Introduction to Parallel Programming for Multicore/Manycore Clusters

Introduction

Kengo Nakajima & Tetsuya Hoshino
Information Technology Center
The University of Tokyo

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

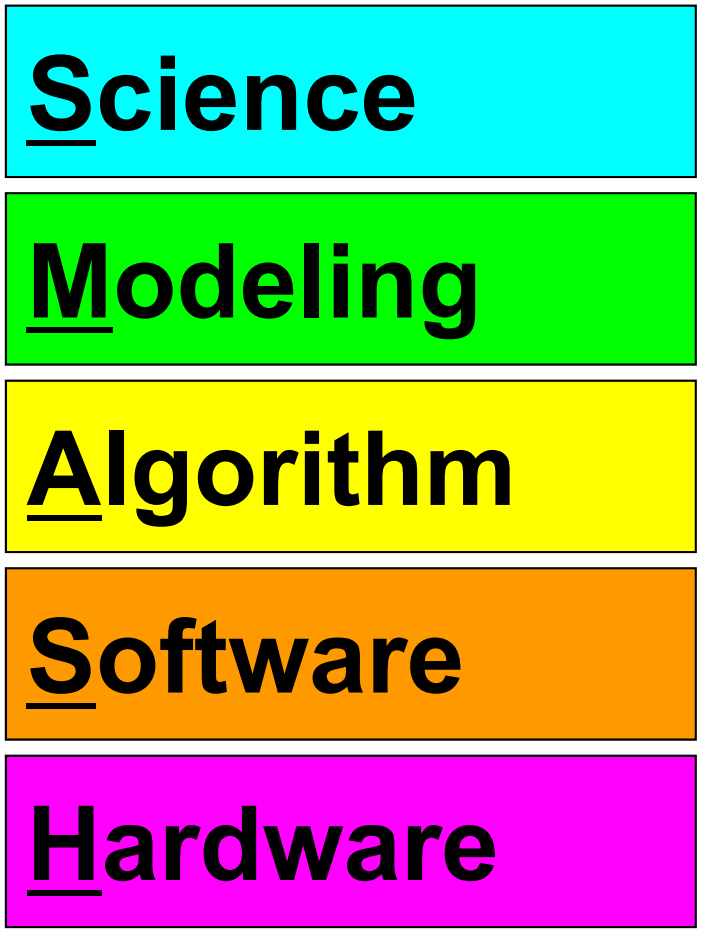
Algorithm

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 ?

This Class ...



- **Target: Parallel FVM (Finite-Volume Method) using OpenMP**
- Science: 3D Poisson Equations
- Modeling: FVM
- Algorithm: Iterative Solvers etc.

- You have to know many components to learn FVM, 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)

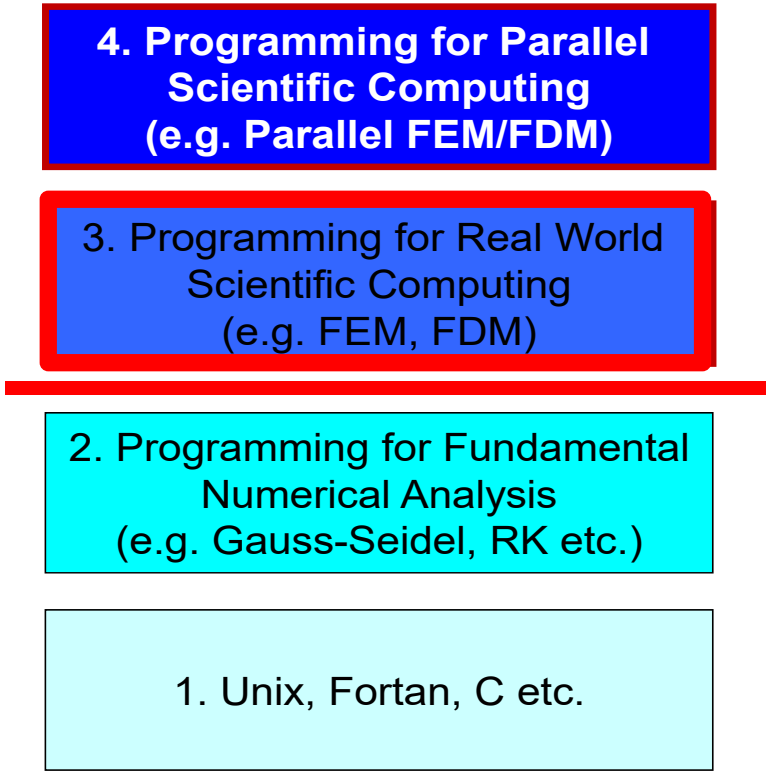
Big gap here !!

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

Unix, Fortan, C etc.

The third step is important !

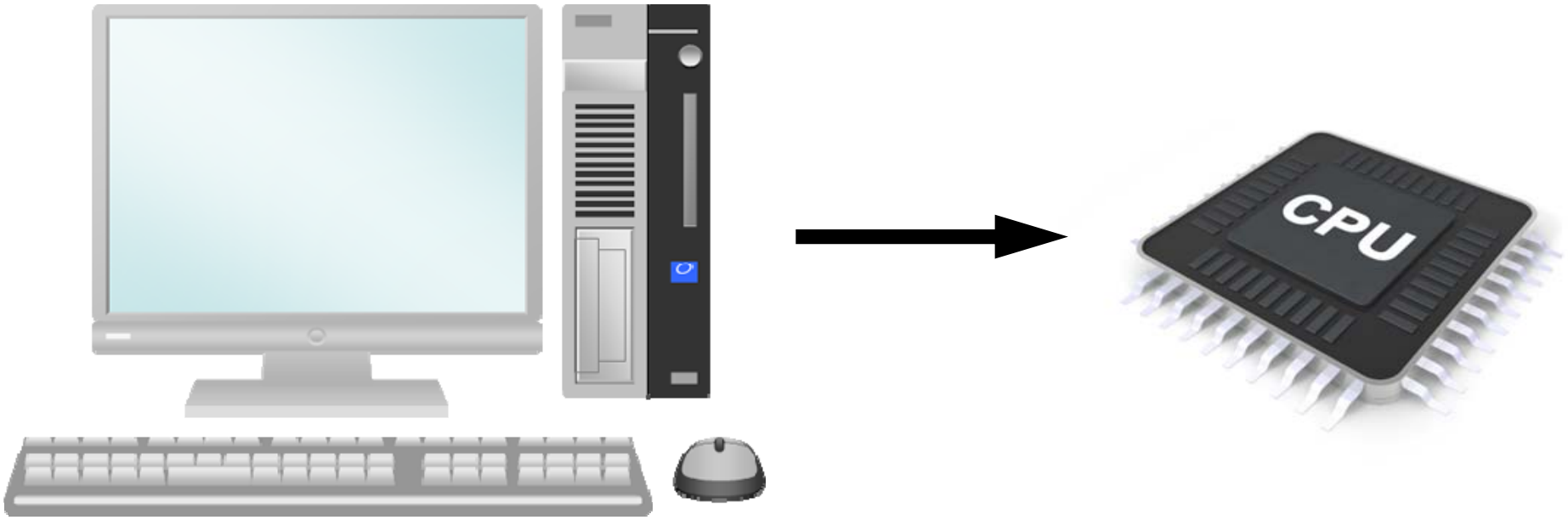
- How to parallelize applications ?
  - How to extract parallelism ?
  - If you understand methods, algorithms, and implementations of the original code, it's easy.
  - "Data-structure" is important



- How to understand the code ?
  - Reading the application code !!
  - It seems primitive, but very effective.
  - In this class, "reading the source code" is encouraged.
  - 3: FVM, 4: Parallel FVM

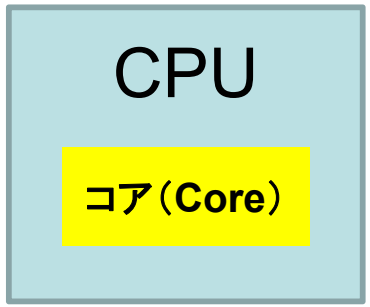
- **Supercomputers and Computational Science**
- Overview of the Class
- Future Issues

Computer & CPU

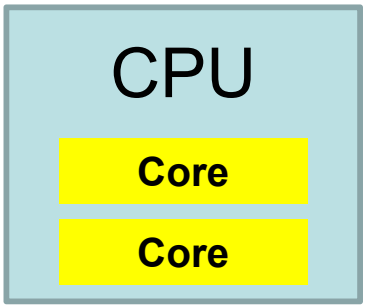


- Central Processing Unit（中央処理装置）:CPU
- CPU’s used in PC and Supercomputers are based on same architecture
- GHz: Clock Rate
  - Frequency: Number of operations by CPU per second
    - GHz -> 10^9 operations/sec
  - Simultaneous 4-8 instructions per clock

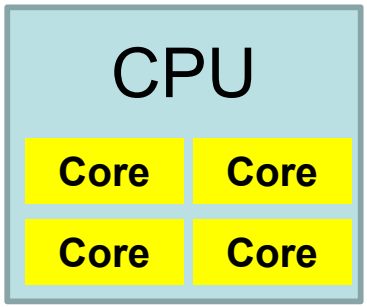
Multicore CPU



Single Core
1 cores/CPU

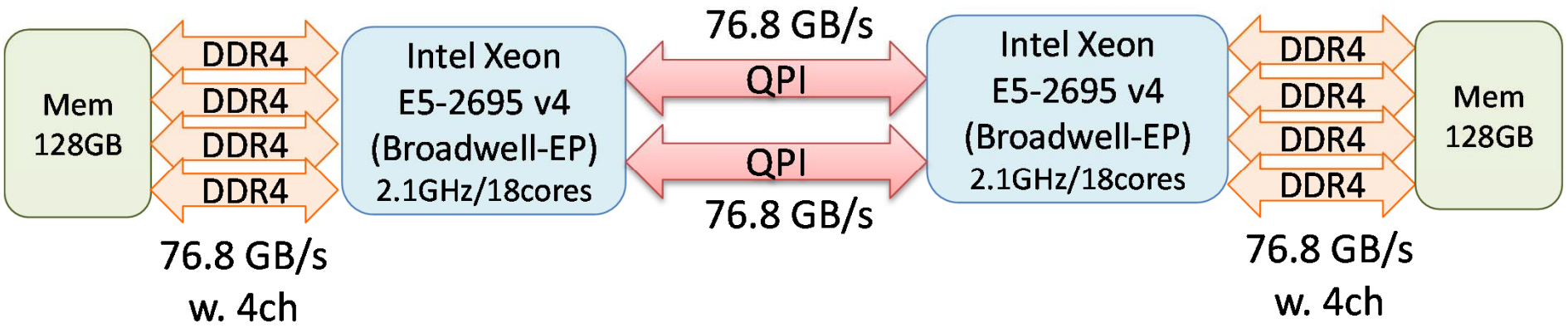


Dual Core
2 cores/CPU



Quad Core
4 cores/CPU

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



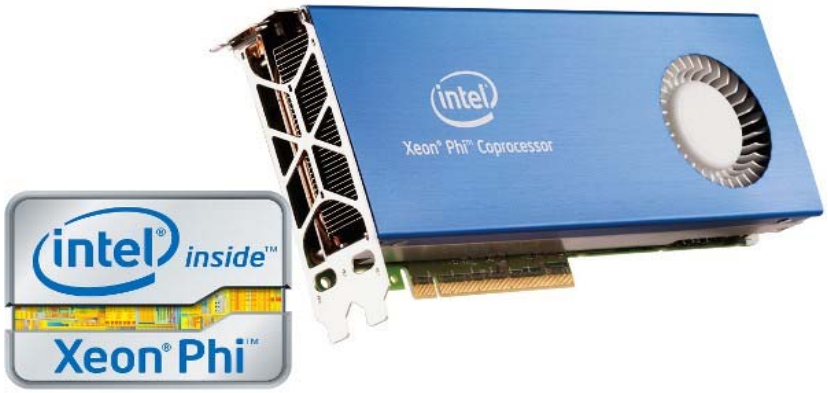
- GPU: Manycore
  - $O(10^1)$-$O(10^2)$ cores
- More and more cores
  - Parallel computing

- Reedbush-U: 18 cores x 2
  - Intel Xeon Broadwell-EP

GPU/Manycores

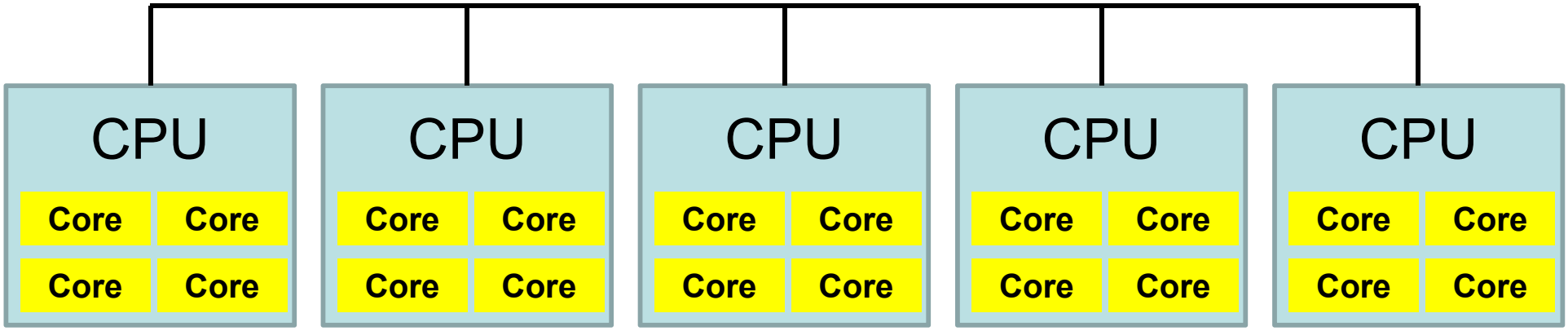
- GPU : Graphic Processing Unit
  - GPGPU: General Purpose GPU
  - $O(10^2)$ cores
  - High Memory Bandwidth
  - Cheap
  - NO stand-alone operations
    - Host CPU needed
  - Programming: CUDA, OpenACC

- 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)

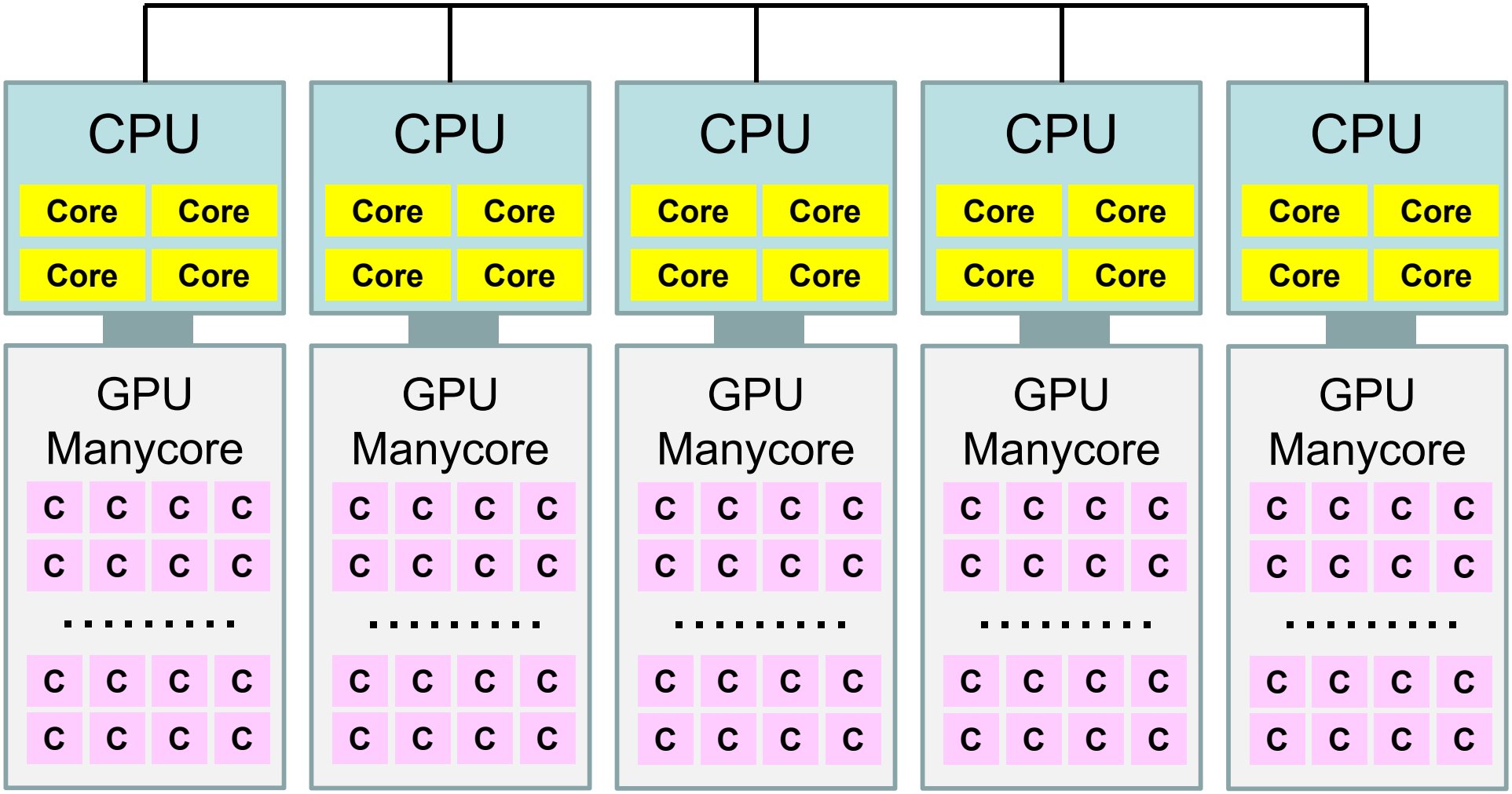


Parallel Supercomputers

Multicore CPU's are connected through network



Supercomputers with Heterogeneous/Hybrid Nodes



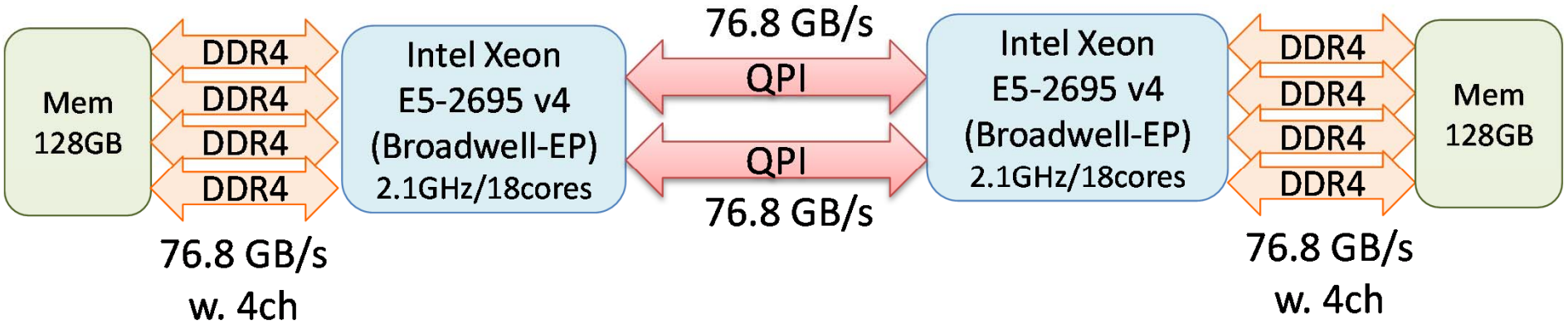
Performance of Supercomputers

- 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 Reedbush-U

Intel Xeon E5-2695 v4 (Broadwell-EP)

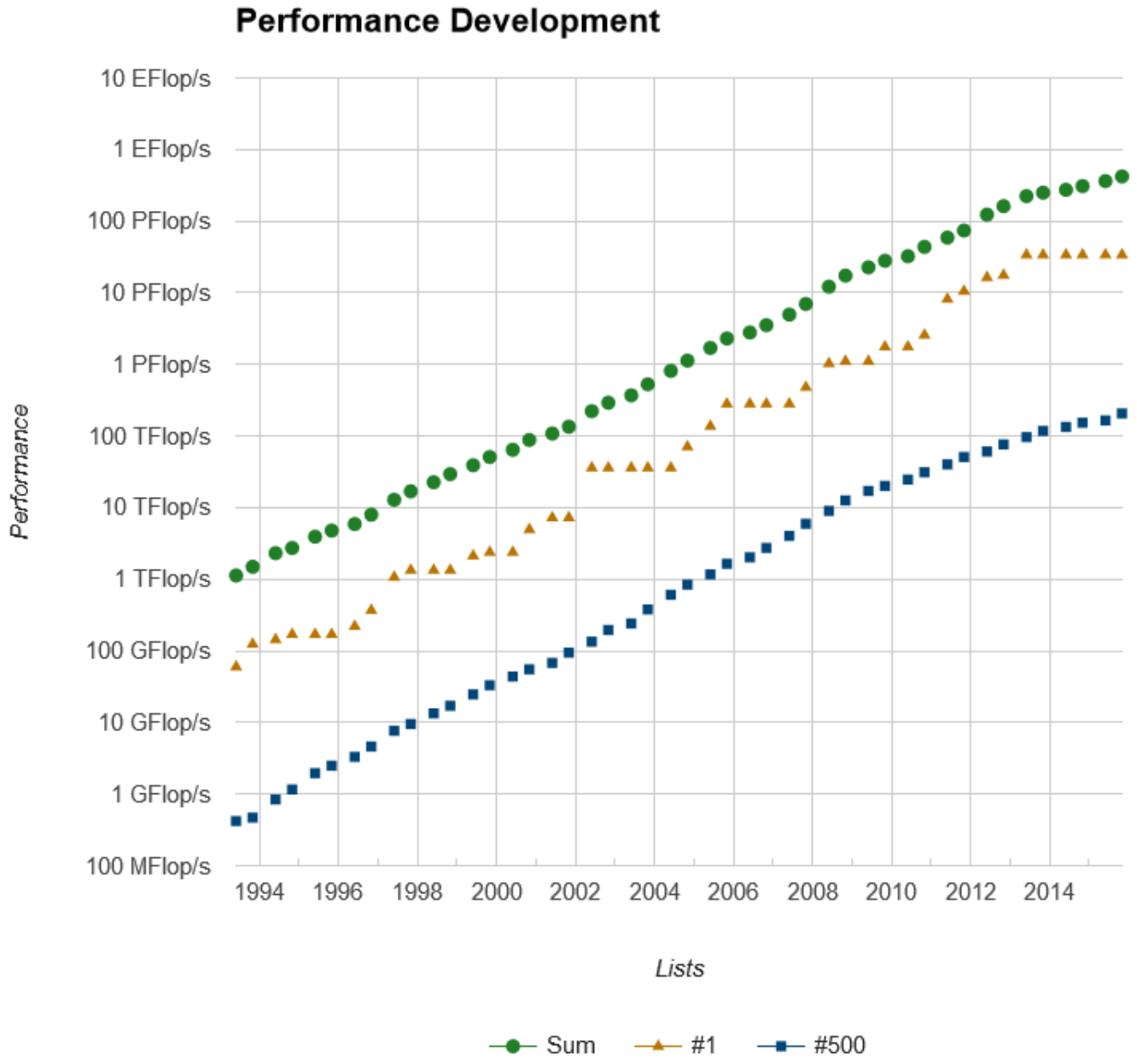


- 2.1 GHz
  - 16 DP (Double Precision) FLOP operations per Clock
- Peak Performance (1 core)
  - 2.1 × 16= 33.6 GFLOPS
- Peak Performance
  - 1-Socket, 18 cores: 604.8 GFLOPS
  - 2-Sockets, 36 cores: 1,209.6 GFLOPS 1-Node

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



- PFLOPS: Peta (=10^{15}) Floating OPerations per Sec.
- Exa-FLOPS (=10^{18}) will be attained after 2023 …

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)

49th TOP500 List (June, 2017)

	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	Swiss Natl. Supercomputer Center, Switzerland	**Piz Daint** Cray XC30/NVIDIA P100, 2013 Cray	361,760	19,590	33,863	2,272
4	Oak Ridge National Laboratory, USA	**Titan** Cray XK7/NVIDIA K20x, 2012 Cray	560,640	17,590	27,113	8,209
5	Lawrence Livermore National Laboratory, USA	**Sequoia** BlueGene/Q, 2011 IBM	1,572,864	17,173	20,133	7,890
6	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
7	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
8	RIKEN AICS, Japan	**K computer**, SPARC64 VIIIfx , 2011 Fujitsu	705,024	10,510	11,280	12,660
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/

HPCG Ranking (June, 2017)

	Computer	Cores	HPL Rmax (Pflop/s)	TOP500 Rank	HPCG (Pflop/s)	Peak
1	**K computer**	705,024	10.510	8	0.6027	5.3%
2	**Tianhe-2 (MilkyWay-2)**	3,120,000	33.863	2	0.5801	1.1%
3	**Sunway TaihuLight**	10,649,600	93.015	1	0.4808	0.4%
4	**Piz Daint**	361,760	19.590	3	0.4767	1.9%
5	**Oakforest-PACS**	557,056	13.555	7	0.3855	1.5%
6	**Cori**	632,400	13.832	6	0.3554	1.3%
7	**Sequoia**	1,572,864	17.173	5	0.3304	1.6%
8	**Titan**	560,640	17.590	4	0.3223	1.2%
9	**Trinity**	301,056	8.101	10	0.1826	1.6%
10	**Pleiades** – NASA/SGI	243,008	5.952	15	0.1752	2.5%

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

Green 500 Ranking (November, 2016)

	Site	Computer	CPU	HPL Rmax (Pflop/s)	TOP500 Rank	Power (MW)	GFLOPS/W
1	NVIDIA Corporation	DGX SATURNV	NVIDIA DGX-1, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100	3.307	28	0.350	9.462
2	Swiss National Supercomputing Centre (CSCS)	Piz Daint	Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100	9.779	8	1.312	7.454
3	RIKEN ACCS	Shoubu	ZettaScaler-1.6 etc.	1.001	116	0.150	6.674
4	National SC Center in Wuxi	Sunway TaihuLight	Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	93.01	1	15.37	6.051
5	SFB/TR55 at Fujitsu Tech. Solutions GmbH	QPACE3	PRIMERGY CX1640 M1, Intel Xeon Phi 7210 64C 1.3GHz, Intel Omni-Path	0.447	375	0.077	5.806
6	JCAHPC	Oakforest-PACS	PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path	1.355	6	2.719	4.986
7	DOE/SC/Argonne National Lab.	Theta	Cray XC40, Intel Xeon Phi 7230 64C 1.3GHz, Aries interconnect	5.096	18	1.087	4.688
8	Stanford Research Computing Center	XStream	Cray CS-Storm, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR, Nvidia K80	0.781	162	0.190	4.112
9	ACCMS, Kyoto University	Camphor 2	Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect	3.057	33	0.748	4.087
10	Jefferson Natl. Accel. Facility	SciPhi XVI	KOI Cluster, Intel Xeon Phi 7230 64C 1.3GHz, Intel Omni-Path	0.426	397	0.111	3.837

http://www.top500.org/

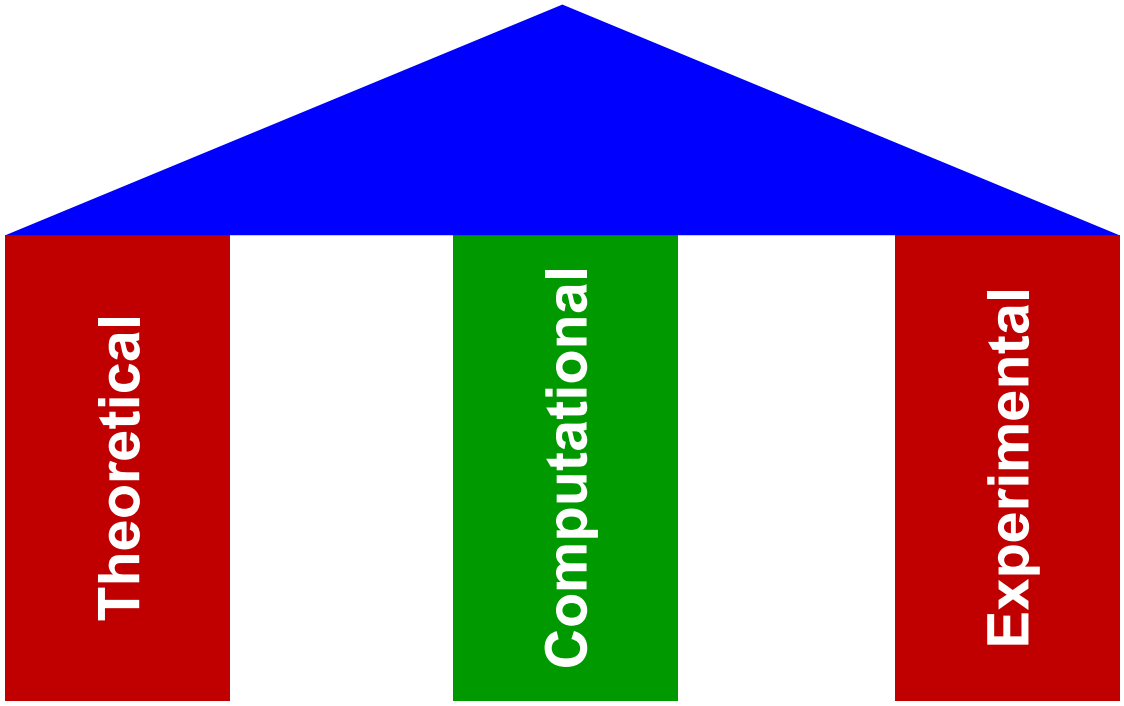
Green 500 Ranking (June, 2017)

	Site	Computer	CPU	HPL Rmax (Pflop/s)	TOP500 Rank	Power (MW)	GFLOPS/ W
1	Tokyo Tech.	TSUBAME3.0	SGI ICE XA, IP139-SXM2, Xeon E5-2680v4, NVIDIA Tesla P100 SXM2, HPE	1,998.0	61	142	14.110
2	Yahoo Japan	kukai	ZettaScaler-1.6, Xeon E5-2650Lv4,, NVIDIA Tesla P100 , Exascalar	460.7	465	33	14.046
3	AIST, Japan	AIST AI Cloud	NEC 4U-8GPU Server, Xeon E5-2630Lv4, NVIDIA Tesla P100 SXM2 , NEC	961.0	148	76	12.681
4	CAIP, RIKEN, JAPAN	RAIDEN GPU subsystem -	NVIDIA DGX-1, Xeon E5-2698v4, NVIDIA Tesla P100 , Fujitsu	635.1	305	60	10.603
5	Univ. Cambridge, UK	Wilkes-2 -	Dell C4130, Xeon E5-2650v4, NVIDIA Tesla P100 , Dell	1,193.0	100	114	10.428
6	Swiss Natl. SC. Center (CSCS)	Piz Daint	Cray XC50, Xeon E5-2690v3, NVIDIA Tesla P100 , Cray Inc.	19,590.0	3	2,272	10.398
7	JAMSTEC, Japan	Gyoukou,	ZettaScaler-2.0 HPC system, Xeon D-1571, PEZY-SC2 , ExaScalar	1,677.1	69	164	10.226
8	Inst. for Env. Studies, Japan	GOSAT-2 (RCF2)	SGI Rackable C1104-GP1, Xeon E5-2650v4, NVIDIA Tesla P100 , NSSOL/HPE	770.4	220	79	9.797
9	Facebook, USA	Penguin Relion	Xeon E5-2698v4/E5-2650v4, NVIDIA Tesla P100 , Acer Group	3,307.0	31	350	9.462
10	NVIDIA, USA	DGX Saturn V	Xeon E5-2698v4, NVIDIA Tesla P100 , Nvidia	3,307.0	32	350	9.462
11	ITC, U.Tokyo, Japan	Reedbush-H	SGI Rackable C1102-GP8, Xeon E5-2695v4, NVIDIA Tesla P100 SXM2 , HPE	802.4	203	94	8.575

http://www.top500.org/

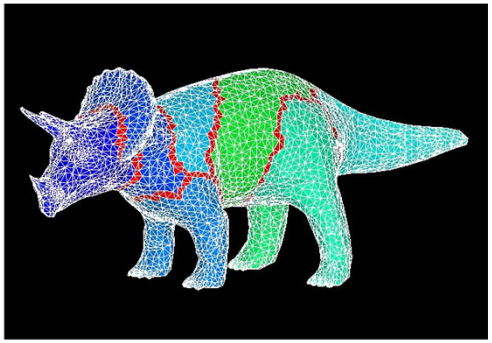
Computational Science
The 3rd Pillar of Science

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

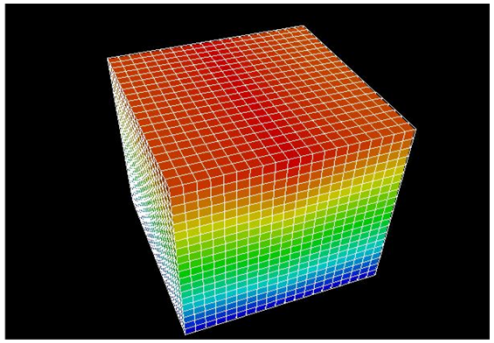


Methods for Scientific Computing

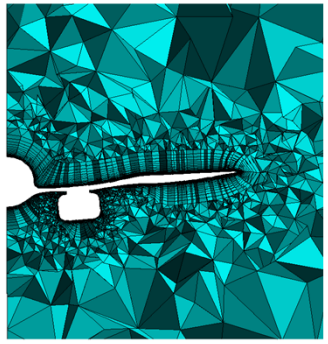
- Numerical solutions of PDE (Partial Diff. Equations)
- Grids, Meshes, Particles
  - Large-Scale Linear Equations
  - Finer meshes provide more accurate solutions



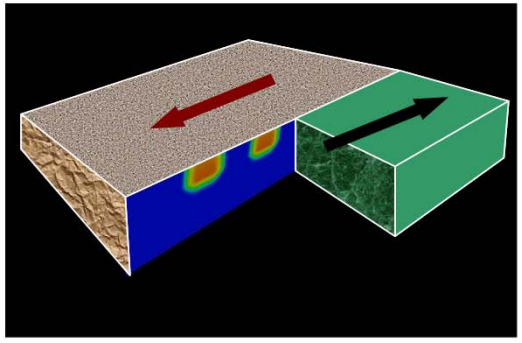
有限要素法
Finite Element Method
FEM



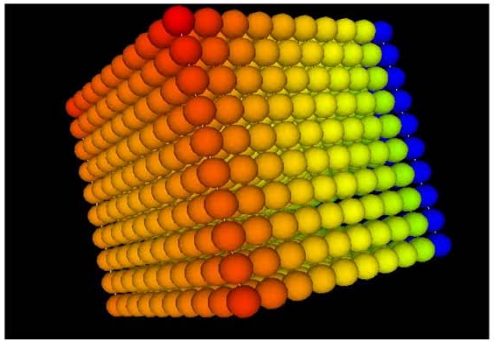
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



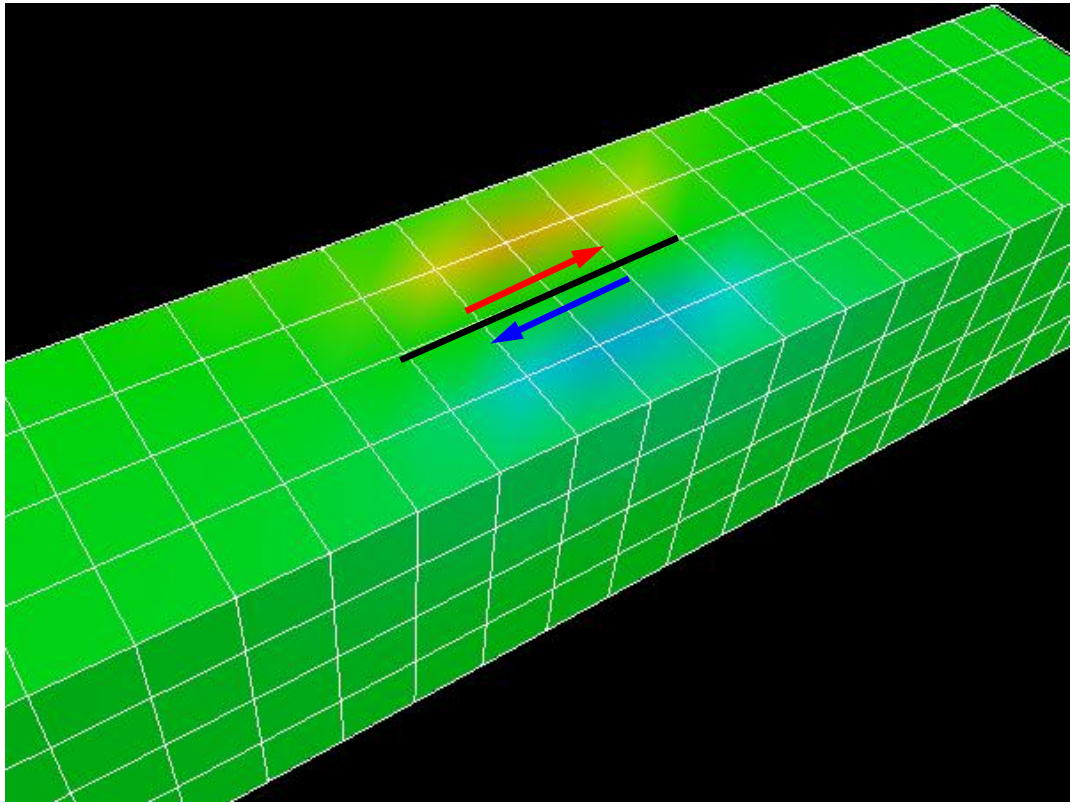
境界要素法
Boundary Element Method
BEM

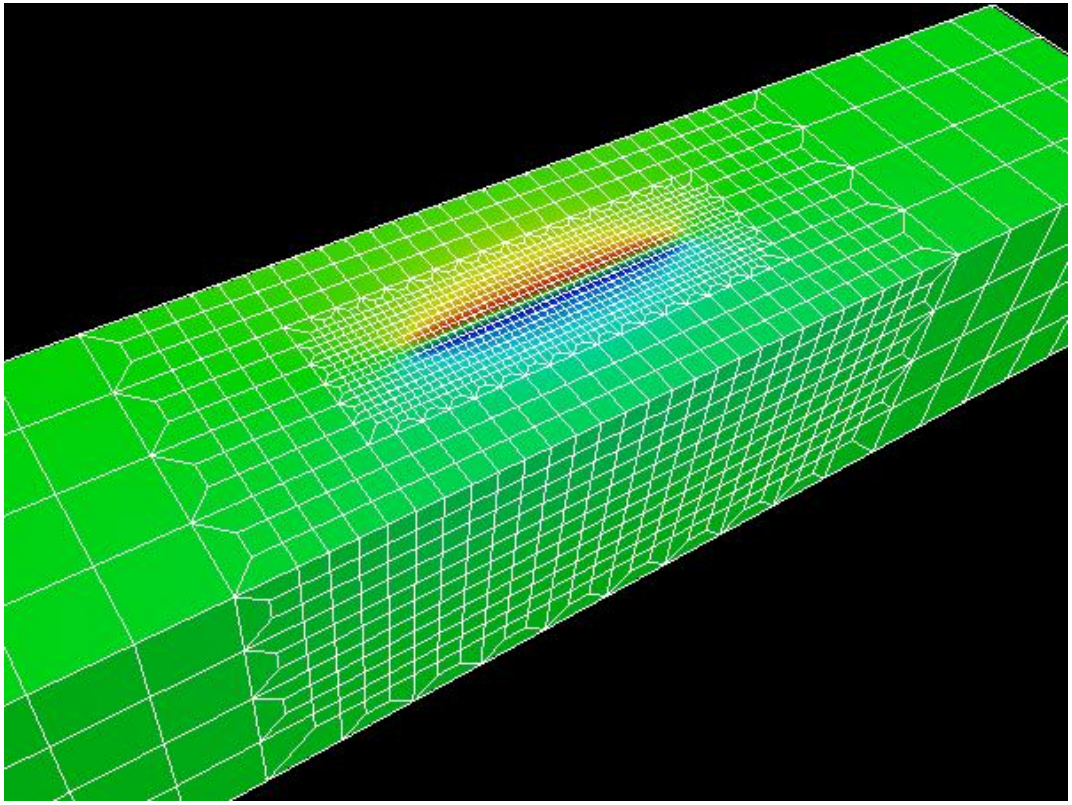


個別要素法
Discrete Element Method
DEM

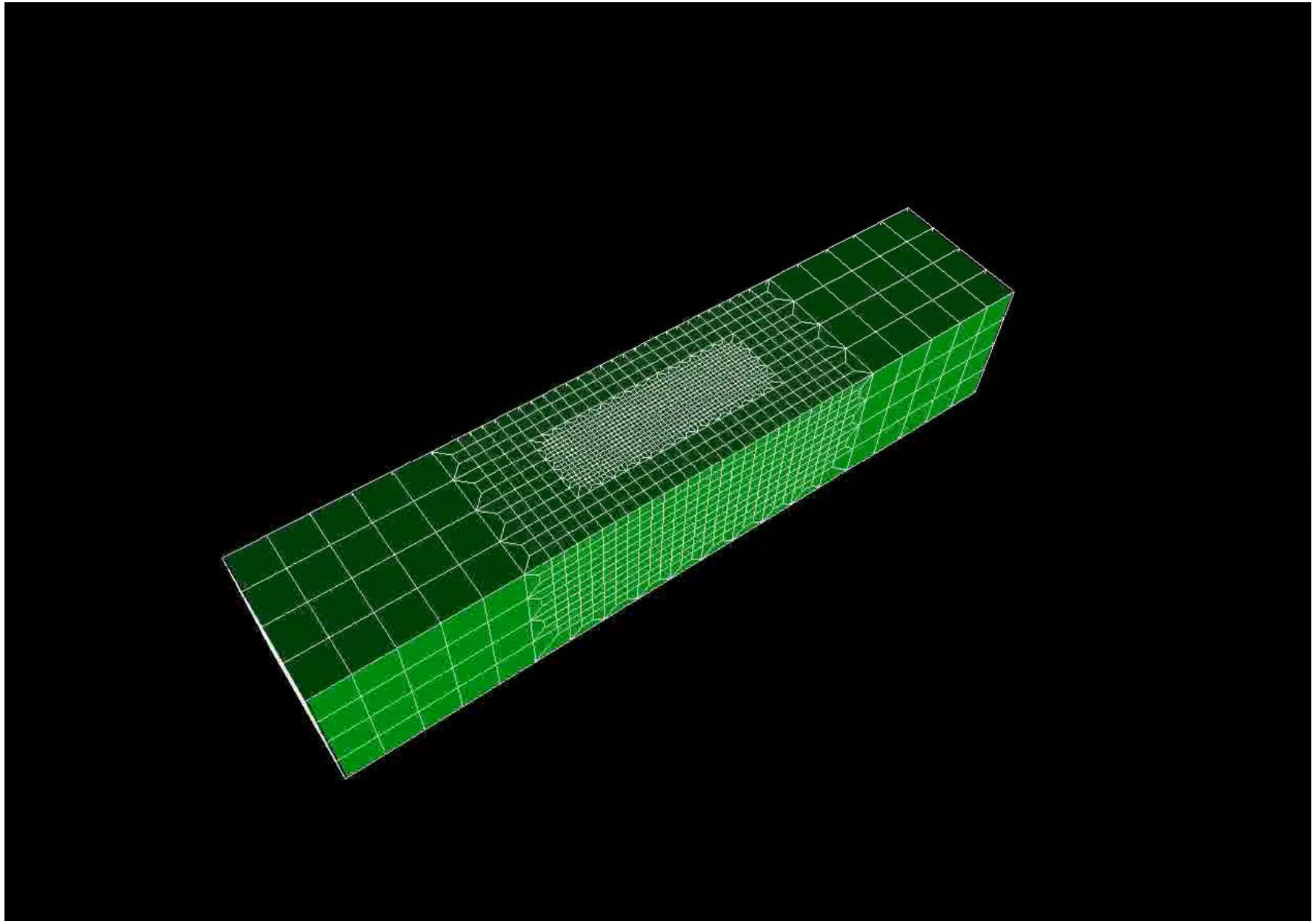
3D Simulations for Earthquake Generation Cycle San Andreas Faults, CA, USA

Stress Accumulation at Transcurrent Plate Boundaries

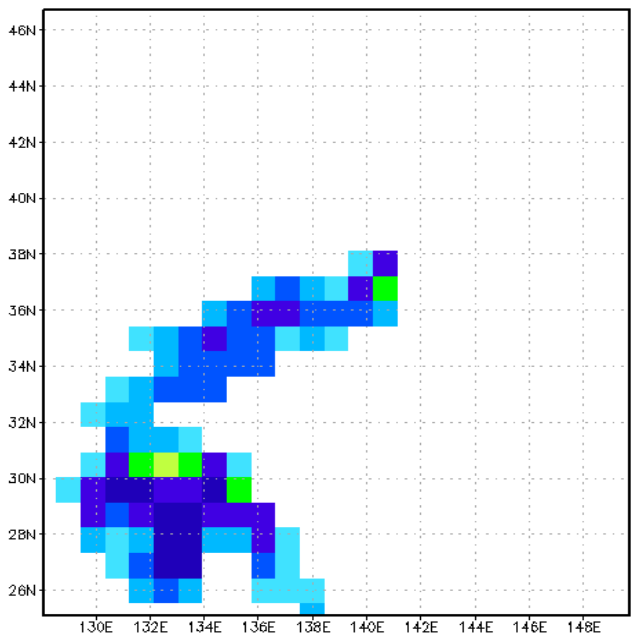


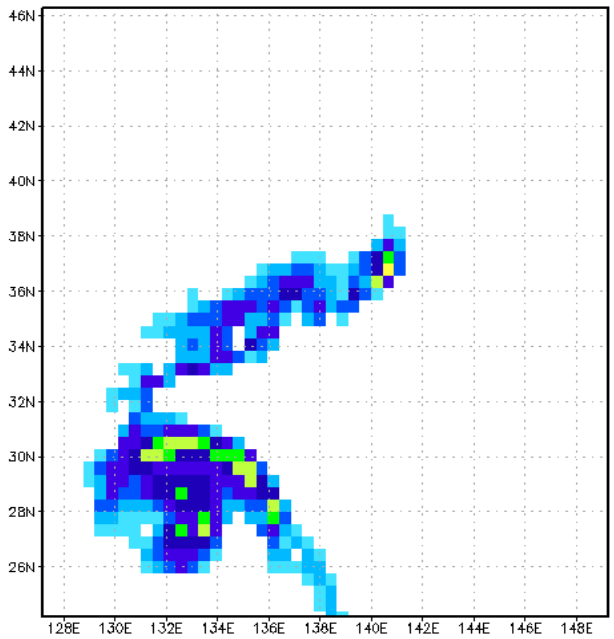


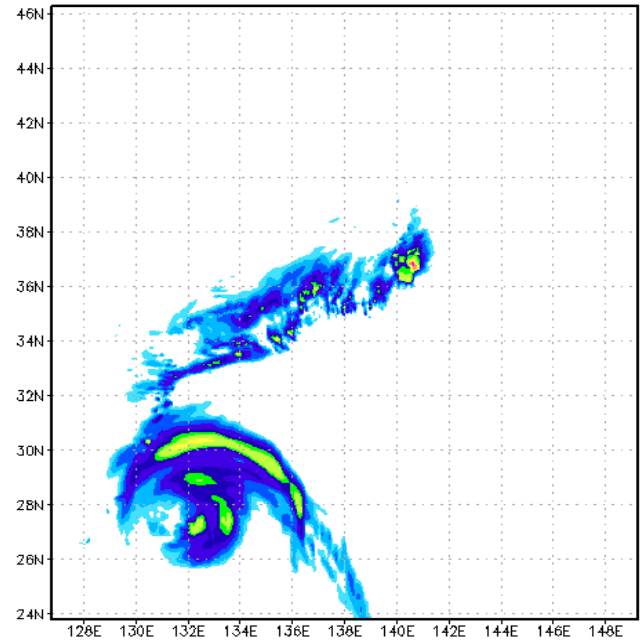
Adaptive FEM: High-resolution needed at meshes with large deformation (large accumulation)



Typhoon Simulations by FDM
Effect of Resolution



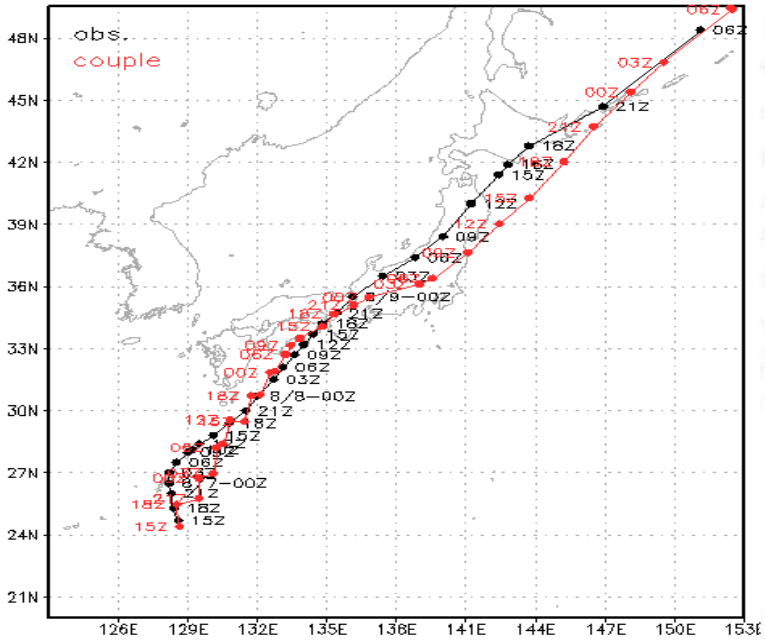


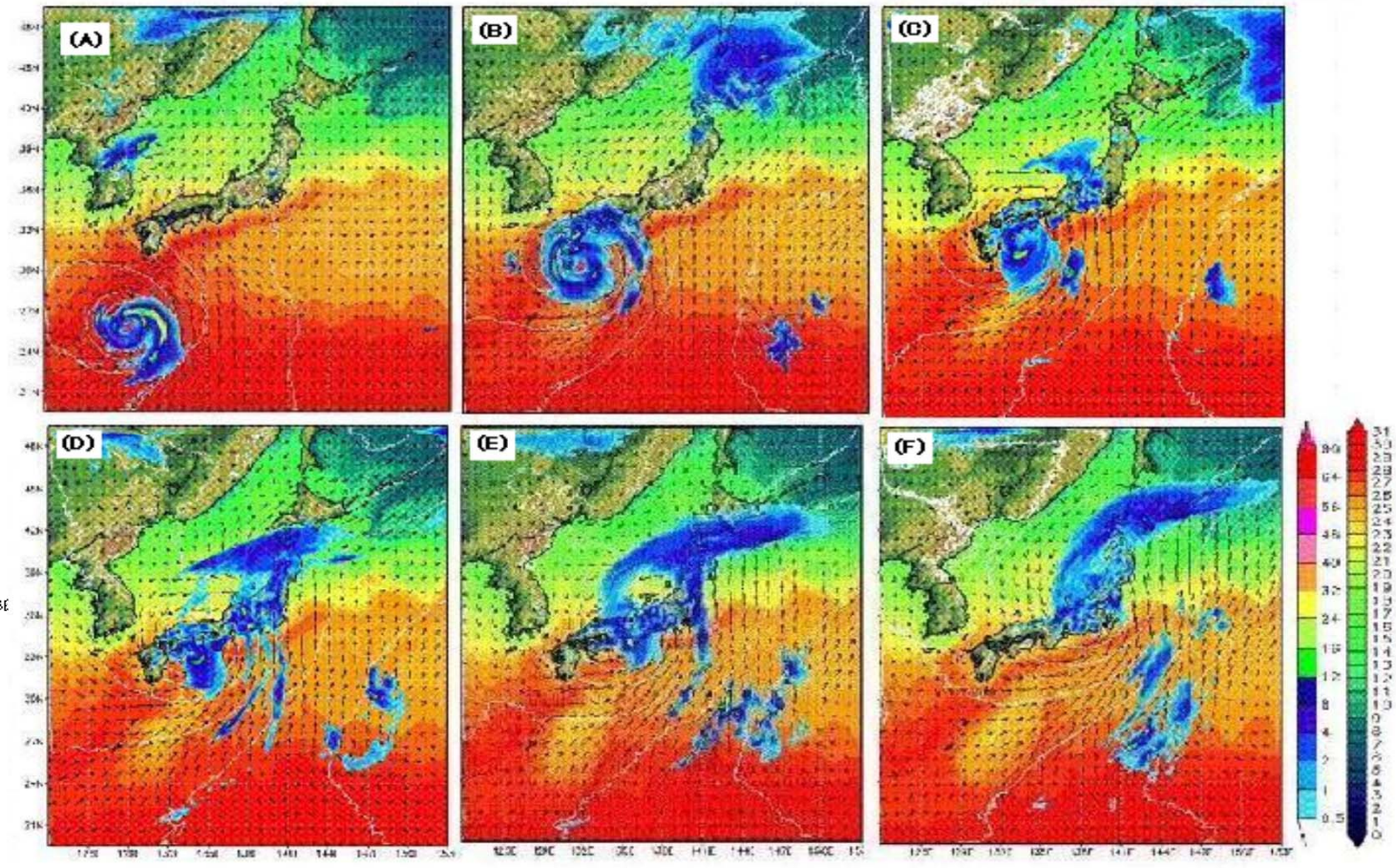


Δh=100km **Δh=50km** **Δh=5km**

[JAMSTEC]

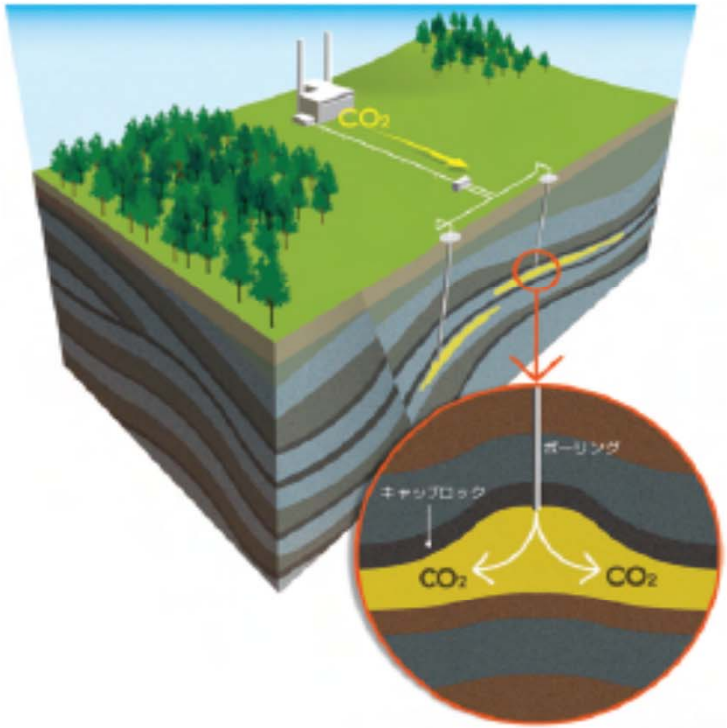
Simulation of Typhoon MANGKHUT in 2003 using the Earth Simulator





[JAMSTEC]

Simulation of Geologic CO_2 Storage



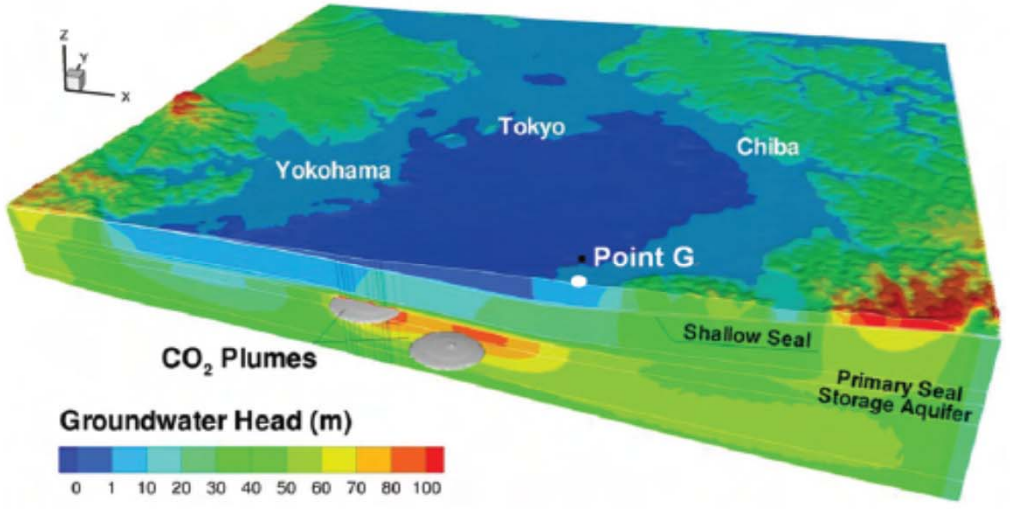
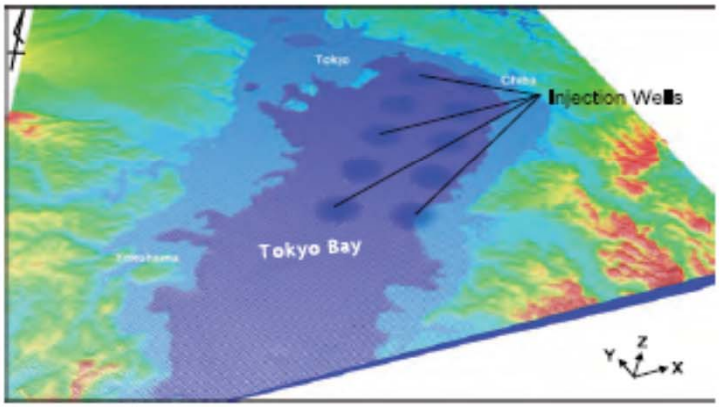
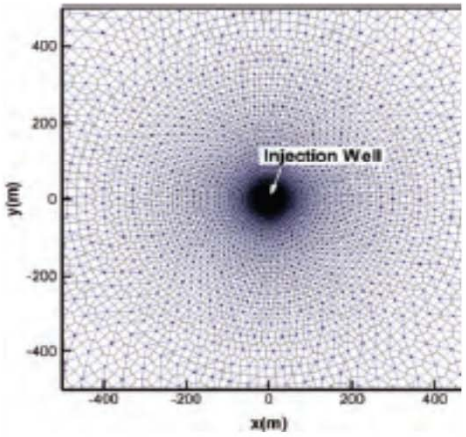
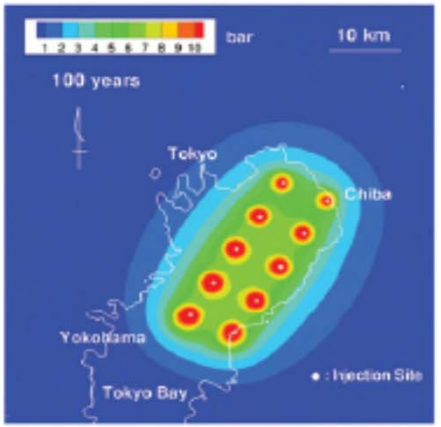
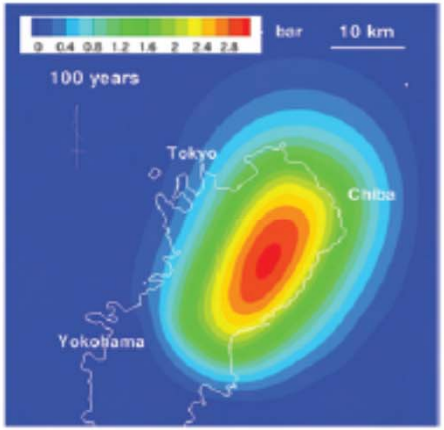


図 -4　CO_2 圧入後の地下水圧（全水頭換算）の分布（100 年後）









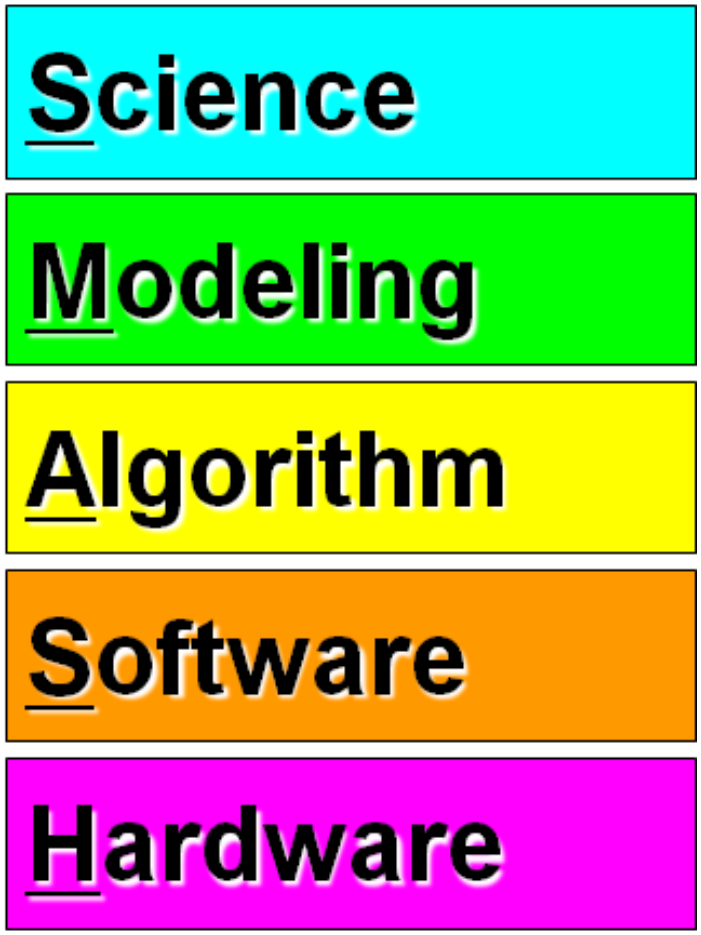
(a) 深部遮蔽層下面　　(b) 浅部遮蔽層下面

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

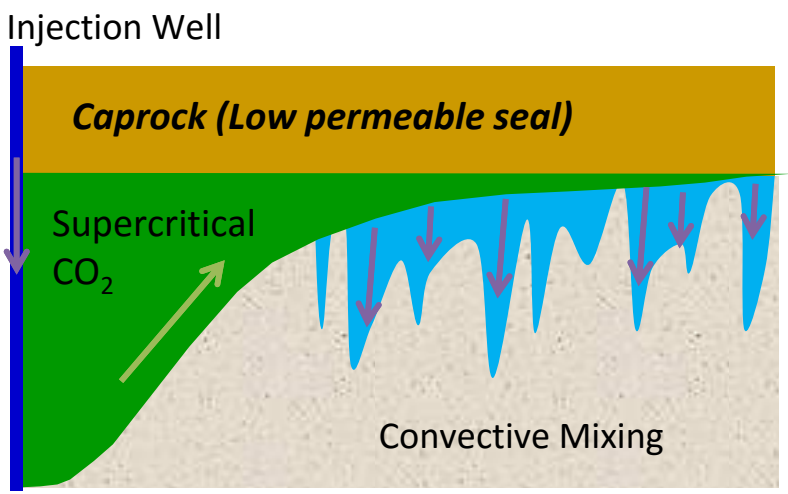
- International/Interdisciplinary Collaborations
  - Taisei (Science, Modeling)
  - Lawrence Berkeley National Laboratory, USA (Modeling)
  - Information Technology Center, the University of Tokyo (Algorithm, Software)
  - JAMSTEC (Earth Simulator Center) (Software, Hardware)
  - NEC (Software, Hardware)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

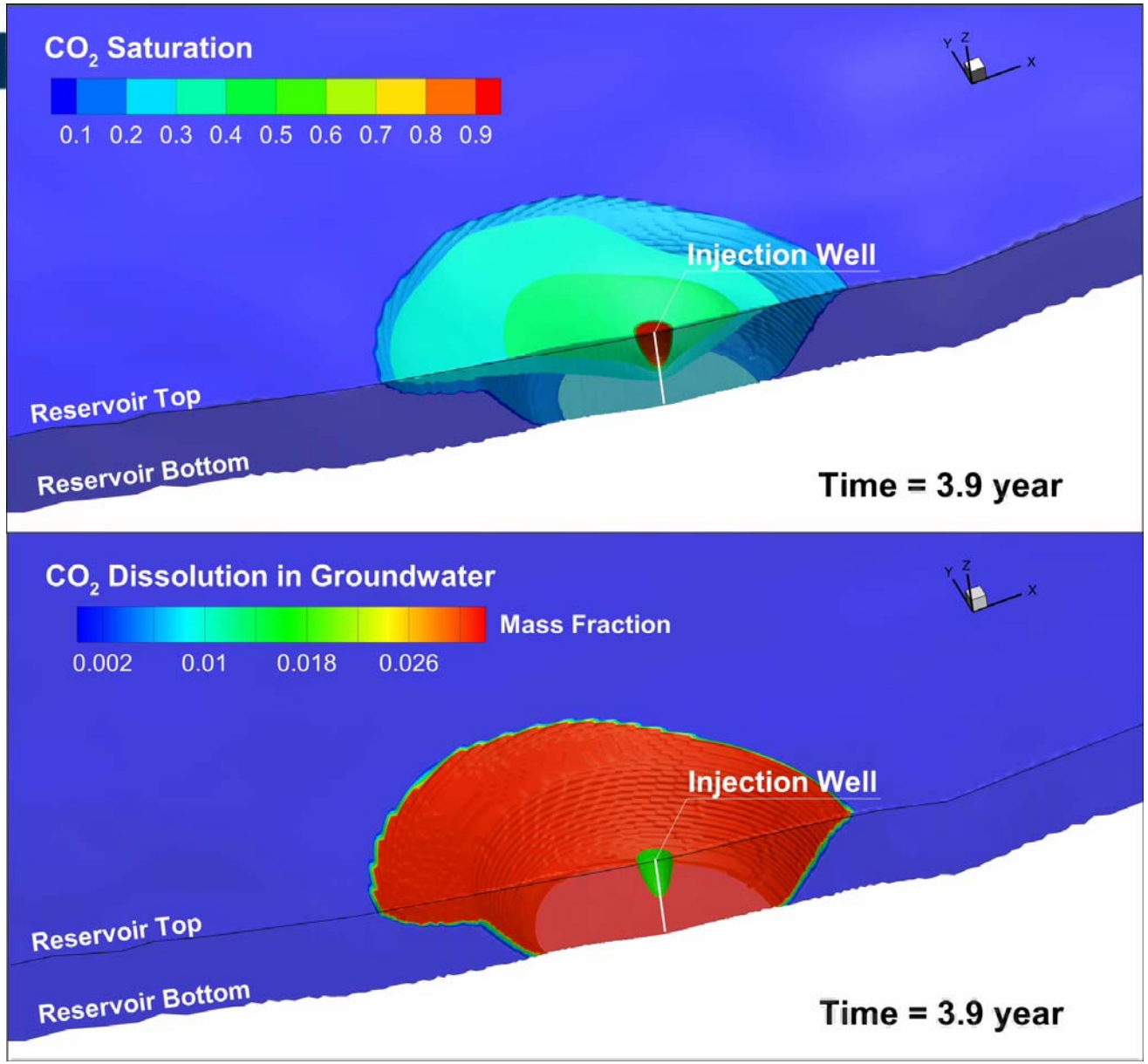
Diffusion-Dissolution-Convection Process



- **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

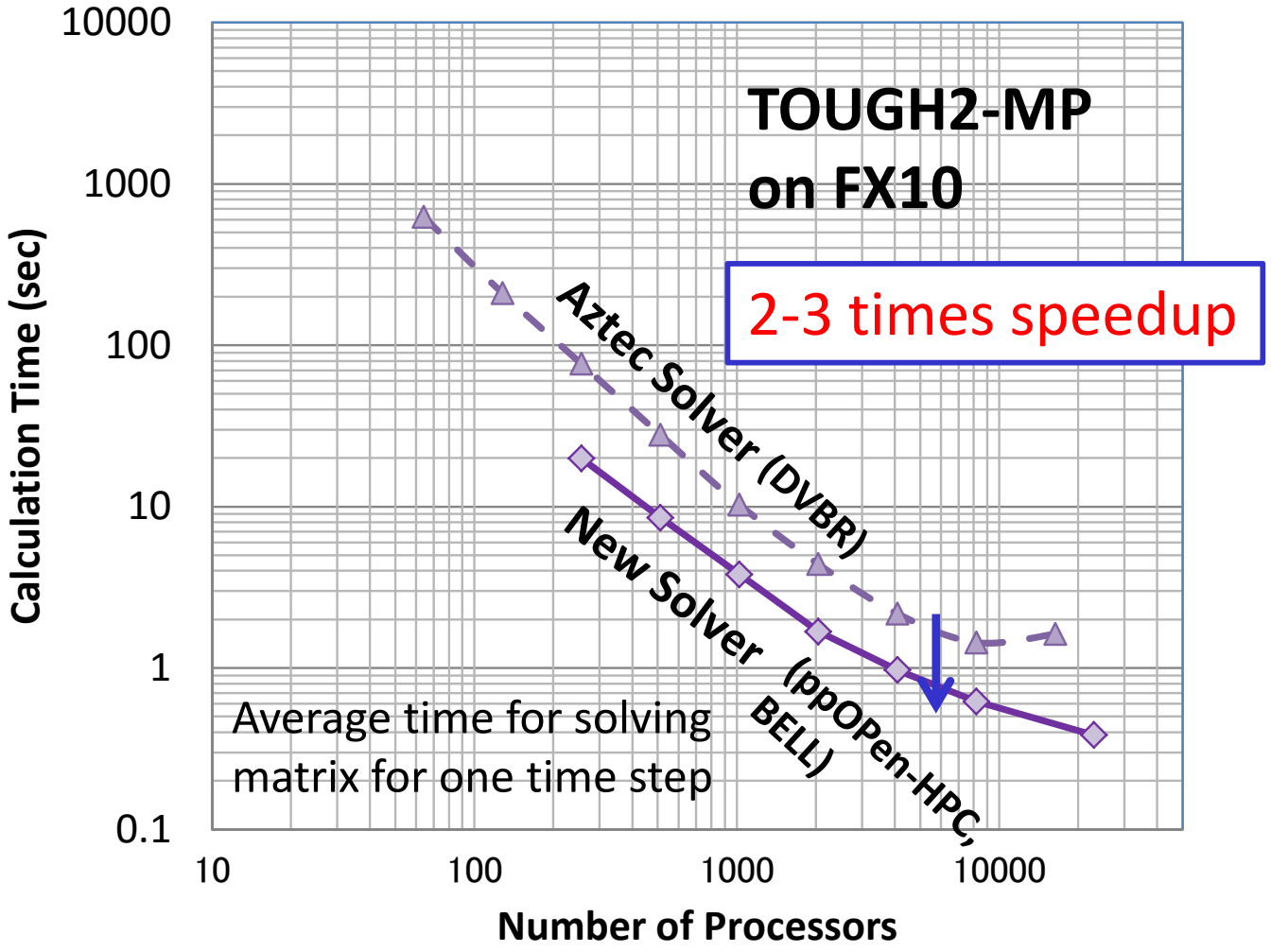
- Permeability: 100 md
- Porosity: 20%
- Pressure: 3MPa
- Temperature: 100°C
- Salinity: 15wt%

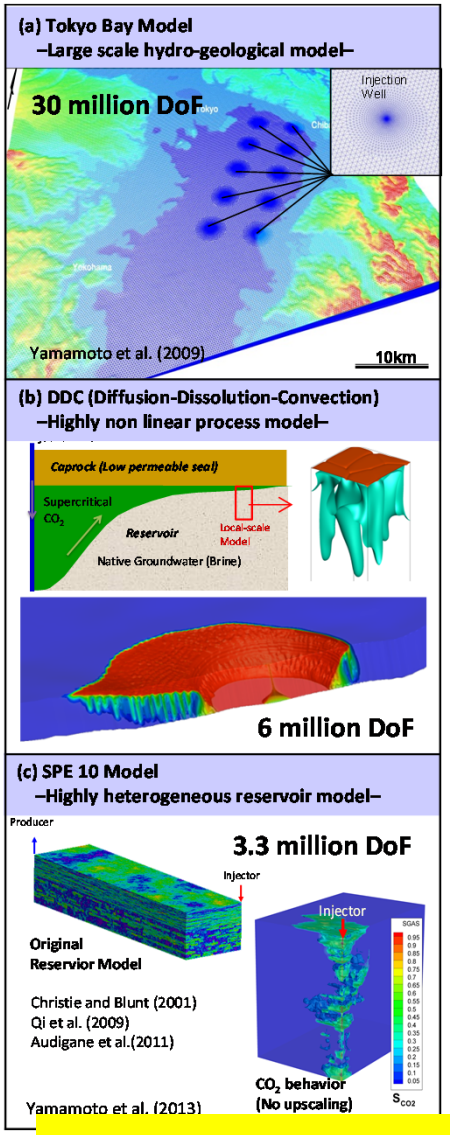
[Dr. Hajime Yamamoto, Taisei]

- 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
    - 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)

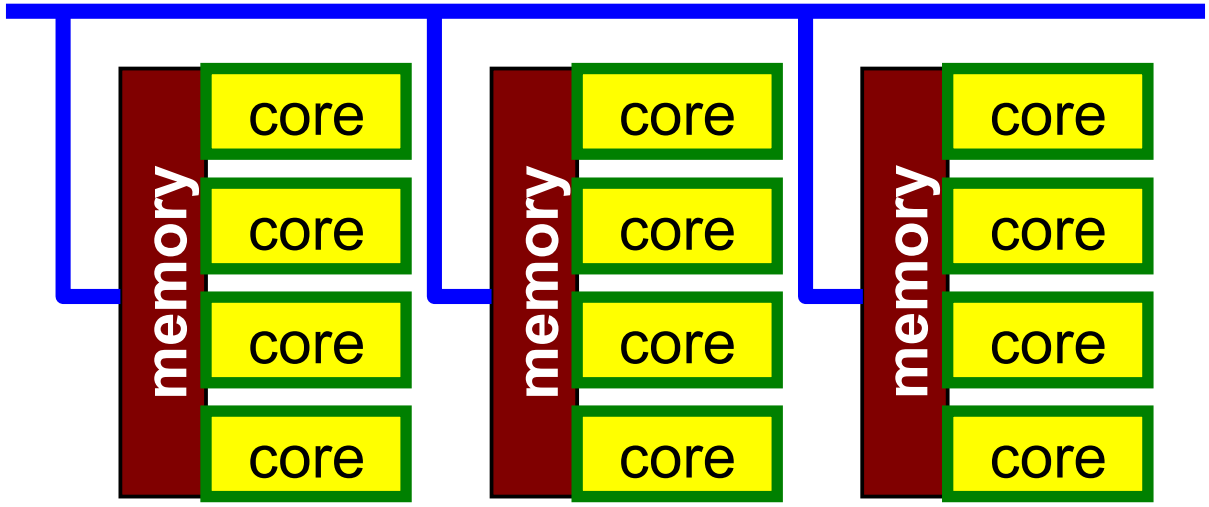
- Supercomputers and Computational Science
- **Overview of the Class**
- Future Issues

Background of This Class (1/4)

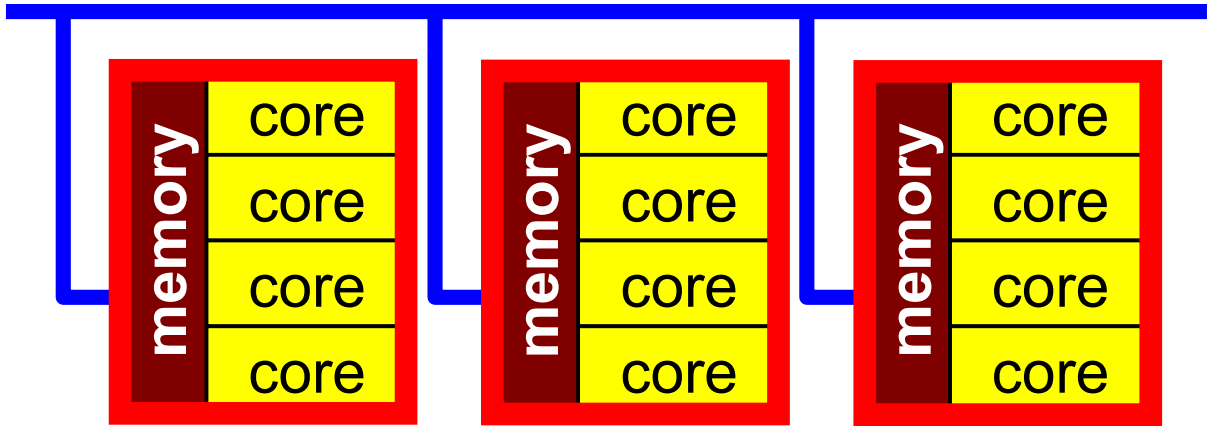
- In order to make full use of modern supercomputer systems with multicore/manycore architectures, hybrid parallel programming with message-passing and multithreading is essential.
- While MPI is widely used for message-passing, OpenMP for CPU and OpenACC for GPU are the most popular ways for multithreading on multicore/manycore clusters.
- In this 4-day course, we focus on optimization of single node performance using OpenMP and OpenACC for CPU and GPU.

Flat MPI vs. Hybrid

Flat-MPI：Each PE -> Independent



Hybrid：Hierarchal Structure



Background of This Class (2/4)

- We “parallelize” a finite-volume method (FVM) code with Krylov iterative solvers for Poisson’s equation on *Reedbush* supercomputer at the University of Tokyo with 1.93 PF peak, which consists of the most recent CPU’s (Intel Xeon E5-2695 v4 (Broadwell-EP)) and GPU’s (NDIVIA Tesla P100 (Pascal)).
- **It is assumed that all the participants have already finished”邁向Petascale高速計算 (February 2016)” or” 國家理論中心數學組「高效能計算」短期課程 (February 2017)”, or they have equivalent knowledge and experiences in numerical analysis, and parallel programming by MPI and OpenMP.**

Background of This Class (3/4)

- In the winter schools in February 2016 and February 2017, the target application was a 3D FVM code for Poisson's equation by Conjugate Gradient (CG) iterative method with very simple Point Jacobi preconditioner.
- This time, our target is same FVM code, but linear equations are solved by ICCG (CG iterative method with Incomplete Cholesky preconditioning), which is more complicated, powerful and widely-used in practical applications.
- Because ICCG includes "data dependency", where writing/reading data to/from memory could occur simultaneously, parallelization using OpenMP/OpenACC is not straight forward.

Background of This Class (4/4)

- We need certain kind of reordering in order to extract parallelism. In this 4-day course, lectures and exercise on the following issues will be provided:
  - Overview of Finite-Volume Method (FVM)
  - Kyrilov Iterative Method, Preconditioning
  - Implementation of the Program
  - Introduction to OpenMP/OpenACC
  - Reordering/Coloring Method
  - Parallel FVM by OpenMP/OpenACC
- You can learn multi-threaded programming by OpenMP on CPU and by OpenACC on GPU in just 4 Days using the world's most advanced system (Reedbush-U/H)

“Prerequisites”

- Completion of one of the following two short courses:
  - 邁向Petascale高速計算 (February 2016)
  - 國家理論中心數學組「高效能計算 」短期課程 (February 2017) https://sites.google.com/site/school4scicomp/2017-b-spring
  - (or equivalent knowledge and experiences in numerical analysis (FVM, preconditioned iterative solvers) and parallel programming using OpenMP and MPI)
- Experiences in Unix/Linux (vi or emacs)
- Fundamental numerical algorithms (Gaussian Elimination, LU Factorization, Jacobi/Gauss-Seidel/SOR Iterative Solvers, Conjugate Gradient Method (CG))
- Experiences in SSH Public Key Authentication Method

Preparation

- Windows
  - Cygwin with gcc/gfortran and OpenSSH
    - Please make sure to install gcc (C) or gfortran (Fortran) in “Devel”, and OpenSSH in “Net”
  - ParaView
- MacOS, UNIX/Linux
  - ParaView
- Cygwin: https://www.cygwin.com/
- ParaView: http://www.paraview.org

Date	Slot	Contents	Instructor
July 4 (T)	AM	Introduction, FVM (1/2)	Kengo Nakajima (born in 1962) 
	PM	FVM(2/2), Login-to-Reedbush, Introduction to OpenMP	
July 5 (W)	AM	Reordering (1/2)	
	PM	Reordering (2/2), Parallel FVM by OpenMP (1/2)	
July 6 (Th)	AM	Parallel FVM by OpenMP (2/2)	
	PM	Introduction to OpenACC	Tetsuya Hoshino (born in 1989) 
July 7 (F)	AM	Parallel FVM by OpenACC (1/2)	
	PM	Parallel FVM by OpenACC (2/2), Exercise	

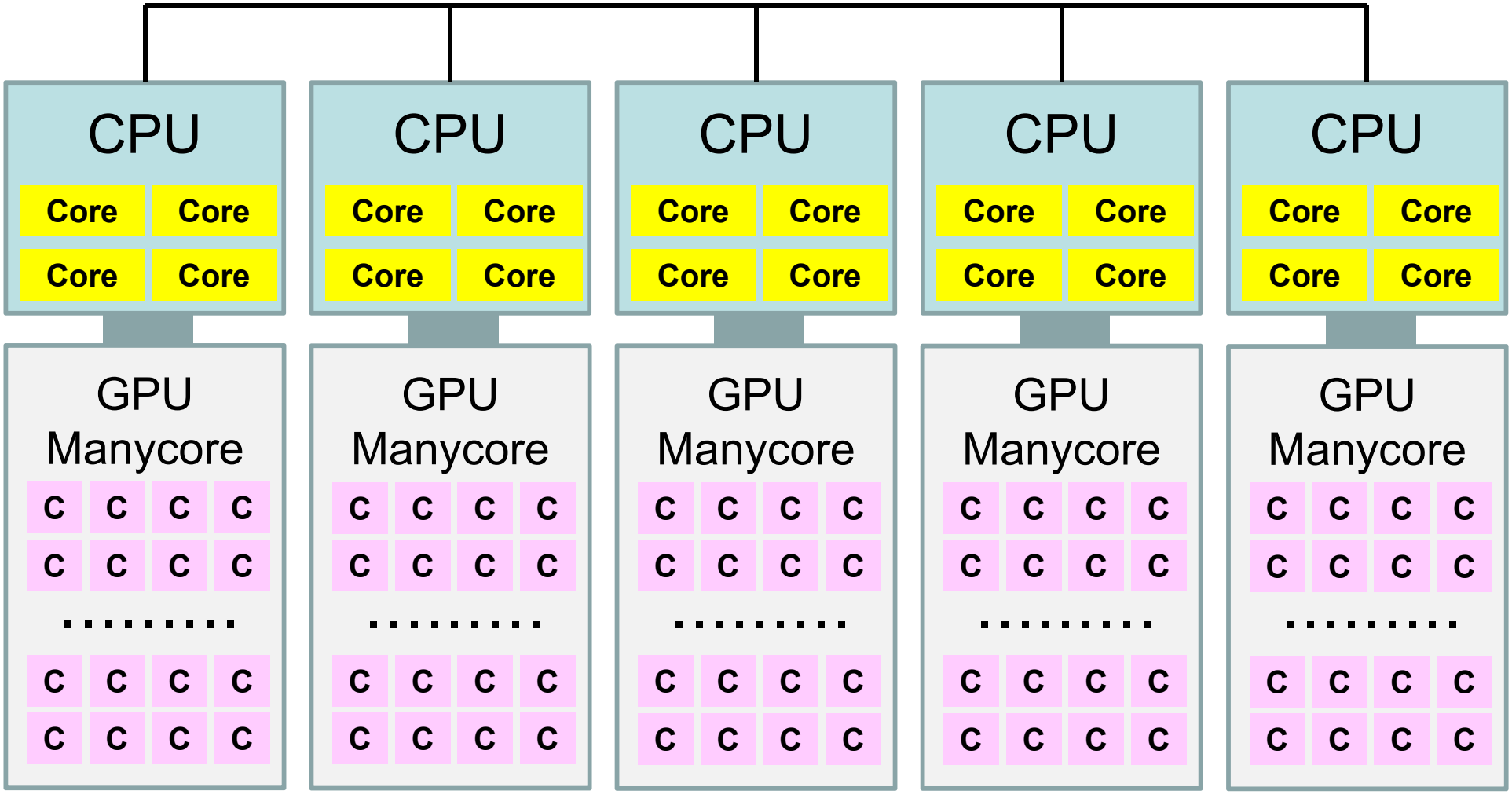
- Supercomputers and Computational Science
- Overview of the Class
- **Future Issues**

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

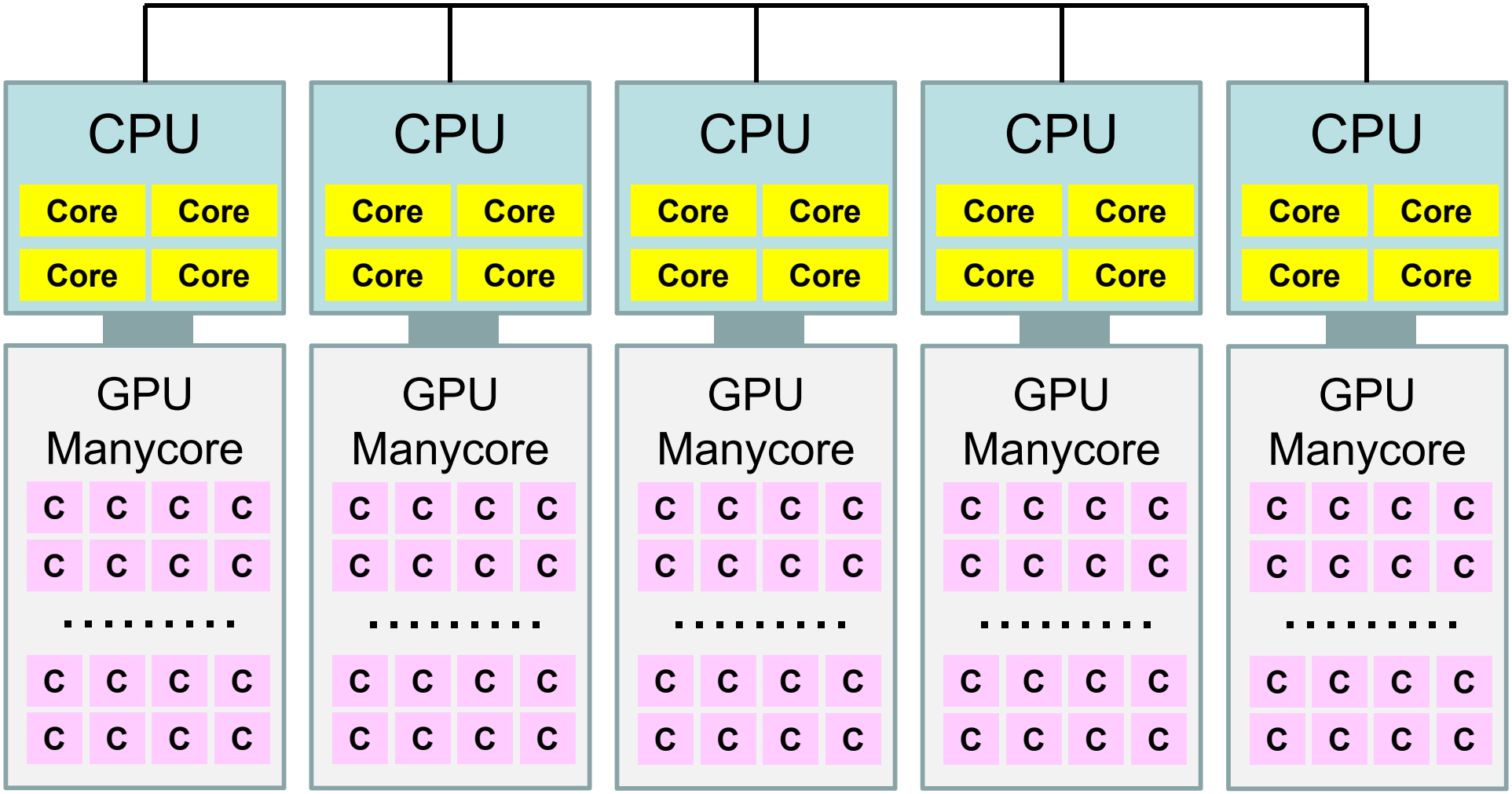
Supercomputers with Heterogeneous/Hybrid Nodes



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

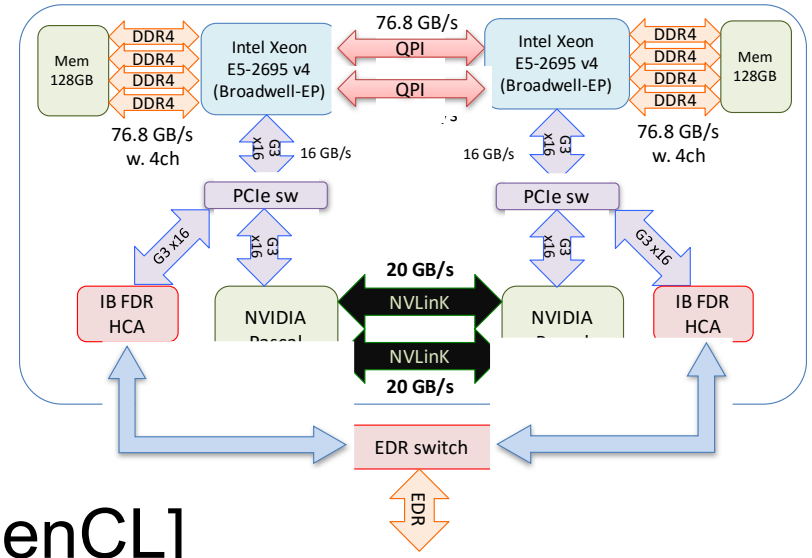
- Message Passing (e.g. MPI) + Multi Threading (e.g. OpenMP, CUDA, OpenCL, OpenACC etc.)
- In K computer and FX10, hybrid parallel programming is recommended
  - MPI + Automatic Parallelization by Fujitsu's Compiler
- Expectations for Hybrid
  - Number of MPI processes (and sub-domains) to be reduced
  - 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

This class is also useful for this type of parallel system



Parallel Programming Models

- Multicore Clusters (Reedbush-U)
  - MPI + OpenMP and (Fortan/C/C++)
- Multicore + GPU (Reedbush-H)
  - 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++)
- Multicore + Intel MIC/Xeon-Phi
- Intel MIC/Xeon-Phi Only
  - MPI + OpenMP and (Fortan/C/C++) is possible
    - + Vectorization



Future of Supercomputers (1/2)

- Technical Issues
  - Power Consumption
  - Reliability, Fault Tolerance, Fault Resilience
  - Scalability (Parallel Performancce)
- Petascale System
  - 2MW including A/C, 2M$/year, O($10^5$～$10^6$) cores
- Exascale System (10^3x Petascale)
  - After 2020 (?)
    - 2GW (2 B$/year !), O($10^8$～$10^9$) cores
  - Various types of innovations are on-going
    - to keep power consumption at 20MW (100x efficiency)
    - CPU, Memory, Network ...
  - Reliability

Future of Supercomputers (2/2)

- Not only hardware, but also numerical models and algorithms must be improved:
  - 省電力（Power-Aware/Reducing Algorithms）
  - 耐故障（Fault Resilient Algorithms）
  - 通信削減（Communication Avoiding/Reducing Algorithms）

- Co-Design by experts from various area (SMASH) is important
  - Exascale system will be a special-purpose system, not a general-purpose one.