

Introduction to Parallel Programming for Multicore/Manycore Clusters

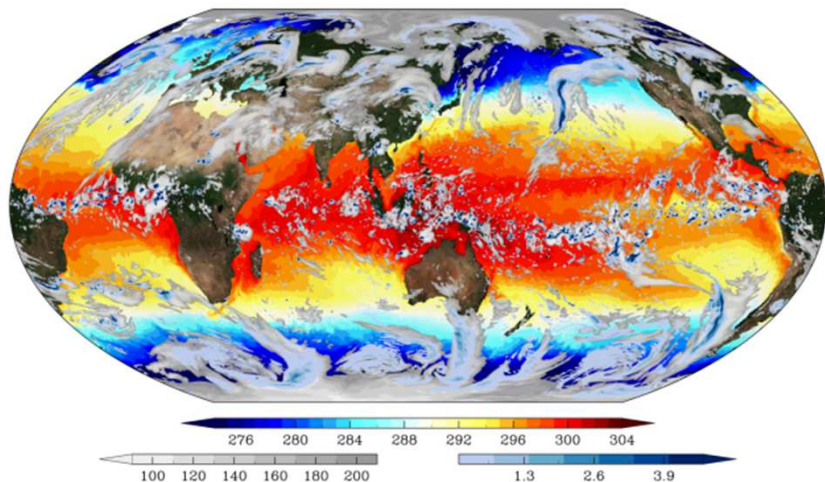
Introduction

Kengo Nakajima
Information Technology Center
The University of Tokyo

<http://nkl.cc.u-tokyo.ac.jp/NTU2023W/>

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.



Physics The Nobel Prize in Physics 2021 Syukuro Manabe - Facts


The Nobel Prize in Physics 2021

Syukuro Manabe
Klaus Hasselmann
Giorgio Parisi

Share this

[Facebook](#) [Twitter](#) [LinkedIn](#) [Email](#)

Syukuro Manabe Facts



Syukuro Manabe
The Nobel Prize in Physics 2021

Born: 21 September 1931, Shinga, Ehime Prefecture, Japan

Affiliation at the time of the award: Princeton University, Princeton, NJ, USA

Prize motivation: "for the physical modelling of Earth's climate, quantifying variability and reliably predicting global warming."

Prize share: 1/4

© Nobel Prize Outreach

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/Co-Design needed
 - They 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
 - **I hope you can extend your knowledge/experiences a little bit from your original area through this class for your future career**
 - It is more difficult than communicating with foreign scientists from same area.
- (Q): Computer Science, Computational Science, or Numerical Algorithms ?

This Class ...

Science

Modeling

Algorithm

Software

Hardware

- Target: Parallel FVM (Finite-Volume Method) using OpenMP
- Science: 3D Poisson's 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)

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

Unix, Fortan, C etc.

Big gap here !!

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

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

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

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

1. Unix, Fortan, C etc.

Kengo Nakajima 中島研吾 (1/2)

- Current Position

- Professor, Supercomputing Research Division, Information Technology Center, The University of Tokyo (情報基盤センター)
 - Department of Mathematical Informatics, Graduate School of Information Science & Engineering, The University of Tokyo (情報理工・数理情報学)
 - Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo (工・電気系工学)
- Deputy Director, RIKEN R-CCS (Center for Computational Science) (Kobe) (20%) (2018.Apr.-)

- Research Interest

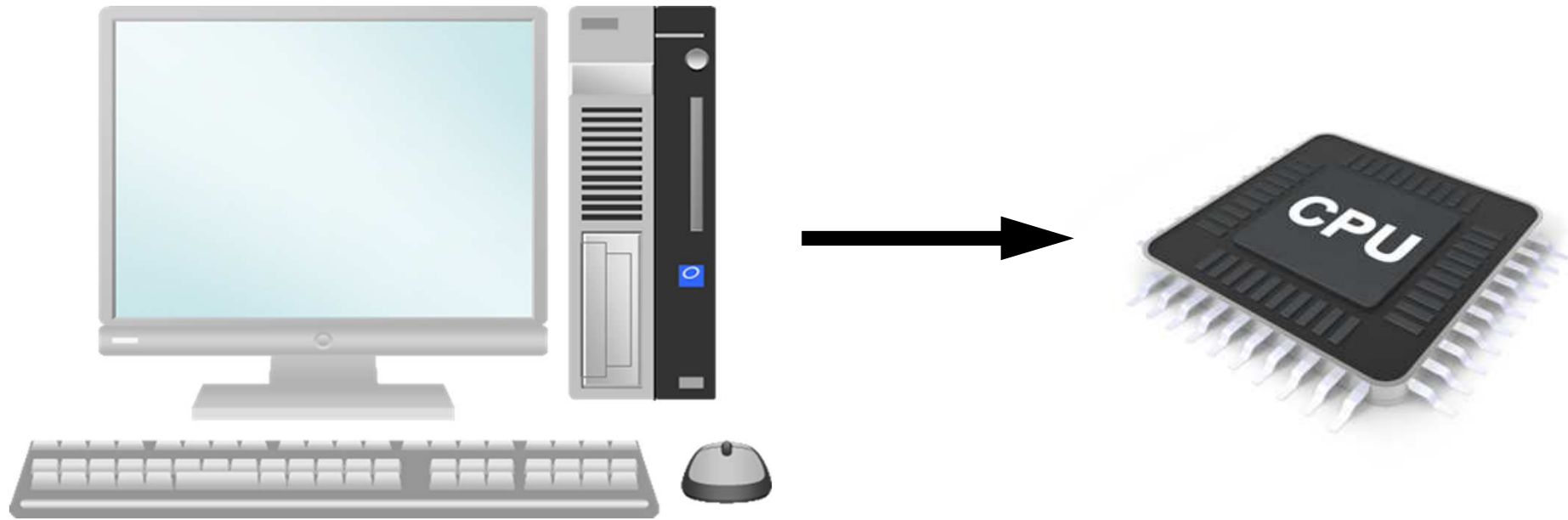
- High-Performance Computing
- Parallel Numerical Linear Algebra (Preconditioning)
- Parallel Programming Model
- Computational Mechanics, Computational Fluid Dynamics
- Adaptive Mesh Refinement, Parallel Visualization

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
 - 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 (2009-2018), part-time

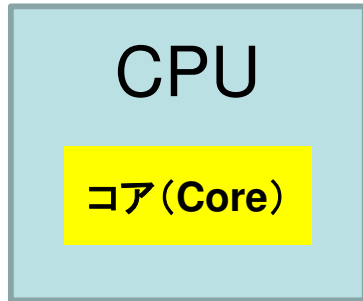
- **Supercomputers and Computational Science**
- Overview of the Class
- Future Issues

Computer & CPU

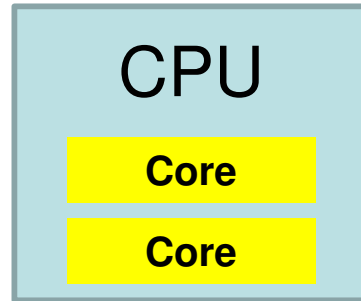


- Central Processing Unit (中央处理装置): CPU
- CPU's used in PC and Supercomputers are based on same architecture
- GHz: Clock Rate
 - Frequency: Number of operations by CPU per second
 - GHz -> 10^9 operations/sec
 - Simultaneous 4-8 (or more) instructions per clock

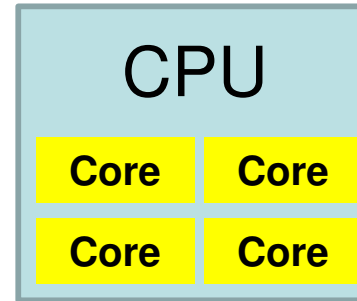
Multicore CPU



Single Core
1 cores/CPU



Dual Core
2 cores/CPU

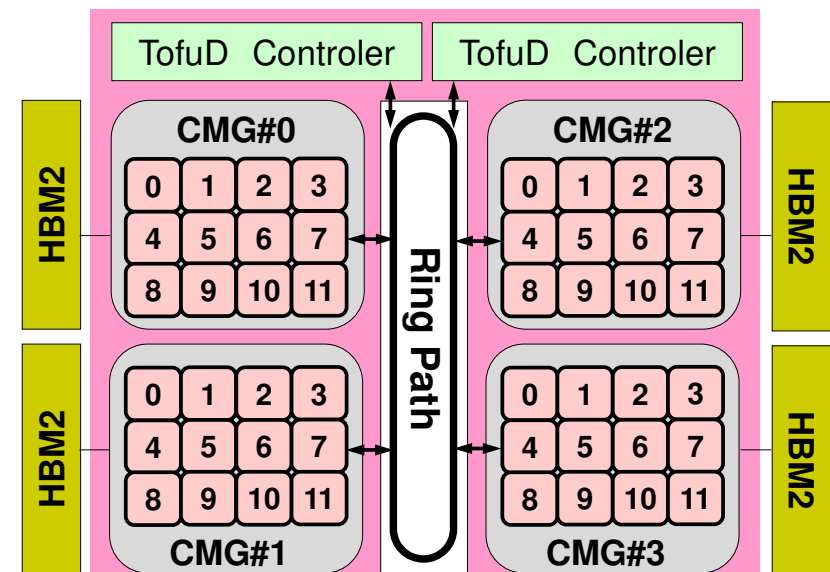


Quad Core
4 cores/CPU

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

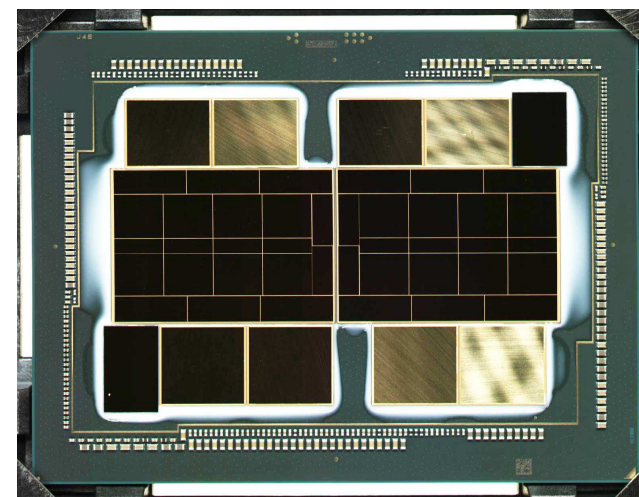
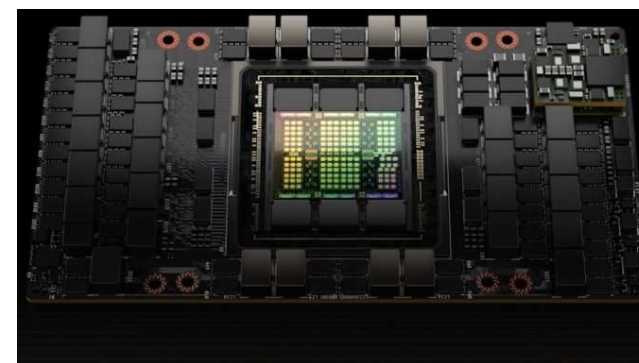
- GPU: Manycore
 - $O(10^1)$ - $O(10^2)$ cores
- More and more cores
 - Parallel computing

- **Odyssey: 48-cores/node**
 - **Fujitsu/ARM A64FX**



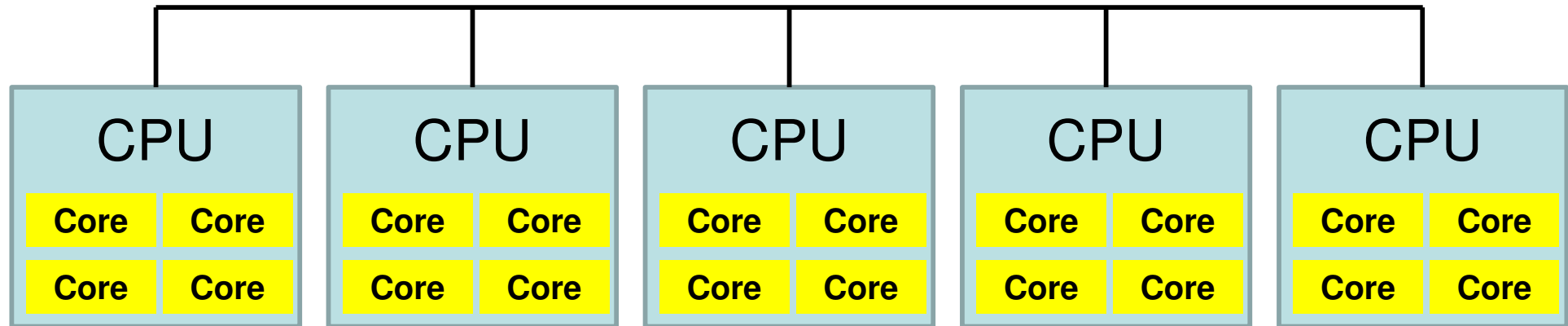
GPU/Accelerators

- GPU: Graphic Processing Unit
 - GPGPU: General Purpose GPU
 - $O(10^2)$ cores
 - High Memory Bandwidth
 - (was) cheap
 - NO stand-alone operations
 - Host CPU needed
 - Programming: CUDA, OpenACC etc.
- NVIDIA, AMD, Intel ...

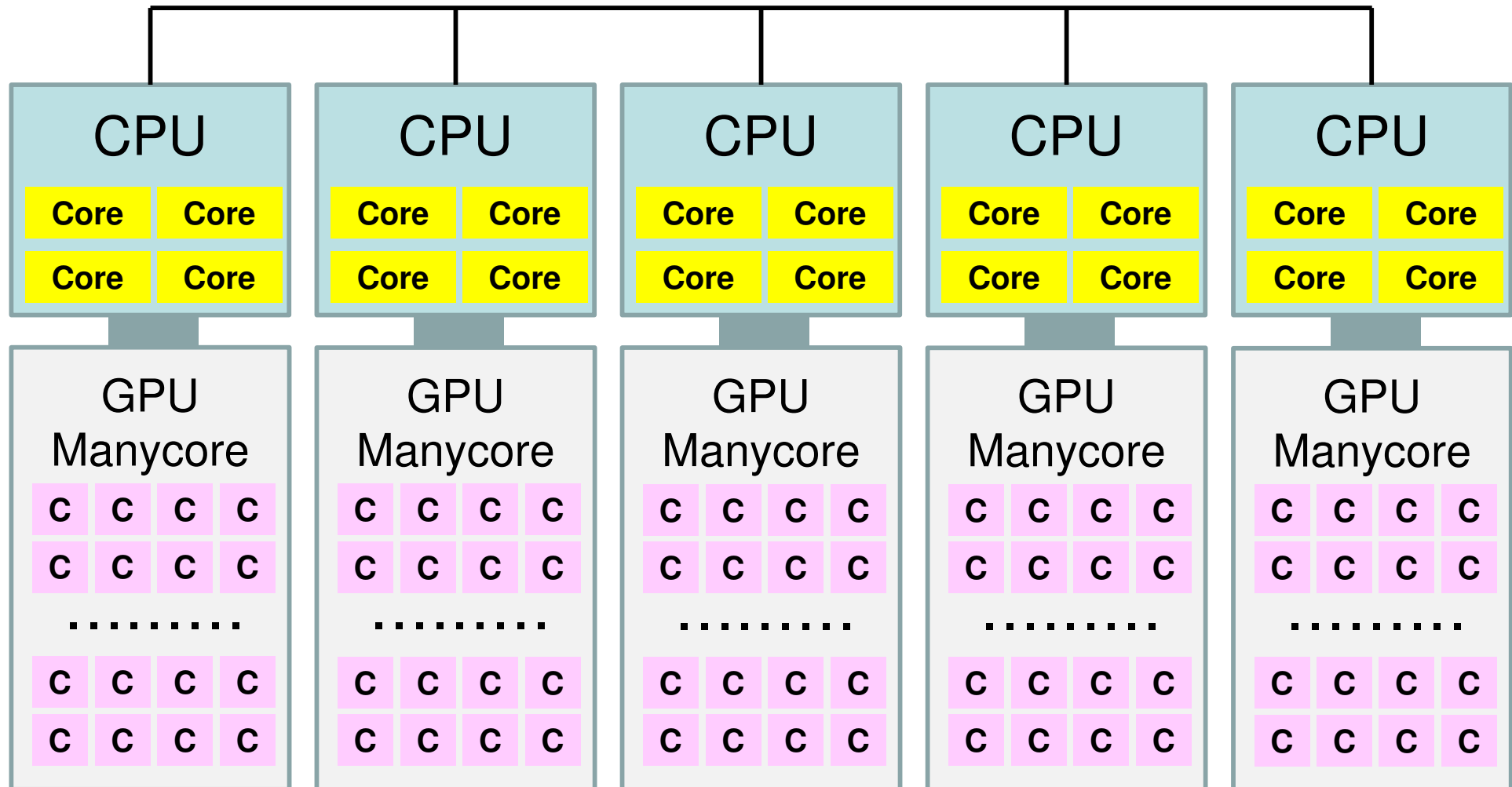


Parallel Supercomputers

Multicore CPU's are connected through network



Supercomputers with Heterogeneous/Hybrid Nodes



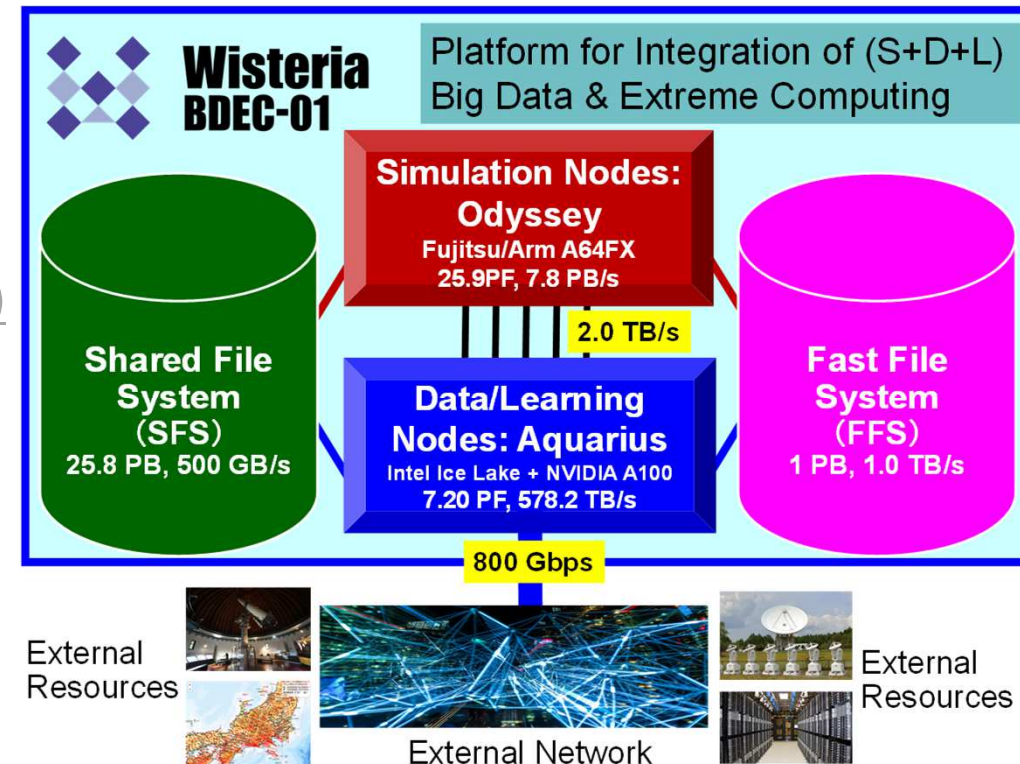
Performance of Supercomputers

- Performance of CPU: Clock Rate
- FLOPS (Floating Point Operations per Second)
 - Real Number
- Recent Multicore CPU
 - 32 (or more) FLOPS per Clock
 - (e.g.) Peak performance of a core with 2.0 GHz
 - $2 \times 10^9 \times 32 = 64 \times 10^9$ FLOPS = 64 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

Wisteria/BDEC-01

The 1st BDEC System (Big Data & Extreme Computing)
Platform for Integration of
(Simulation+Data+Learning) (S+D+L)

- **Operation started on May 14, 2021**
- **33.1 PF, 8.38 PB/sec by Fujitsu**
 - **~4.5 MVA with Cooling, ~360m²**
- 2 Types of Node Groups
 - Hierarchical, Hybrid, Heterogeneous (h3)
 - Simulation Nodes: Odyssey
 - Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF
 - 7,680 nodes (368,640 cores), Tofu-D
 - General Purpose CPU + HBM
 - Commercial Version of “Fugaku”
 - Data/Learning Nodes: Aquarius
 - Data Analytics & AI/Machine Learning
 - Intel Xeon Ice Lake + NVIDIA A100, 7.2PF
 - 45 nodes (90x Ice Lake, 360x A100), IB-HDR
 - Some of the DL nodes are connected to external resources directly
- File Systems: SFS (Shared/Large) + FFS (Fast/Small)



Wisteria/BDEC-01

The 1st BDEC System (Big Data & Extreme Computing)
Platform for Integration of
(Simulation+Data+Learning) (S+D+L)

- Operation started on May 14, 2021
- 33.1 PF, 8.38 PB/sec by **Fujitsu**
 - ~4.5 MVA with Cooling, ~360m²

• 2 Types of Node Groups

– Hierarchical, Hybrid, Heterogeneous (h3)

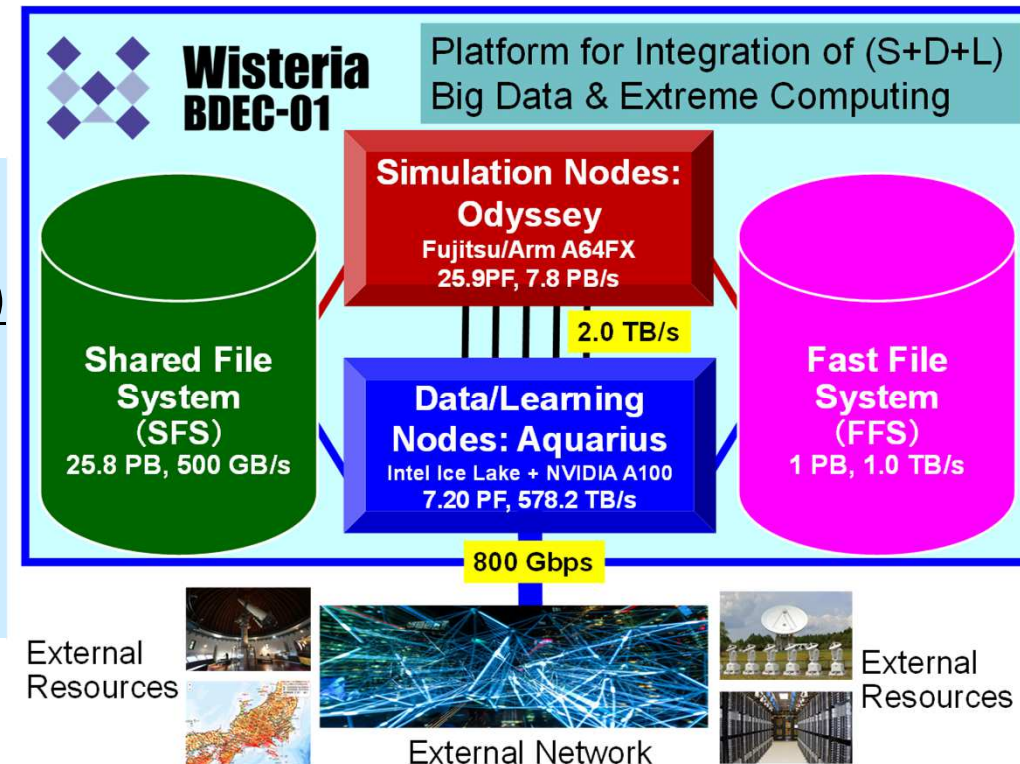
– Simulation Nodes: Odyssey

- Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF
 - 7,680 nodes (368,640 cores), Tofu-D
 - General Purpose CPU + HBM
 - Commercial Version of “Fugaku”

– Data/Learning Nodes: Aquarius

- Data Analytics & AI/Machine Learning
- Intel Xeon Ice Lake + NVIDIA A100, 7.2PF
 - 45 nodes (90x Ice Lake, 360x A100), IB-HDR
- Some of the DL nodes are connected to external resources directly

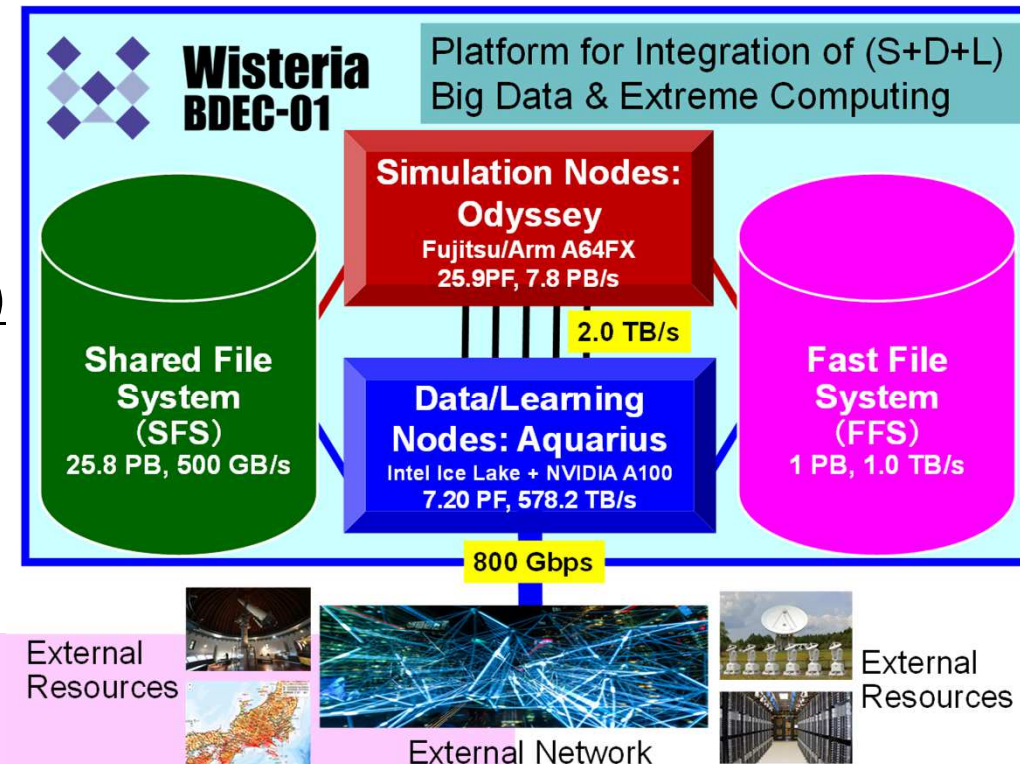
• File Systems: SFS (Shared/Large) + FFS (Fast/Small)



Wisteria/BDEC-01

The 1st BDEC System (Big Data & Extreme Computing)
Platform for Integration of
(Simulation+Data+Learning) (S+D+L)

- Operation started on May 14, 2021
- 33.1 PF, 8.38 PB/sec by **Fujitsu**
 - ~4.5 MVA with Cooling, ~360m²
- **2 Types of Node Groups**
 - Hierarchical, Hybrid, Heterogeneous (h3)
 - **Simulation Nodes: Odyssey**
 - **Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF**
 - 7,680 nodes (368,640 cores), Tofu-D
 - General Purpose CPU + HBM
 - Commercial Version of “Fugaku”
 - **Data/Learning Nodes: Aquarius**
 - **Data Analytics & AI/Machine Learning**
 - **Intel Xeon Ice Lake + NVIDIA A100, 7.2PF**
 - 45 nodes (90x Ice Lake, 360x A100), IB-HDR
 - **Some of the DL nodes are connected to external resources directly**
- File Systems: SFS (Shared/Large) + FFS (Fast/Small)

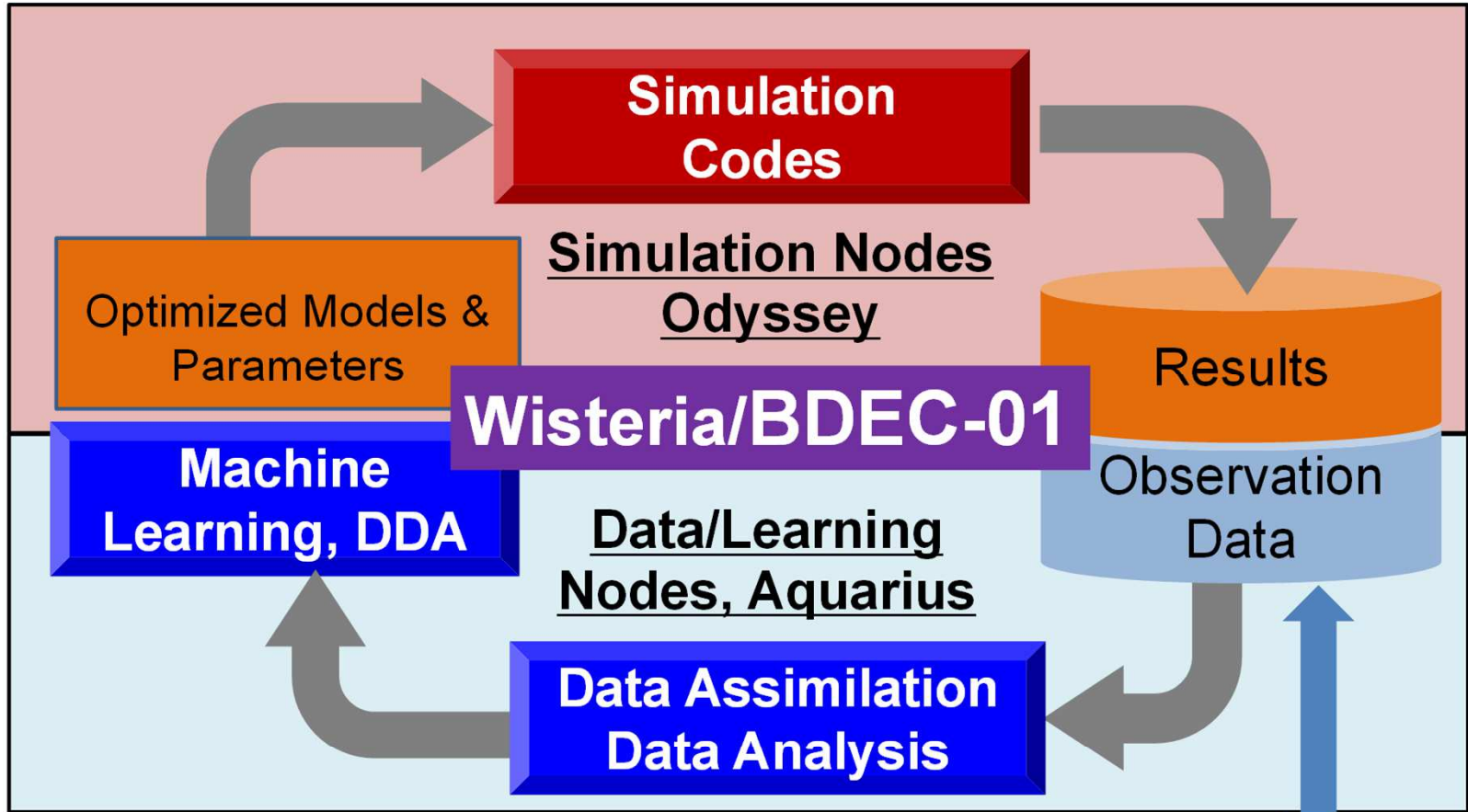


**Simulation Nodes
Odyssey**
25.9 PF, 7.8 PB/s

**Fast File System
(FFS)**
1.0 PB,
1.0 TB/s

**Shared File System
(SFS)**
25.8 PB,
0.50 TB/s

**Data/Learning Nodes
Aquarius**
7.20 PF, 578.2 TB/s



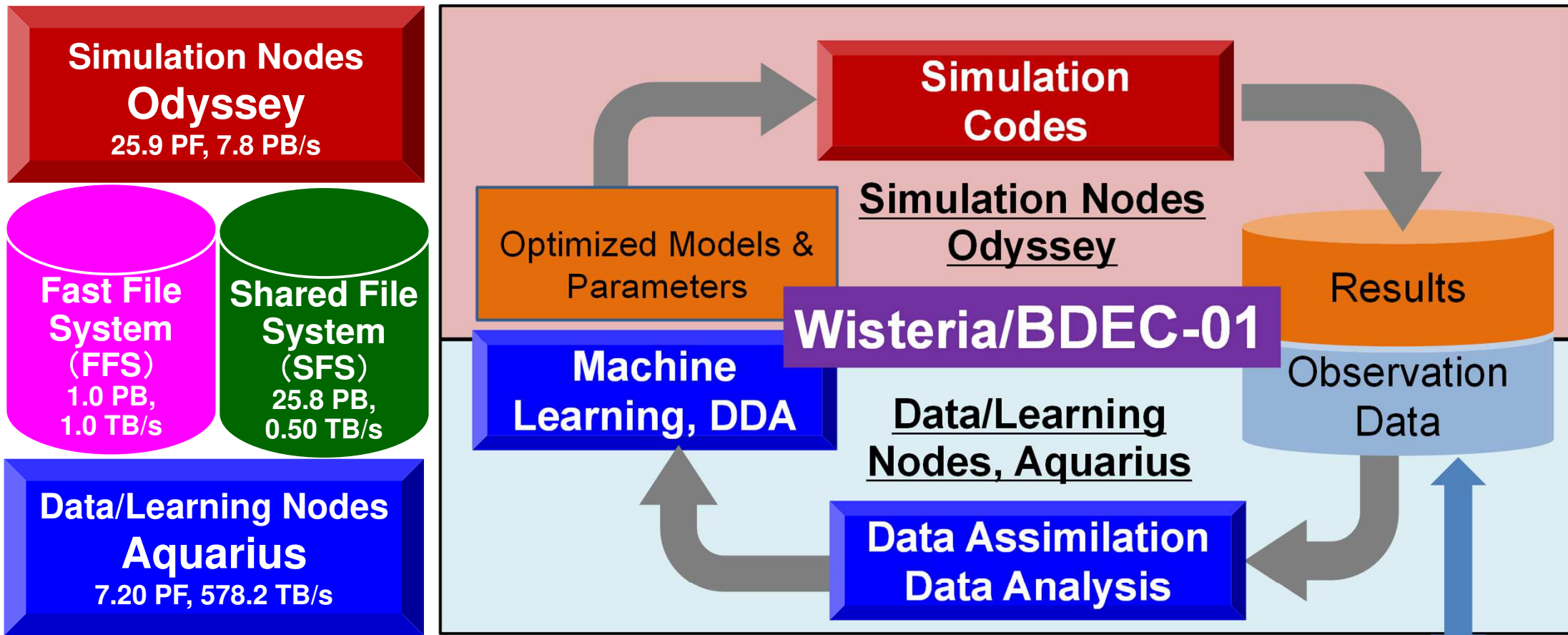
Server,
Storage,
DB,
Sensors,
etc.



External Network

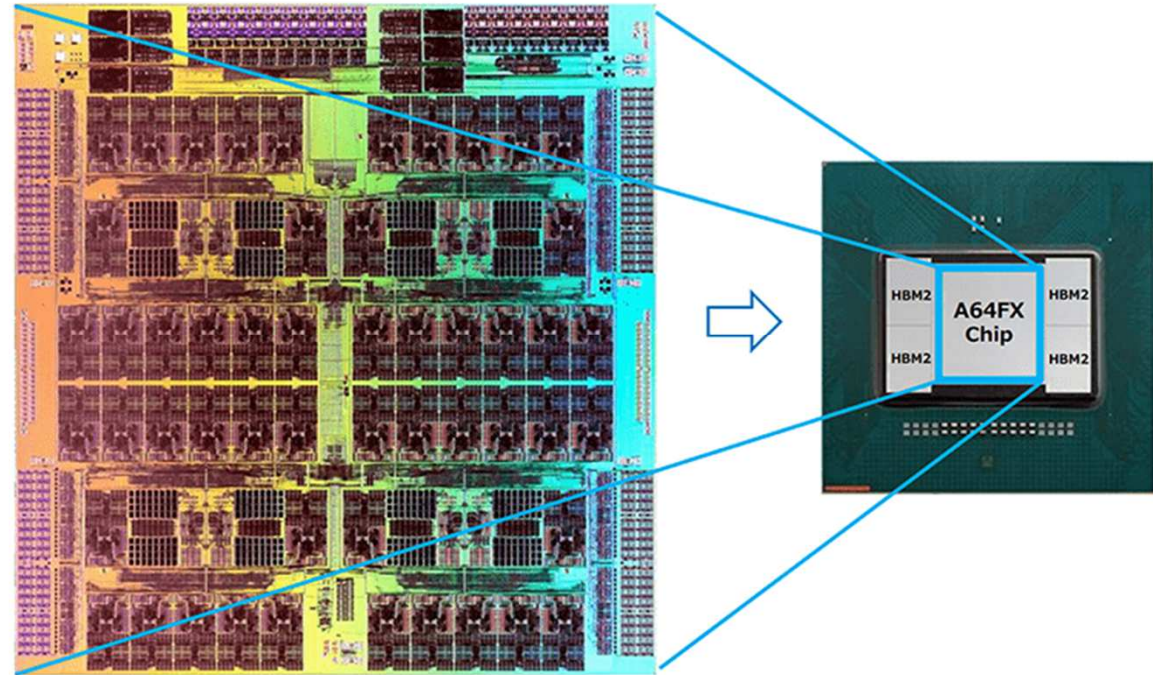
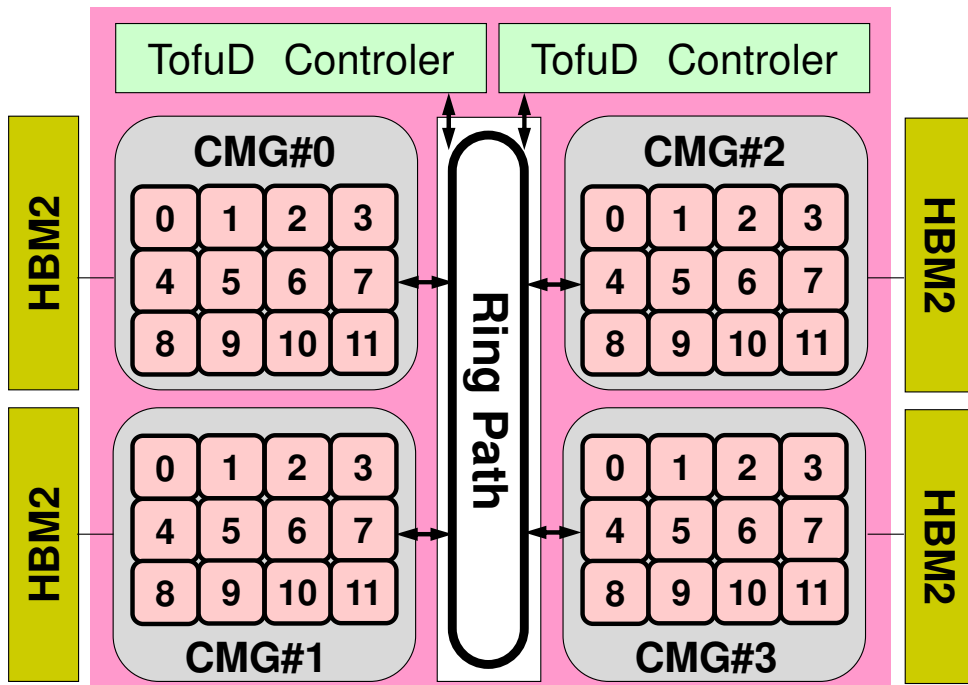


External Resources



Optimization of Models/Parameters for Simulations by Data Analytics & Machine Learning (S+D+L)

A64FX Processor on Odyssey



- 4 CMG's (Core Memory Group), 12 cores/CMG
- 2.2 GHz
 - 32 DP (Double Precision) FLOP operations per Clock
- Peak Performance
 - Single Core: $2.2 \times 32 = 70.4$ GFLOPS
 - Single CMG: $70.4 \times 12 = 844.8$ GFLOPS
 - Single Node (CPU): $844.8 \times 4 = 3,379.2$ GFLOPS = 3.3792 TFLOPS

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
- **Wisteria/BDEC-01 (Odyssey) is 17th in the TOP 500 (November 2021)**

10^{19} = 10 ExaFlops

10^{18} = 1 ExaFlops

10^{17} = 100 PetaFlops

10^{16} = 10 PetaFlops

10^{15} = 1 PetaFlops

10^{14} = 100 TeraFlops

10^{13} = 10 TeraFlops

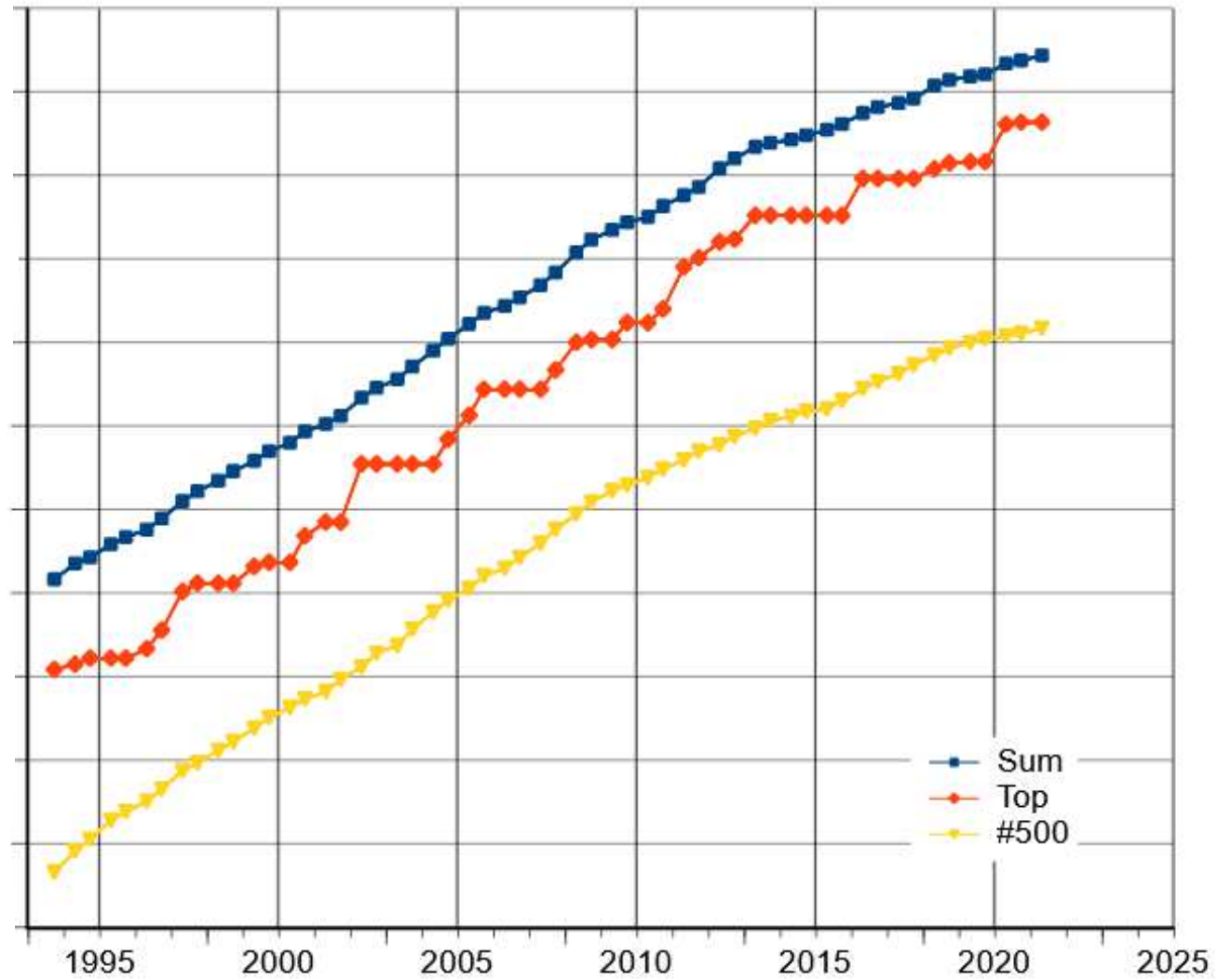
10^{12} = 1 TeraFlops

10^{11} = 100 GigaFlops

10^{10} = 10 GigaFlops

10^9 = 1 GigaFlops

10^8 = 100 MegaFlops



60th TOP500 List (Nov., 2022)

Wisteria/BDEC-01 (Odyssey) is 23rd

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

	Site	Computer/Year Vendor	Cores	R_{max} (PFLOPS)	R_{peak} (PFLOPS)	Power (kW)
1	Frontier, 2022, USA DOE/SC/Oak Ridge National Laboratory	HPE Cray, EX235a, AMD Optimized 3 rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	8,730,112	1,102.00 (=1.102 EF)	1,685.65	21,100
2	Fugaku, 2020, Japan R-CCS, RIKEN	Fujitsu, PRIMEHPC FX1000, Fujitsu A64FX 48C 2.2GHz, Tofu-D	7,630,848	442,010 (= 442.0 PF)	537,212.0	29,899
3	LUMI, 2022, Finland EuroHPC/CSC	HPE Cray, EX235a, AMD Optimized 3 rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	1,110,144	151.90	214.35	2,942
4	Leonardo, 2022, Italy EuroHPC/CINECA	Atos, BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Infiniband HDR	2,414,592	148.60	200.79	10,096
5	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	148.60	200.79	10,096
6	Sierra, 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	94.64	125.71	7,438
7	Sunway TaihuLight, 2016, China National Supercomputing Center in Wuxi	Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	10,649,600	93.01	125.44	15,371
8	Perlmutter, 2021, USA DOE/NERSC/LBNL	HPE Cray, EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10	761,856	70.87	93.75	2,528
9	Selene, 2020, USA NVIDIA	NVIDIA DGX A100 SuperPOD, AMD EPYC 7742 64C 2.25GHz, NVIDIA GA100, Mellanox Infiniband HDR	555,520	63.46	79.22	2,646
10	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	61.44	100.68	18,482
22	ABC1 2.0, 2021, Japan AIST	Fujitsu, PRIMERGY GX2570 M6, Xeon Platinum 8360Y 36C 2.4GHz, NVIDIA A100 SXM4 40 GB, InfiniBand HDR	504,000	22.21	54.34	1,600
23	Wisteria/BDEC-01 (Odyssey), 2021, Japan ITC, University of Tokyo	Fujitsu, PRIMEHPC FX1000, A64FX 48C 2.2GHz, Tofu interconnect D	368,640	22.12	25.95	1,468

Linpack on My iPhone XS



Cray-1S

Normal Mode

Low-Power Mode

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10

Mflop/s: 18800.05

Time: 0.0364

Norm Res: 5.1700

Precision: 2.22044605e-16

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10

Mflop/s: 8359.34

Time: 0.0726

Norm Res: 5.1700

Precision: 2.22044605e-16

Multithread (parallel)

Max Mflop/s: 20627.07
Avg Mflop/s: 17294.78

Max Mflop/s: 11868.06
Avg Mflop/s: 9329.87

20:29

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10

Mflop/s: 5511.57

Time: 0.0152

Norm Res: 5.1700

Precision: 2.22044605e-16

20:28

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10

Mflop/s: 3737.27

Time: 0.0224

Norm Res: 5.1700

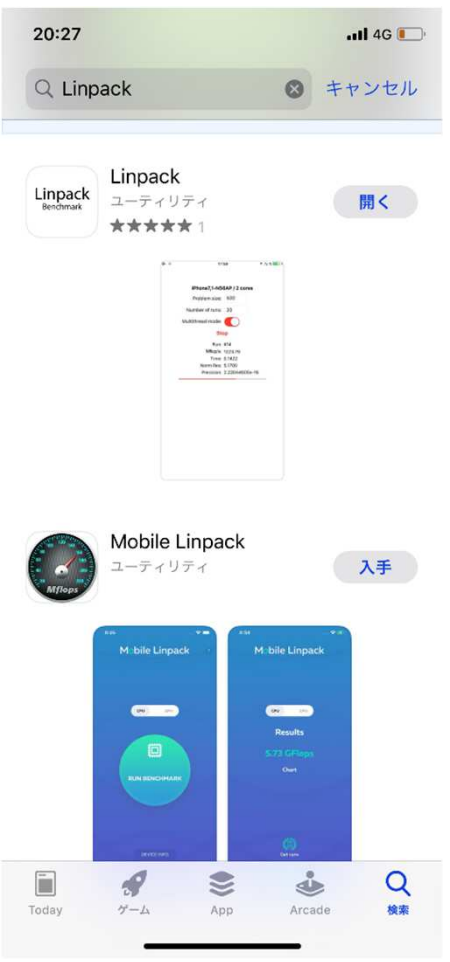
Precision: 2.22044605e-16

Single-thread (serial)

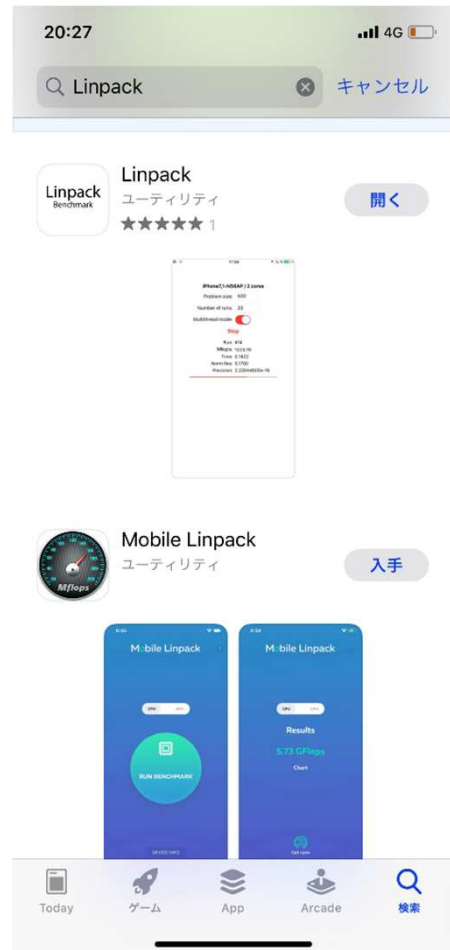
Max Mflop/s: 5521.89
Avg Mflop/s: 4921.39

Max Mflop/s: 3769.22
Avg Mflop/s: 3586.08

- Performance of my iPhone XS is about 20,000 Mflops
 - Odyssey: 3.38Tflops
- Cray-1S
 - Supercomputer of my company in 1985 with 80 Mflops
 - I do not know the price, but we had to pay 10 USD for 1 sec. computing !



Linpack on My iPhone XS



Normal Mode

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10
Mflop/s: 18800.05
Time: 0.0364
Norm Res: 5.1700
Precision: 2.22044605e-16

Max Mflop/s: 20627.07
Avg Mflop/s: 17294.78

Low-Power Mode

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10
Mflop/s: 8359.34
Time: 0.0726
Norm Res: 5.1700
Precision: 2.22044605e-16

Max Mflop/s: 11868.06
Avg Mflop/s: 9329.87

20:29

ライブドアニュース **インストール**

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10
Mflop/s: 5511.57
Time: 0.0152
Norm Res: 5.1700
Precision: 2.22044605e-16

Max Mflop/s: 5521.89
Avg Mflop/s: 4921.39

20:28

App Store

ライブドアニュース **インストール**

iPhone11,2-D321AP / 6 cores

Problem size: 500

Number of runs: 10

Multithread mode:

Run benchmark

Run: #10
Mflop/s: 3737.27
Time: 0.0224
Norm Res: 5.1700
Precision: 2.22044605e-16

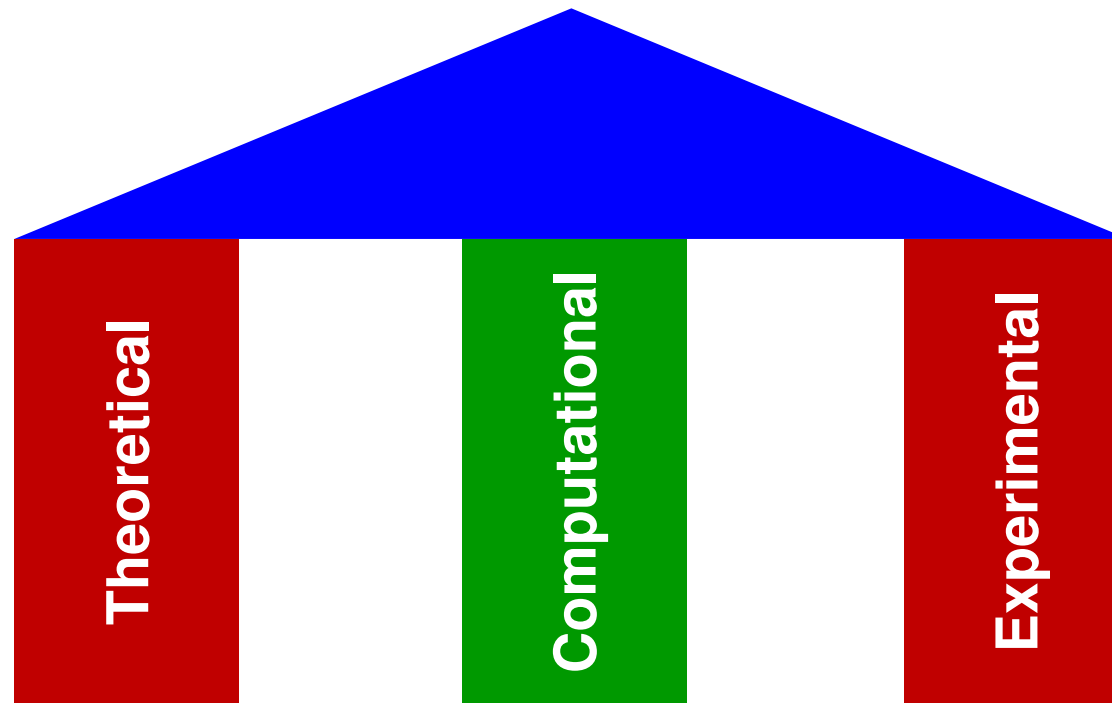
Max Mflop/s: 3769.22
Avg Mflop/s: 3586.08

- You can change Problem size, and # of runs.
 - “Size=500” means linear equations $Ax=b$ with 500 unknowns are solved
- Actually, problem size affects performance of computing so much !!

Computational Science

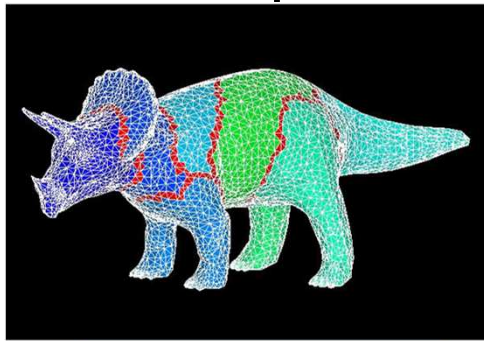
The 3rd Pillar of Science

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

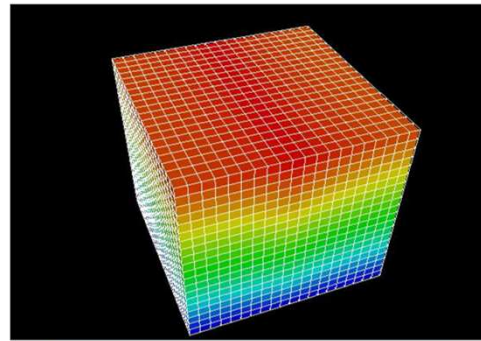


Methods for Scientific Computing

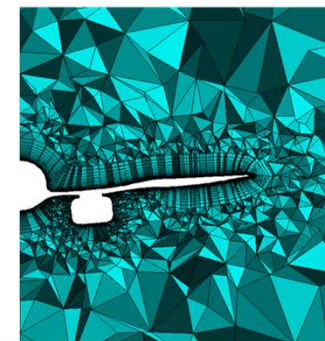
- Numerical solutions of PDE (Partial Diff. Equations)
- Grids, Meshes, Particles
 - Large-Scale Linear Equations
 - Finer meshes provide more accurate solutions



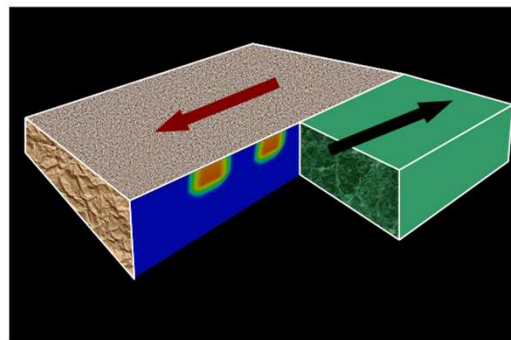
有限要素法
Finite Element Method
FEM



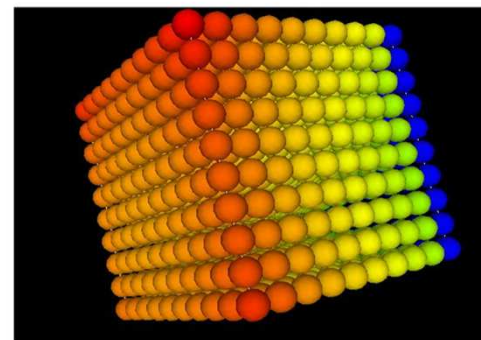
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



境界要素法
Boundary Element Method
BEM

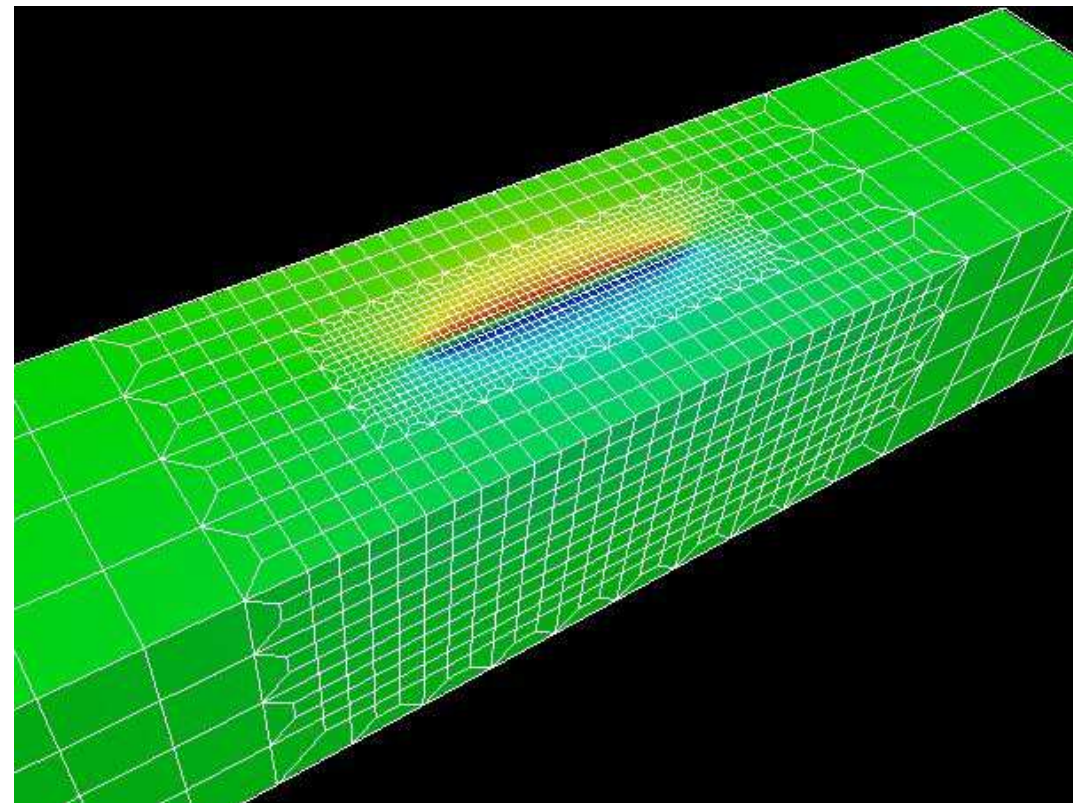
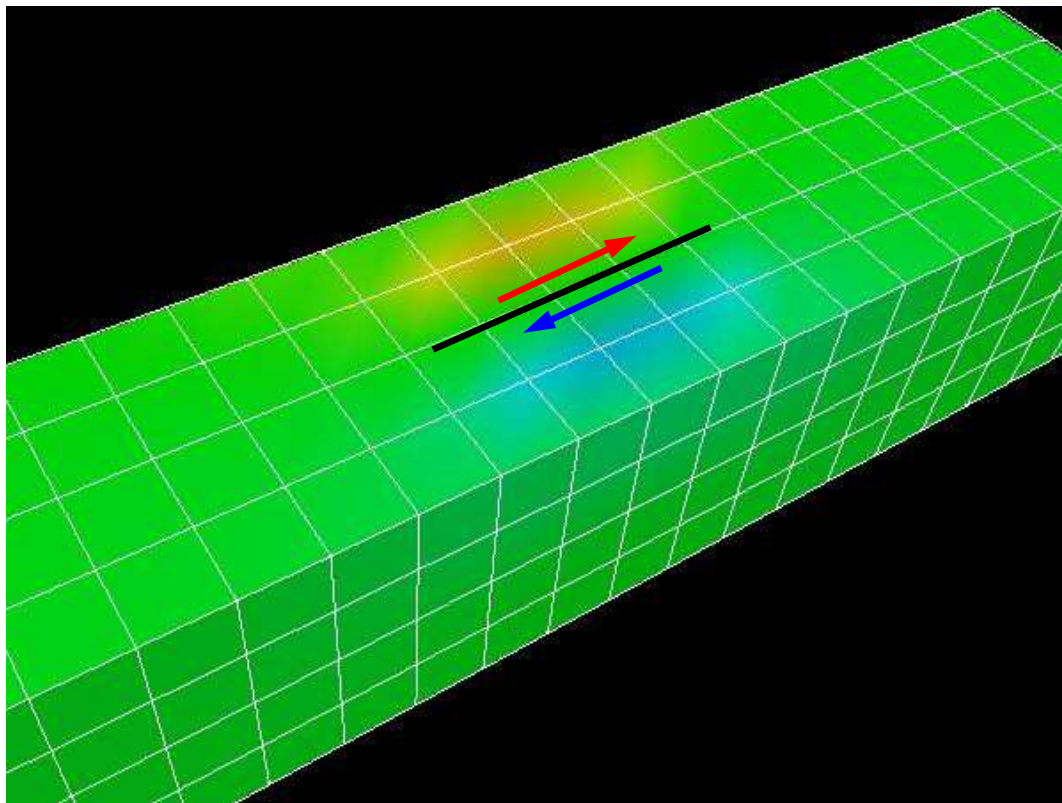


個別要素法
Discrete Element Method
DEM

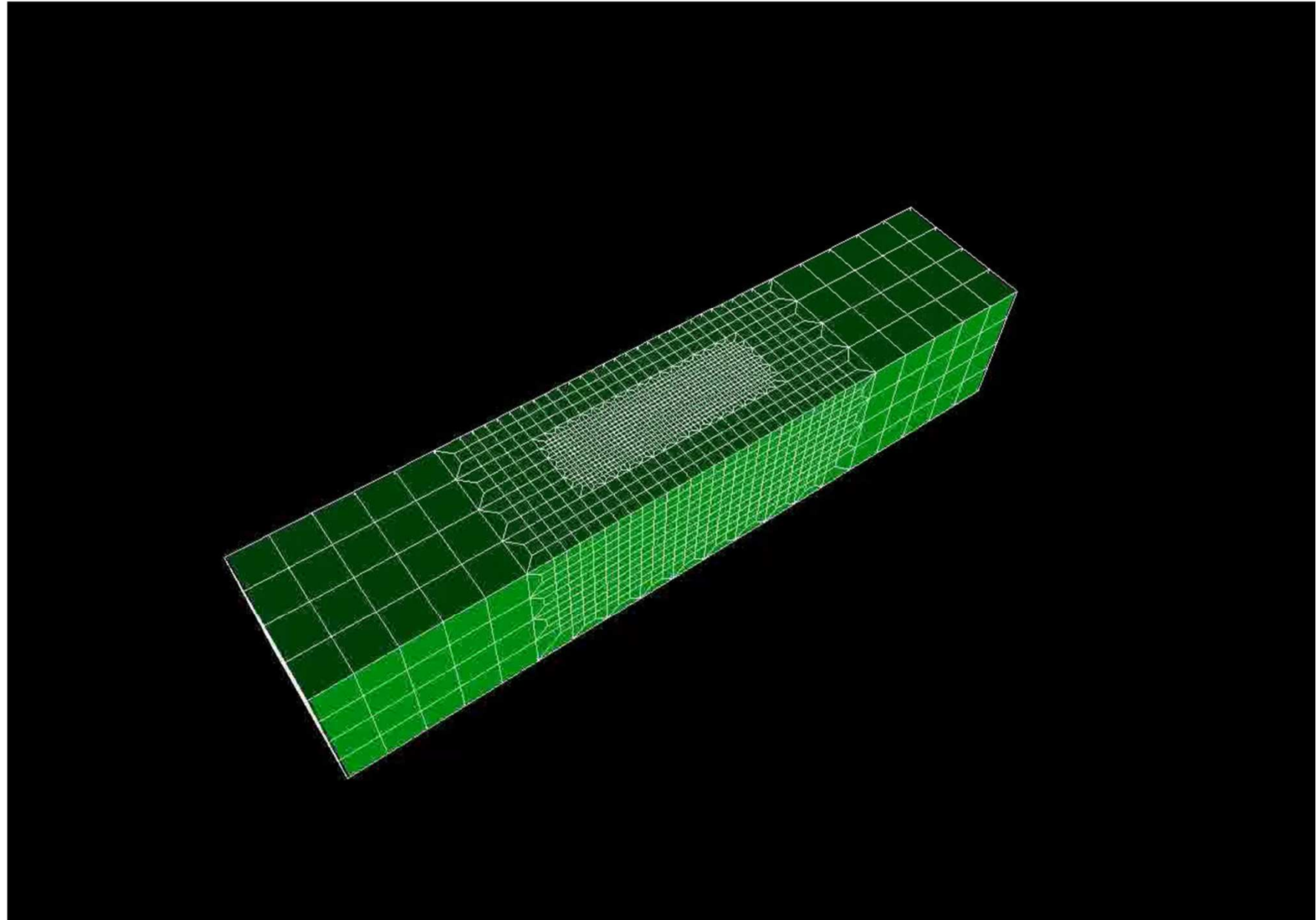
3D Simulations for Earthquake Generation Cycle

San Andreas Faults, CA, USA

Stress Accumulation at Transcurrent Plate Boundaries
Adaptive Mesh Refinement (AMR)

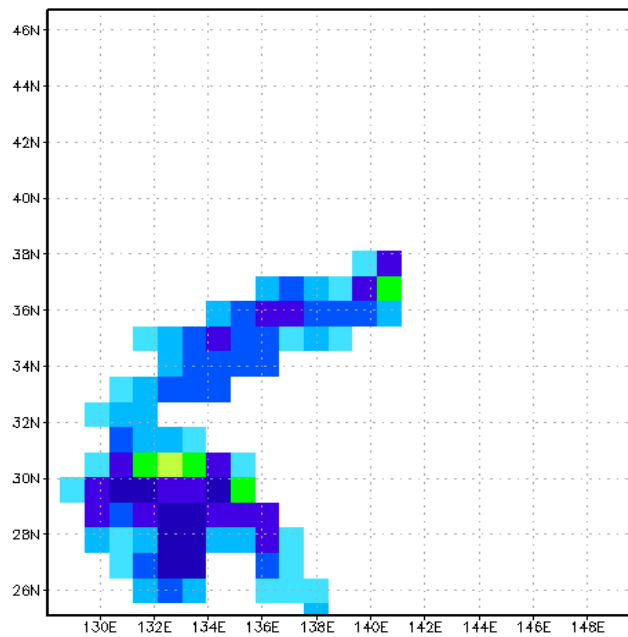


Adaptive FEM: High-resolution needed at meshes with large deformation (large accumulation)

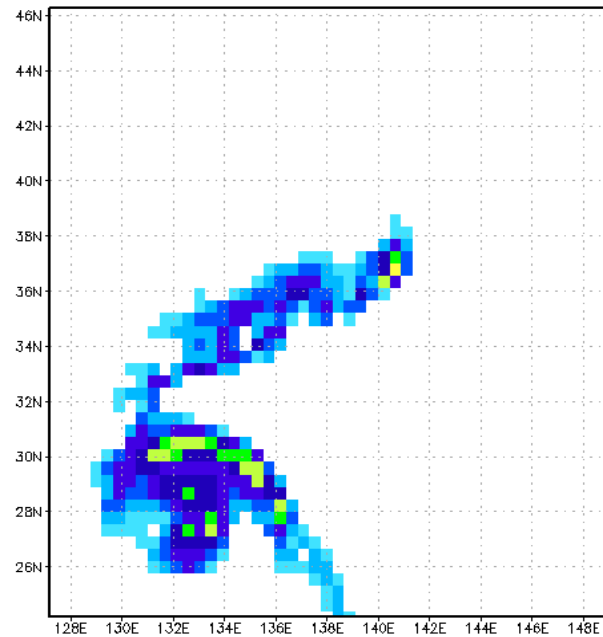


Typhoon Simulations by FDM

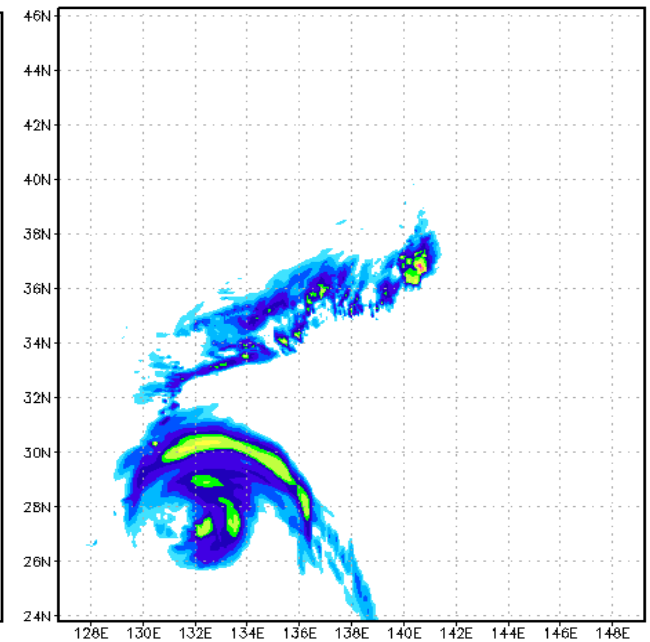
Effect of Resolution



$\Delta h = 100\text{km}$



$\Delta h = 50\text{km}$



$\Delta h = 5\text{km}$

Simulation of Geologic CO₂ Storage

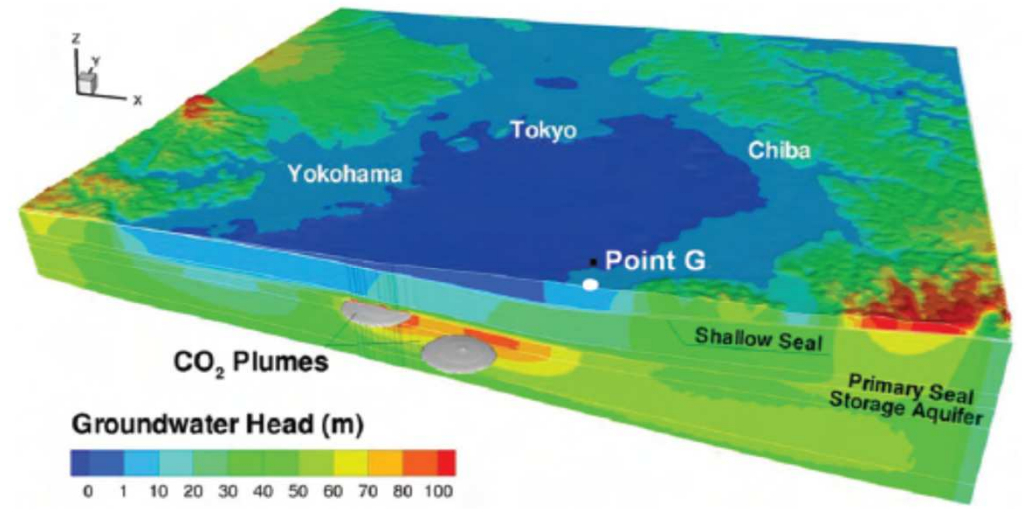
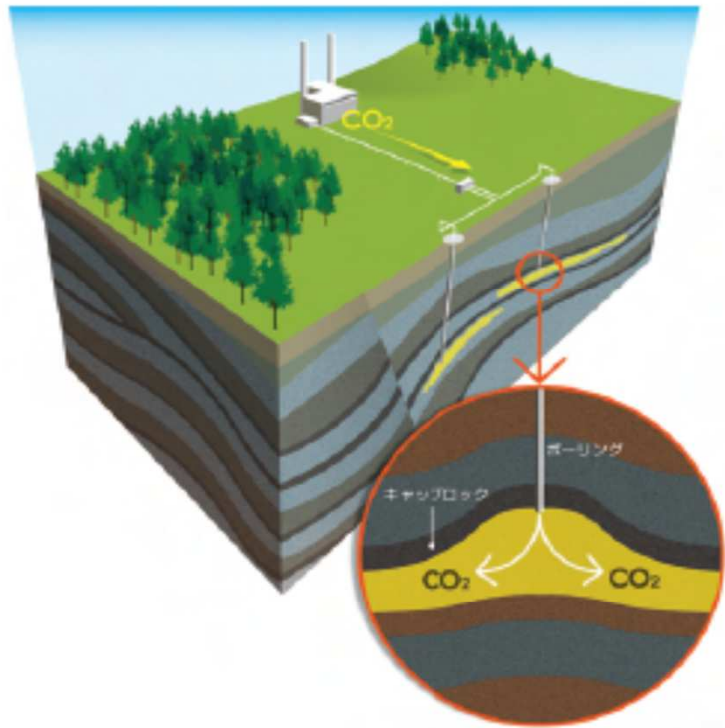
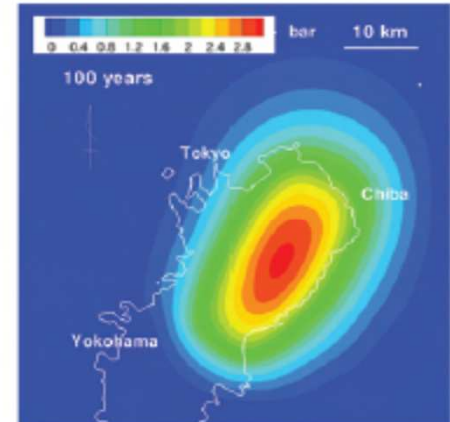
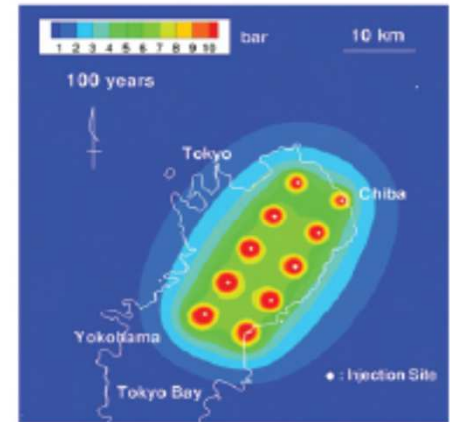
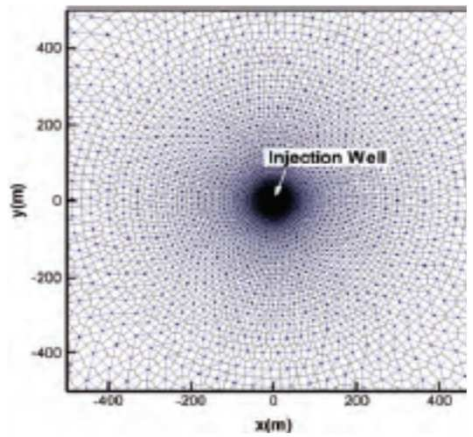
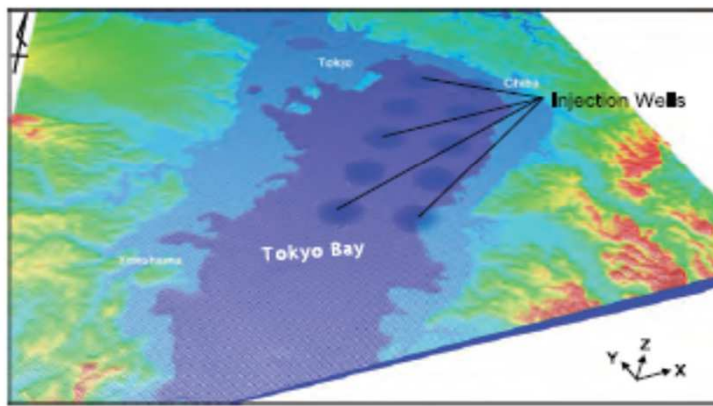


図-4 CO₂ 圧入後の地下水圧 (全水頭換算) の分布 (100 年後)



(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 CO₂ 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

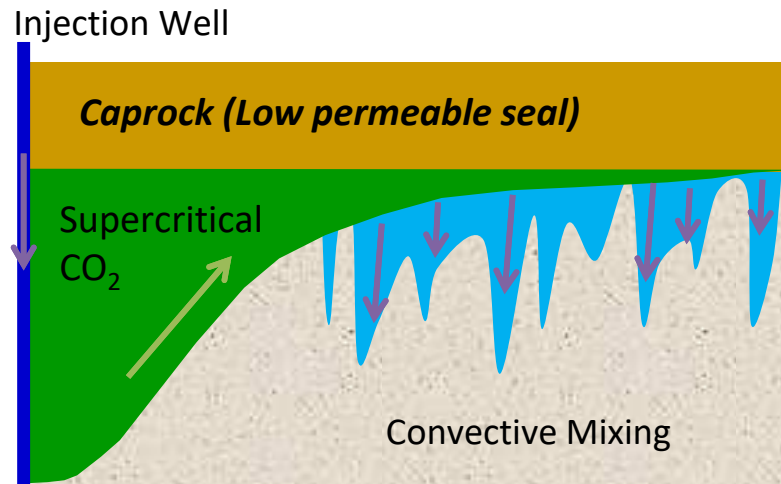
Software

Hardware

Simulation of Geologic CO₂ Storage

- Science
 - Behavior of CO₂ in supercritical state at deep reservoir
- PDE's
 - 3D Multiphase Flow (Liquid/Gas) + 3D Mass Transfer
- Method for Computation
 - TOUGH2 code based on FVM, developed by Lawrence Berkeley National Laboratory, USA
 - More than 90% of computation time is spent for solving large-scale linear equations with more than 10⁷ 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)

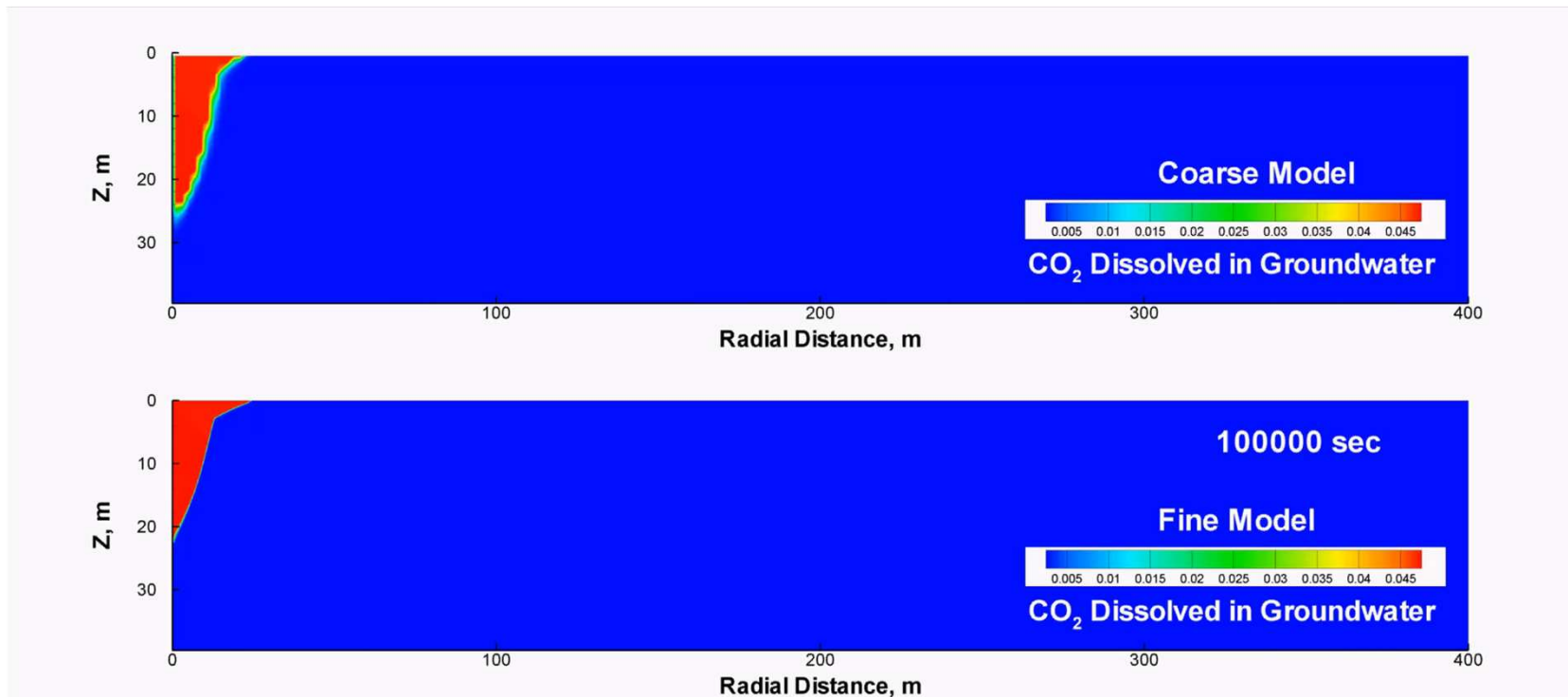
Diffusion-Dissolution-Convection Process



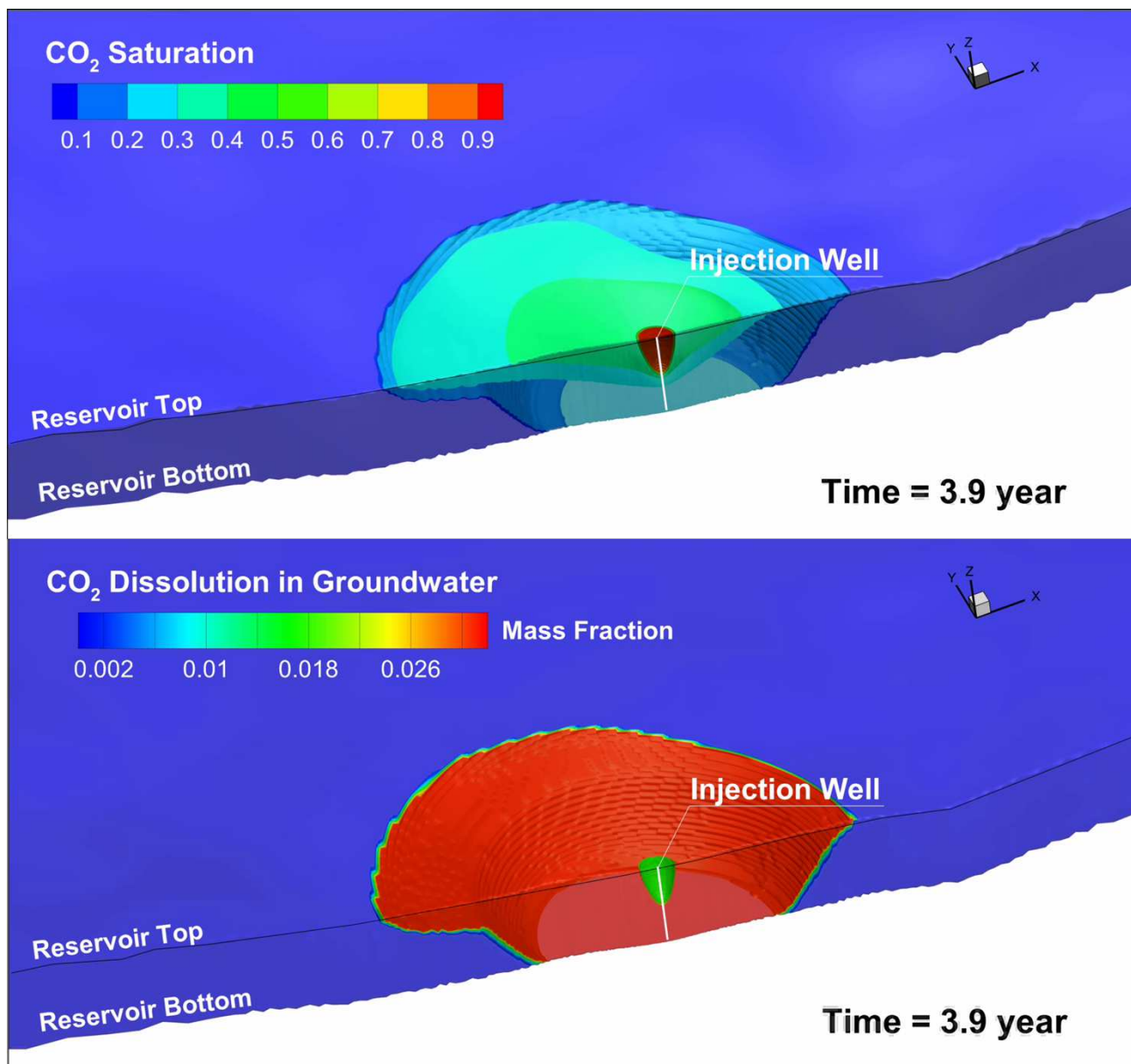
- Buoyant scCO₂ overrides onto groundwater
- Dissolution of CO₂ increases water density
- Denser fluid laid on lighter fluid
- Rayleigh-Taylor instability invokes convective mixing of groundwater

The mixing significantly enhances the CO₂ 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

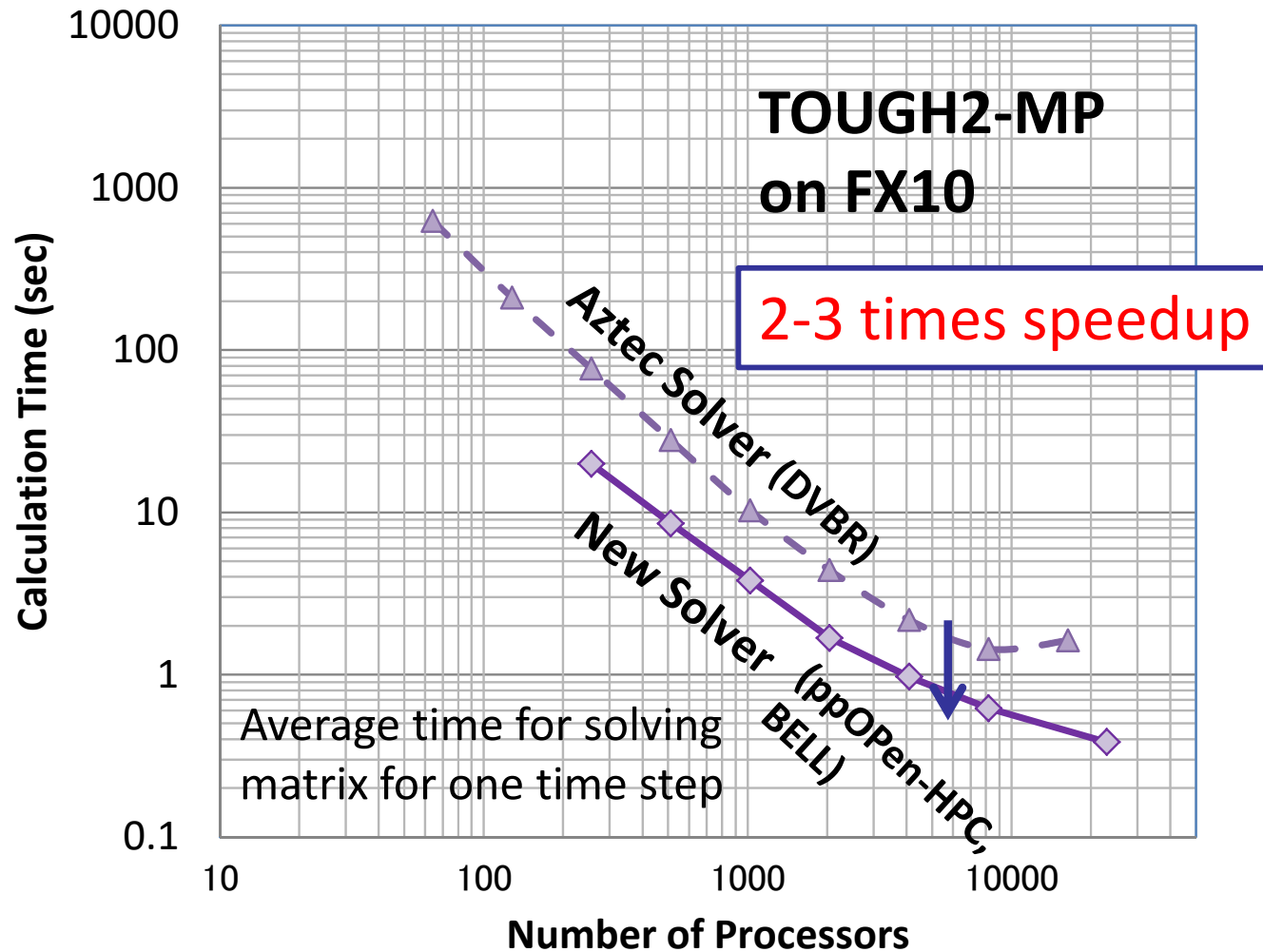
- Permeability: 100 md
- Porosity: 20%
- Pressure: 3MPa
- Temperature: 100°C
- Salinity: 15wt%

[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-yr simulation
- Onset time (10-20 yrs) is comparable to theoretical (linear stability analysis, 15.5yrs)

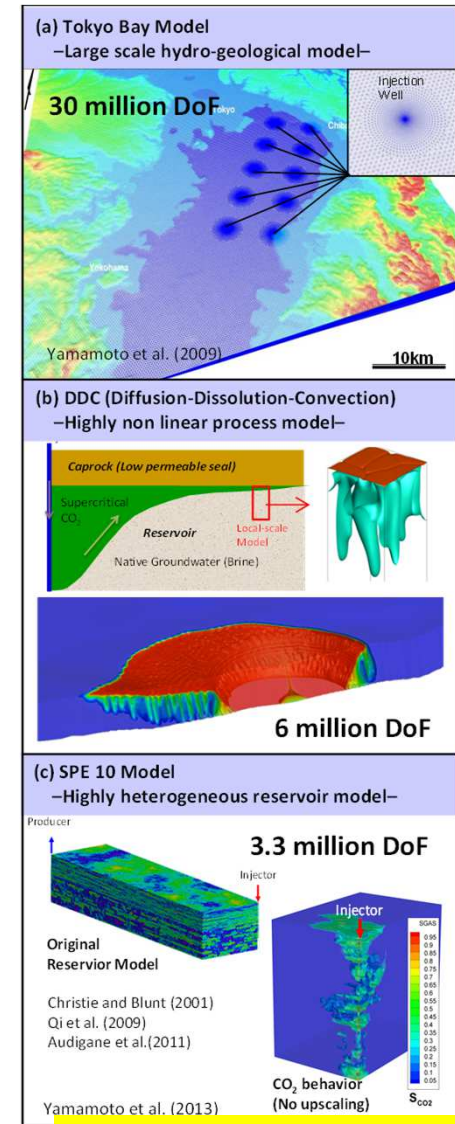
Simulation of Geologic CO₂ Storage

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

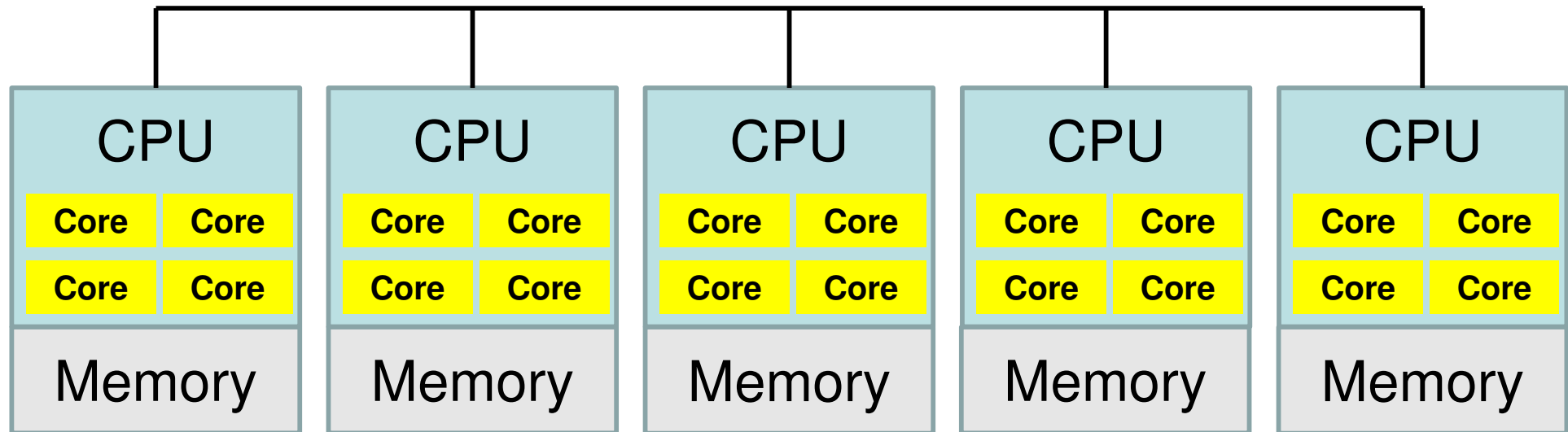
Motivation for Parallel Computing, again

- 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

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

Our Current Target: Multicore Cluster

Multicore CPU's are connected through network



- OpenMP

- ✓ Multithreading
- ✓ Intra Node (Intra CPU)
- ✓ Shared Memory

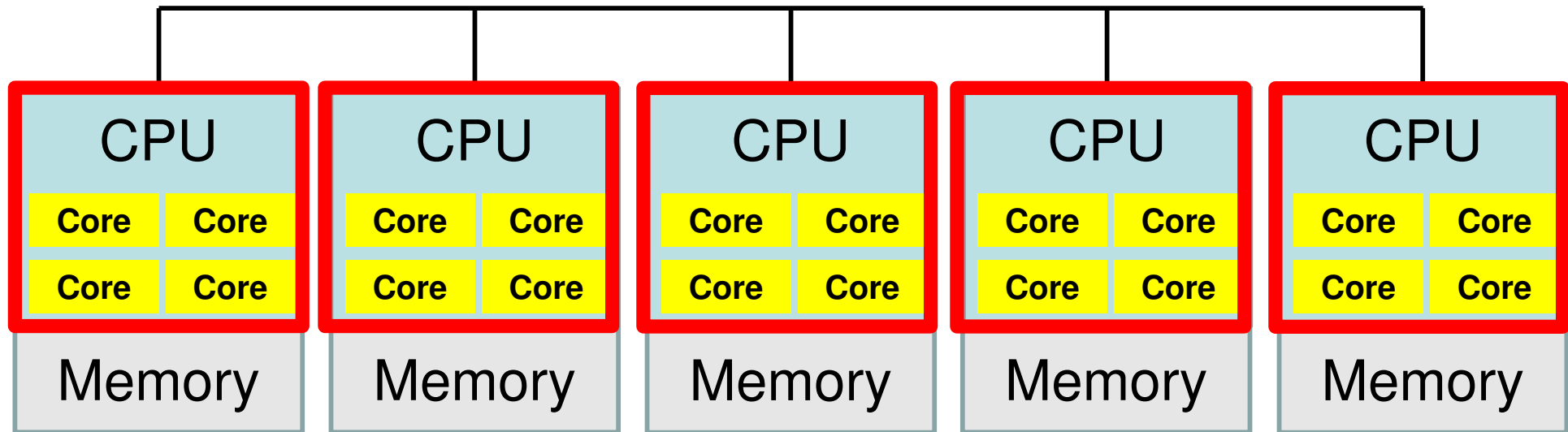
- MPI

- ✓ Message Passing
- ✓ Inter Node (Inter CPU)
- ✓ Distributed Memory



Our Current Target: Multicore Cluster

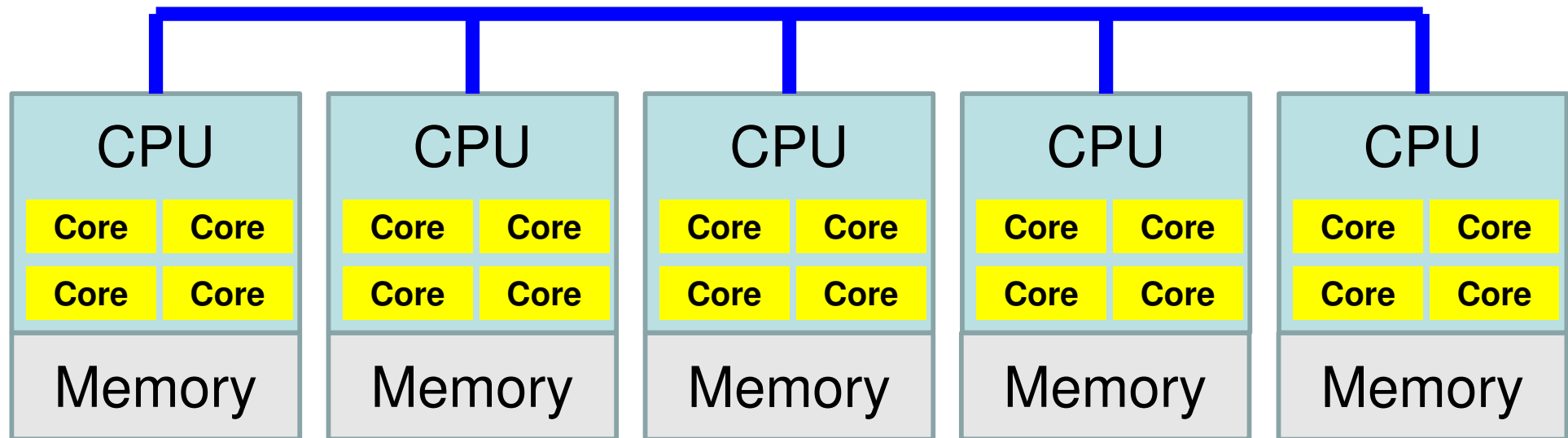
Multicore CPU's are connected through network



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

Our Current Target: Multicore Cluster

Multicore CPU's are connected through network

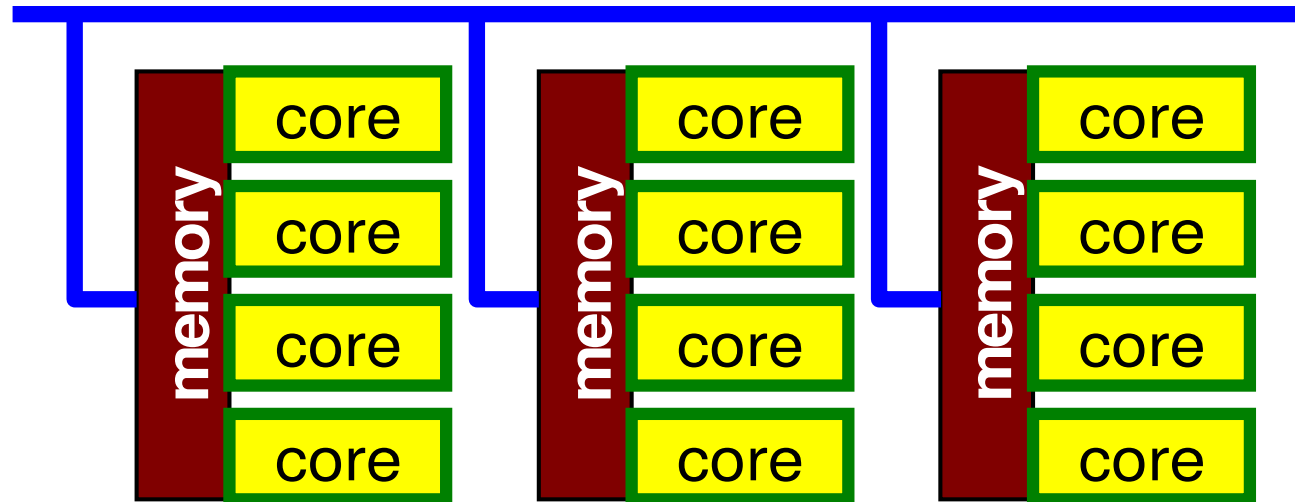


- OpenMP
 - ✓ Multithreading
 - ✓ Intra Node (Intra CPU)
 - ✓ Shared Memory
- MPI (after October)
 - ✓ Message Passing
 - ✓ Inter Node (Inter CPU)
 - ✓ Distributed Memory

Flat MPI vs. Hybrid

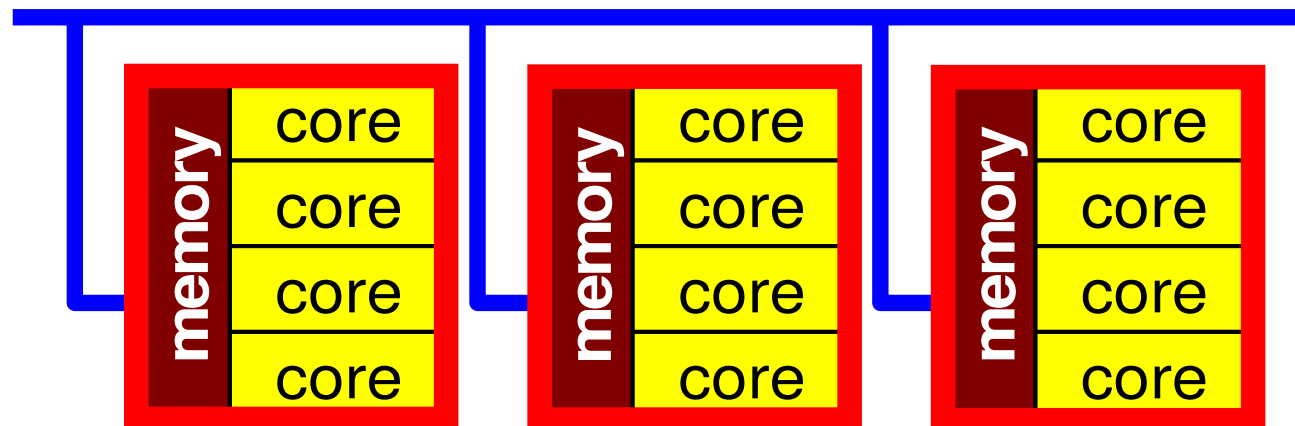
Flat-MPI: Each Core -> Independent

- MPI only
- Intra/Inter Node



Hybrid: Hierarchical Structure

- OpenMP
- MPI



Example of OpenMP/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

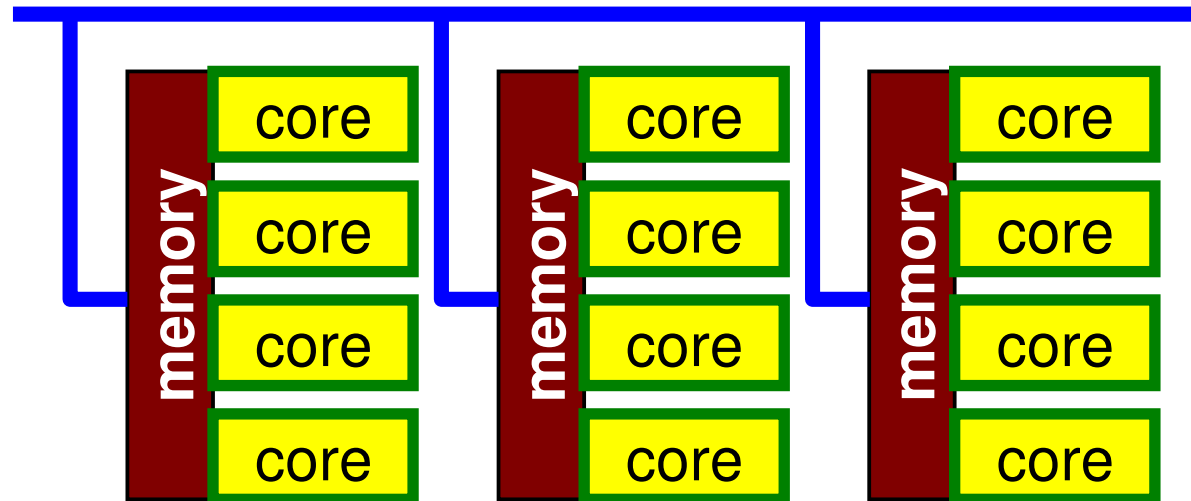
  call MPI_Isend (WS(istart+1-NE0), inum, MPI_DOUBLE_PRECISION, &
& NEIBPE(neib), 0, MPI_COMM_WORLD, &
& req1(neib), ierr)
enddo
```

Overview of This Class (1/3)

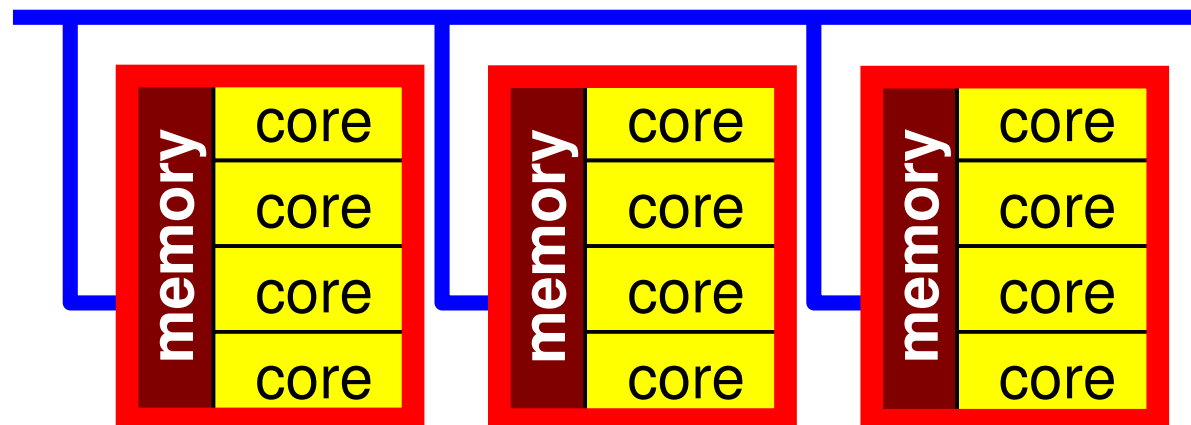
- <http://nkl.cc.u-tokyo.ac.jp/NTU2023W/>
- In order to make full use of modern supercomputer systems with multicore/manycore architectures, hybrid parallel programming with message-passing and multithreading is essential.
- While MPI is widely used for message-passing, OpenMP for CPU and OpenACC for GPU are the most popular ways for multithreading on multicore/manycore clusters.
- **In this class, we “parallelize” a finite-volume method code with Krylov iterative solvers for Poisson’s equation on Wisteria/BDEC-01 (Odyssey) System with Fujitsu/Arm A64FX at the University of Tokyo.**
 - **Because of limitation of time, we are focusing on multithreading by OpenMP.**

Flat MPI vs. Hybrid

Flat-MPI: Each PE -> Independent



Hybrid: Hierarchical Structure



Overview of This Class (2/3)

- We “parallelize” a finite-volume method (FVM) code with Krylov iterative solvers for Poisson’s equation.
- Derived linear equations are solved by ICCG (Conjugate Gradient iterative solvers with Incomplete Cholesky preconditioning), which is a widely-used method for solving linear equations.
- Because ICCG includes “data dependency”, where writing/reading data to/from memory could occur simultaneously, parallelization using OpenMP is not straight forward.
- We need certain kind of reordering in order to extract parallelism.

Overview of This Class (3/3)

- Lectures and exercise on the following issues will be conducted:
 - Overview of Finite-Volume Method (FVM)
 - Krylov Iterative Method, Preconditioning
 - Implementation of the Program
 - Introduction to OpenMP
 - **Reordering/Coloring Method**
 - Parallel FVM by OpenMP

Date	Hour	Content
February 14 (Tue), 2023	09:10-10:00	Introduction
	10:10-11:00	Finite Volume Method (FVM) (1/4)
	11:10-12:00	Finite Volume Method (FVM) (2/4)
	13:10-14:00	Finite Volume Method (FVM) (3/4)
	14:10-15:00	Finite Volume Method (FVM) (4/4)
	15:10-16:00	Introduction to OpenMP (1/4)
	16:10-17:00	Login to Odyssey
February 15 (Wed), 2023	09:10-10:00	Introduction to OpenMP (2/4)
	10:10-11:00	Introduction to OpenMP (3/4)
	11:10-12:00	Introduction to OpenMP (4/4)
	13:10-14:00	ICCG Method (1/3)
	14:10-15:00	ICCG Method (2/3)
	15:10-16:00	ICCG Method (3/3)
	16:10-17:00	Reordering (1/4)
February 16 (Thu), 2023	09:10-10:00	Reordering (2/4)
	10:10-11:00	Reordering (3/4)
	11:10-12:00	Reordering (4/4)
	13:10-14:00	Parallel FVM using OpenMP (1/4)
	14:10-15:00	Parallel FVM using OpenMP (2/4)
	15:10-16:00	Parallel FVM using OpenMP (3/4)
	16:10-17:00	Parallel FVM using OpenMP (4/4)

“Prerequisites”

- Fundamental physics and mathematics
 - Linear algebra, analytics
- Experiences in fundamental numerical algorithms
 - Gaussian Elimination, LU Factorization
 - Jacobi/Gauss-Seidel/SOR Iterative Solvers
 - Conjugate Gradient Method (CG)
- Experiences in programming by C/C++/Fortran
- **Experiences in Unix/Linux (vi or emacs)**
 - **If you are not familiar with Unix/Linux (vi or emacs), please try “Introduction Unix”, “Introduction emacs” in google.**
- User account of ECCS2016 must be obtained (later)
 - <https://www.ecc.u-tokyo.ac.jp/en/newaccount.html>

Keywords for OpenMP

- OpenMP
 - Directive based, (seems to be) easy
 - Many books
- Data Dependency
 - Conflict of reading from/writing to memory
 - Appropriate reordering of data is needed for “consistent” parallel computing
 - NO detailed information in OpenMP books: very complicated

Preparation

- Windows
 - WSL (Windows Subsystem for Linux) or Cygwin
 - ParaView
- MacOS, UNIX/Linux
 - ParaView
- Cygwin: <https://www.cygwin.com/>
- ParaView: <http://www.paraview.org>

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

Technical Issues: Future of Supercomputers

- Power Consumption
 - 1MW=1,000kW~ 1M USD/yr, 100M JPY/yr
- Reliability, Fault Tolerance, Fault Resilience
- Scalability (Parallel Performance)

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

