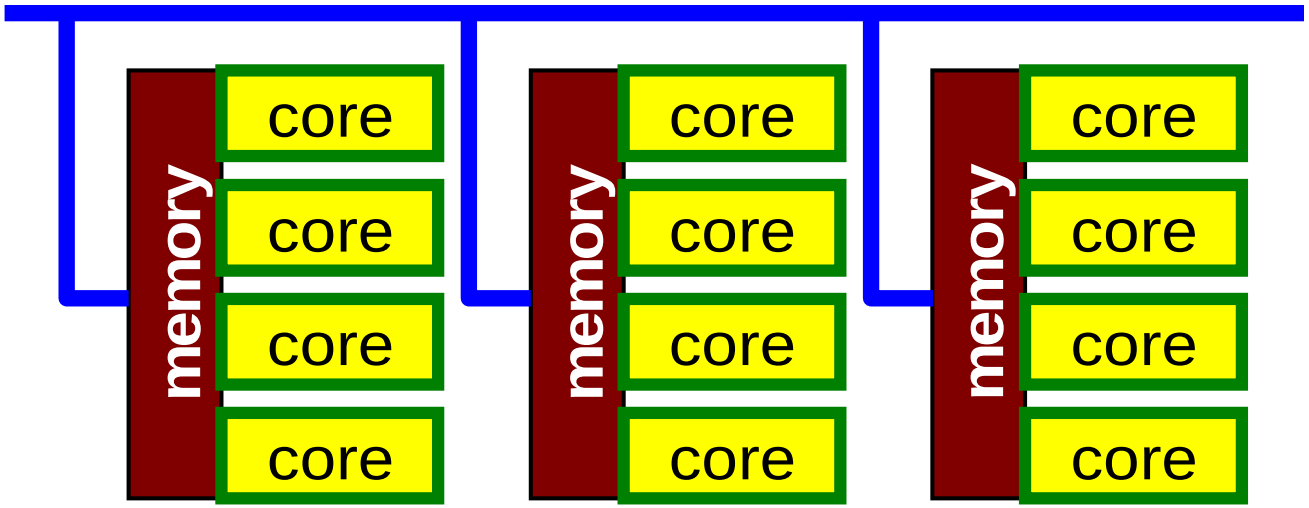


Hybrid Parallel Programming Model is essential for Post-Peta/Exascale Computing

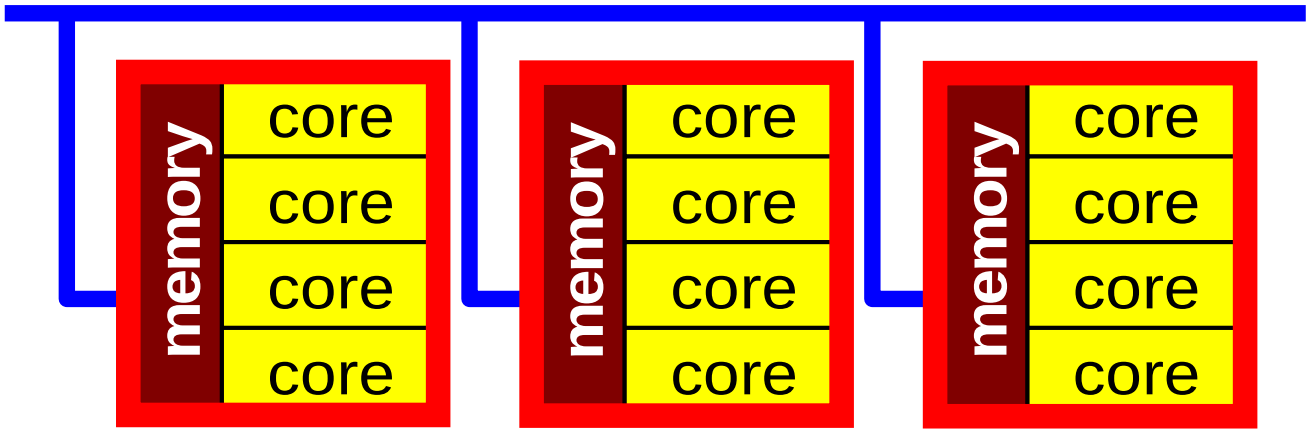
- Message Passing (e.g. MPI) + Multi Threading (e.g. OpenMP, CUDA, OpenCL, OpenACC etc.)
- Expectations for Hybrid
  - Number of MPI processes (and sub-domains) to be reduced
  - O(10^8-10^9)-way MPI might not scale in Exascale Systems
  - Easily extended to Heterogeneous Architectures
    - CPU+GPU, CPU+Manycores (e.g. Intel MIC/Xeon Phi)
    - MPI+X: OpenMP, OpenACC, CUDA, OpenCL

Flat MPI vs. Hybrid

Flat-MPI：Each PE -> Independent



Hybrid：Hierarchal Structure



In this class...

- Very brief introduction of OpenMP and OpenMP/MPI Hybrid Parallel Programming Model will be provided.

- MPI is essential for large-scale scientific computing. If you want to something new using supercomputers, you must learn MPI, then OpenMP.

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
```

Parallel Programming Models

- Multicore/Manycore Clusters
  - MPI + OpenMP and (Fortan/C/C++)
- Multicore + GPU
  - GPU needs host CPU
  - MPI and [(Fortan/C/C++) + CUDA, OpenCL]
    - complicated,
  - MPI and [(Fortran/C/C++) with OpenACC]
    - close to MPI + OpenMP and (Fortran/C/C++)