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/21s/

Descriptions of the Class

- Technical & Scientific Computing I (4820-1027)
  - 科学技術計算 I
  - Department of Mathematical Informatics
- Special Lecture on Computational Alliance I(4810-1215)
  - 計算科学アライアンス特別講義 I
  - Department of Computer Science
- Multithreaded Parallel Computing (3747-110)
  - スレッド並列コンピューティング
  - Department of Electrical Engineering & Information Systems
- This class is certificated as the category “D” lecture of "the Computational Alliance, the University of Tokyo"

- 2009-2014
  - Introduction to FEM Programming
    - FEM: Finite-Element Method：有限要素法
  - Summer (I) : FEM Programming for Solid Mechanics
  - Winter (II): Parallel FEM using MPI
    - The 1st part (summer) is essential for the 2nd part (winter)
- Problems
  - Many new (international) students in Winter, who did not take the 1st part in Summer
  - **They are generally more diligent than Japanese students**
- 2015
  - Summer (I) : Multicore programming by OpenMP
  - Winter (II): FEM + Parallel FEM by MPI/OpenMP for Heat Transfer
    - Part I & II are independent (maybe...)
- **2017**
  - **Lectures are given in English**
  - **Reedbush-U Supercomputer System**
- **2020**
  - **Oakbridge-CX Supercomputer System, Online**

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 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)

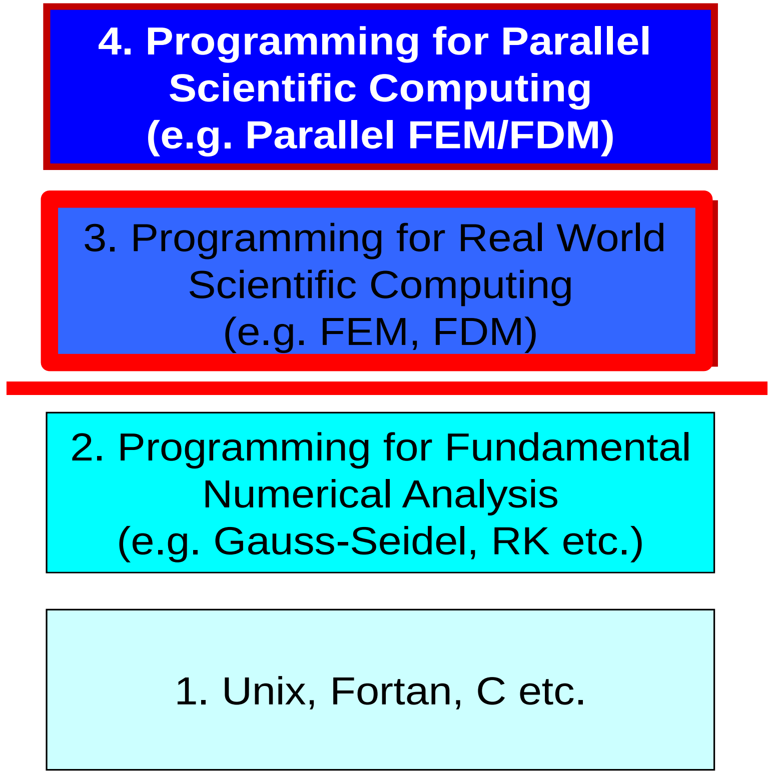
<u>Big gap here !!</u>

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

Kengo Nakajima 中島研吾 (1/2)

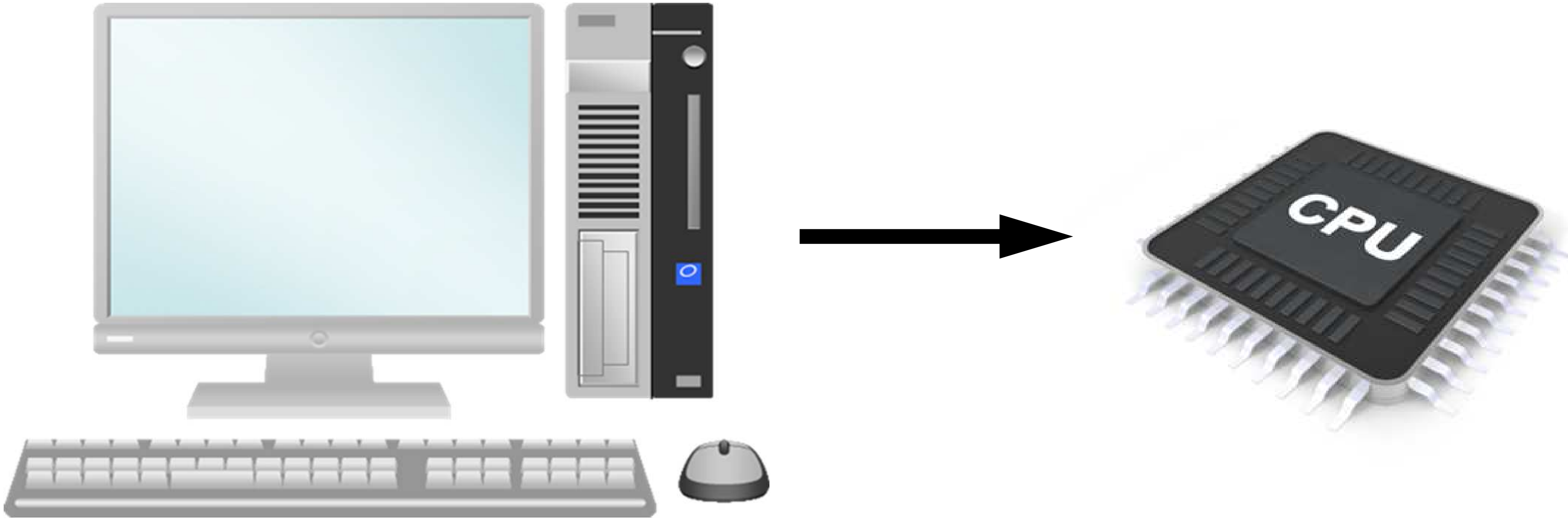
- Current Position
  - Professor, Supercomputing Research Division, Information Technology Center, The University of Tokyo（情報基盤センター）
    - Department of Mathematical Informatics, Graduate School of Information Science & Engineering, The University of Tokyo（情報理工・数理情報学）
    - Department of Electrical Engineering 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

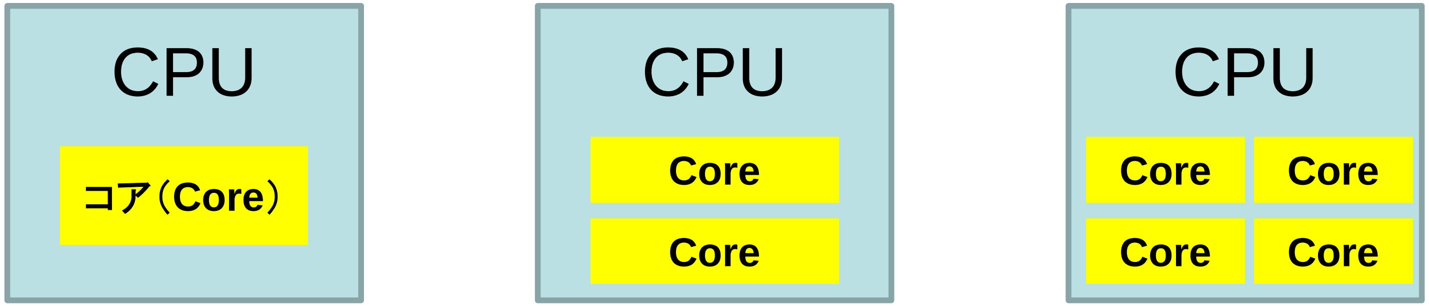
- **Supercomputers and Computational Science**
- Overview of the Class
- Future Issues

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 (or more) instructions per clock

Multicore CPU

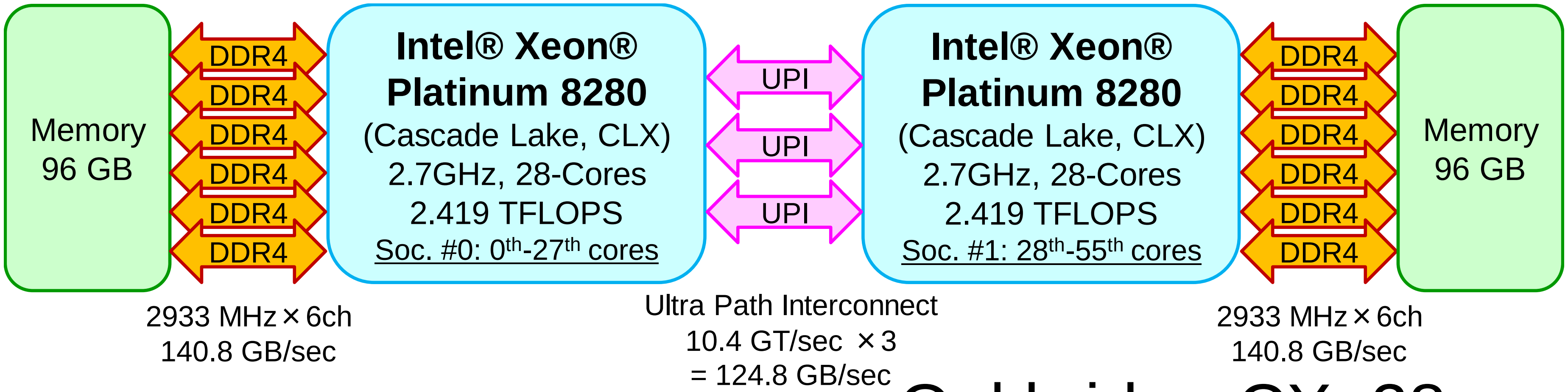


Single Core
1 cores/CPU

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

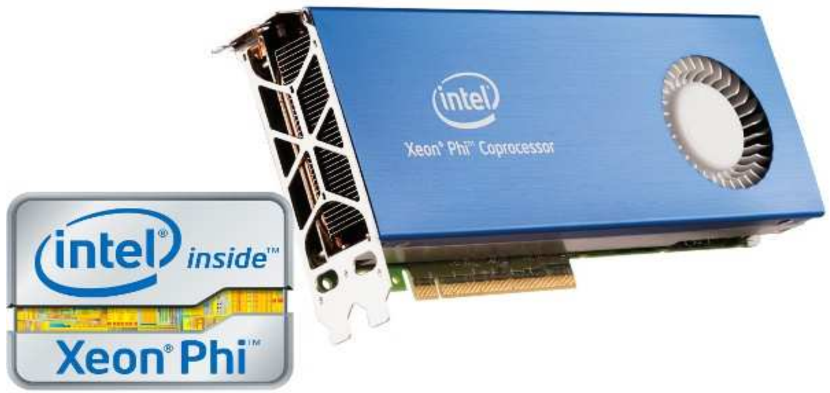


- GPU: Manycore
  - $O(10^1)$-$O(10^2)$ cores
- More and more cores
  - Parallel computing

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

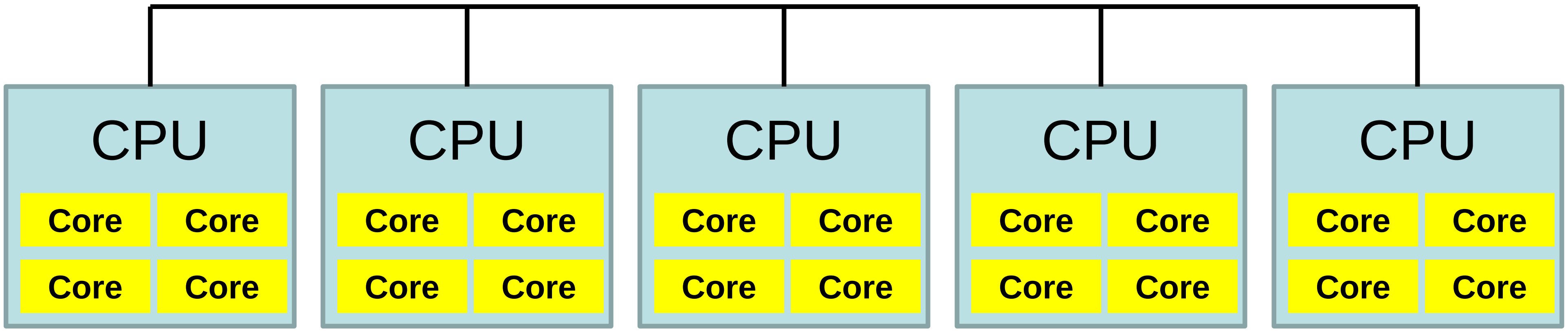
GPU/Manycores

- GPU : Graphic Processing Unit
  - GPGPU: General Purpose GPU
  - O(10^2) cores
  - High Memory Bandwidth
  - (was) 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 NOT needed
    - Needed in the 1st generation

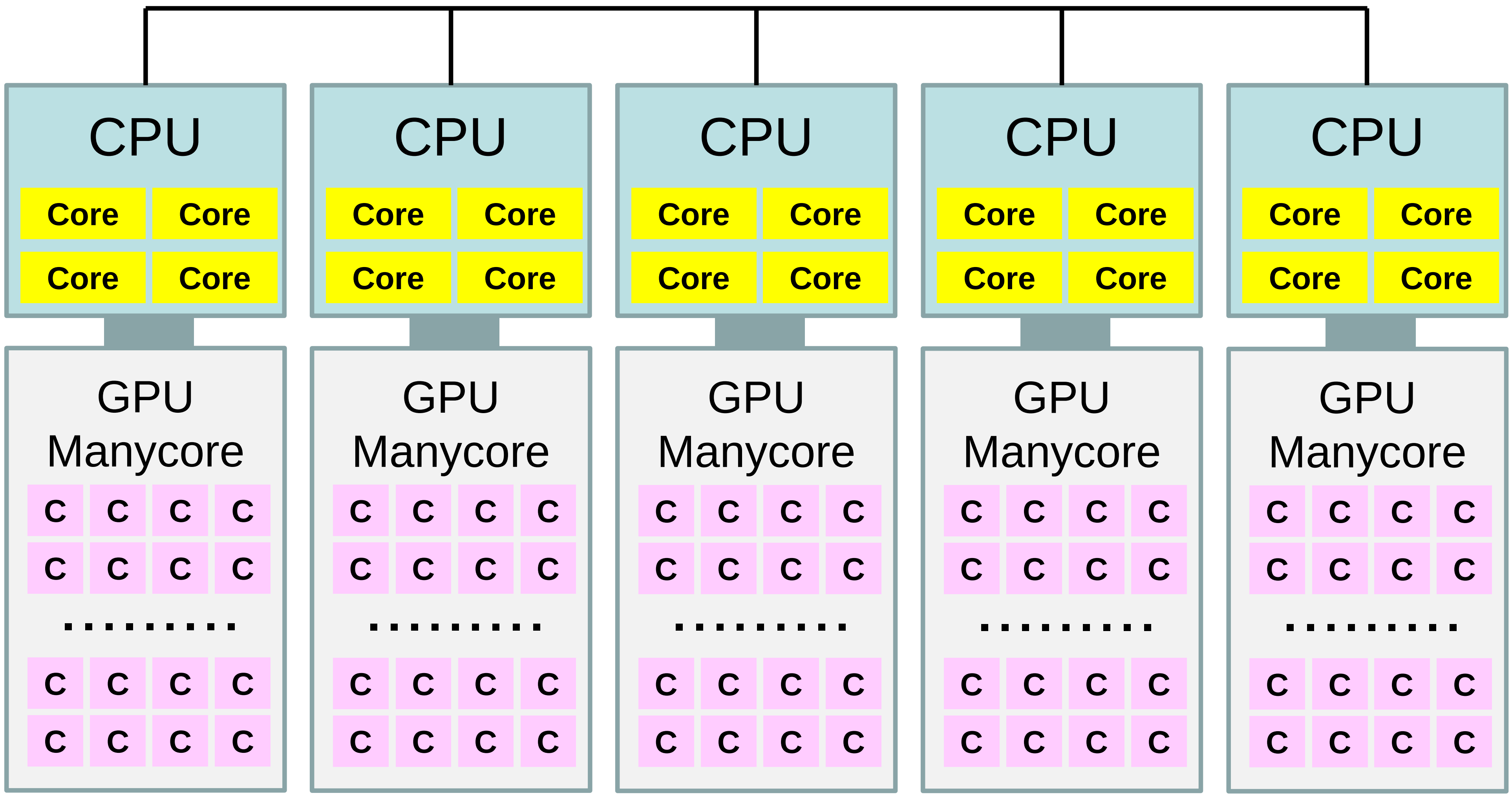


Parallel Supercomputers

Multicore CPU’s are connected through network



Supercomputers with Heterogeneous/Hybrid Nodes



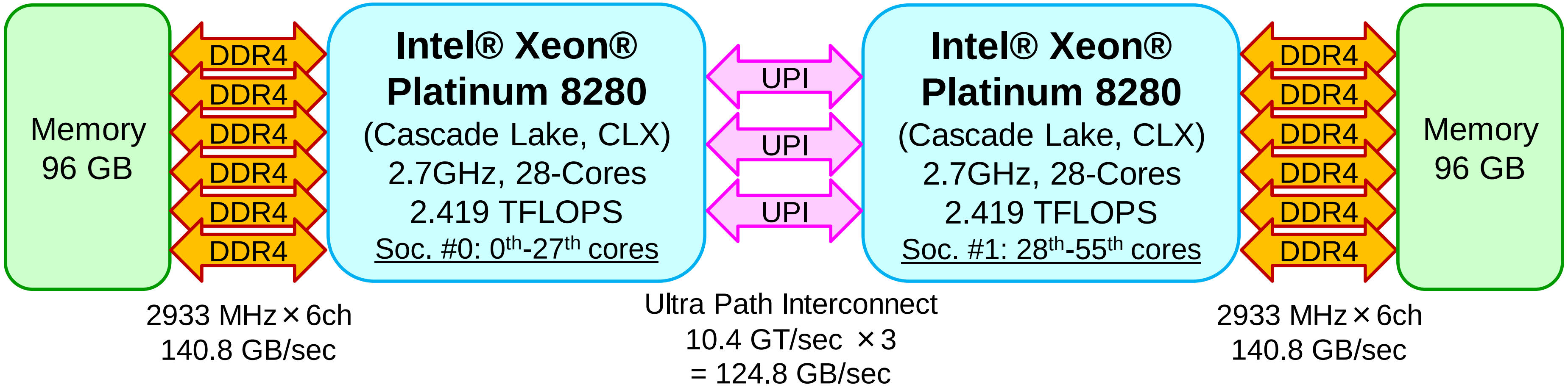
Performance of Supercomputers

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

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

Peak Performance of Oakbridge-CX

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



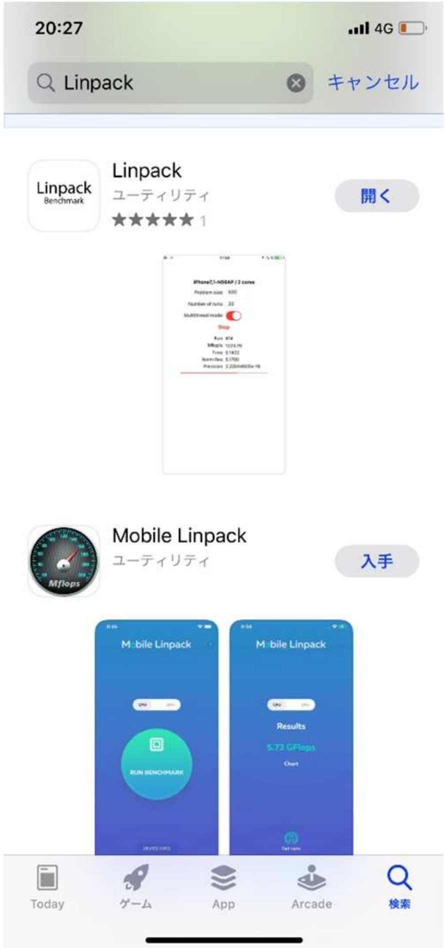
- 2.7 GHz
  - 32 DP (Double Precision) FLOP operations per Clock
- Peak Performance (1 core)
  - 2.7 × 32= 86.4 GFLOPS
- Peak Performance
  - 1-Socket, 28 cores: 2,419.2 GFLOPS= 2.419 TFLOPS
  - 2-Sockets, 56 cores: 4,838.4 GFLOPS= 4.838 TFLOPS 1-Node

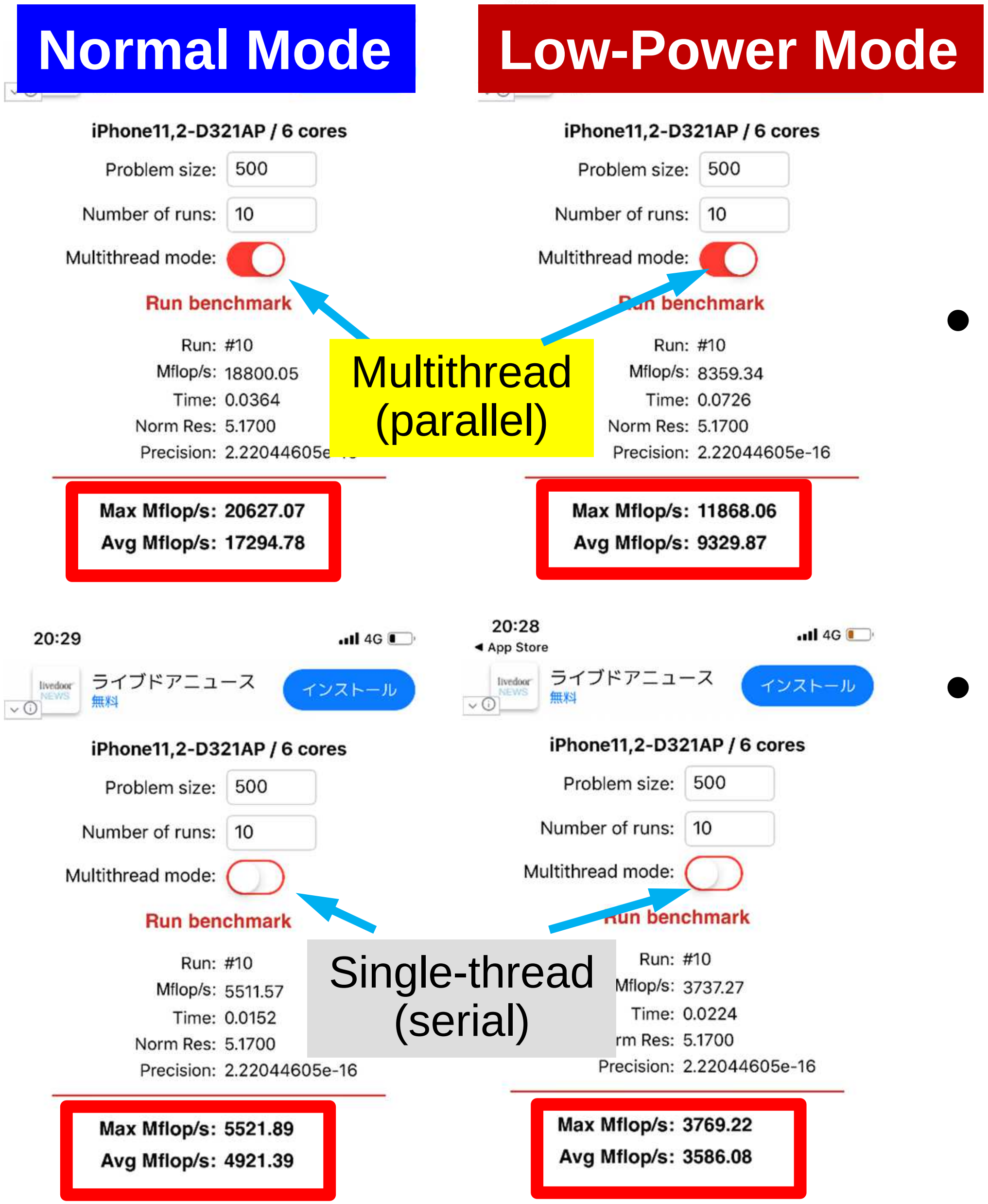
TOP 500 List

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
  - Available in iPhone and Android
- Oakbridge-CX (OBCX) is 69th in the TOP 500 (November 2020)

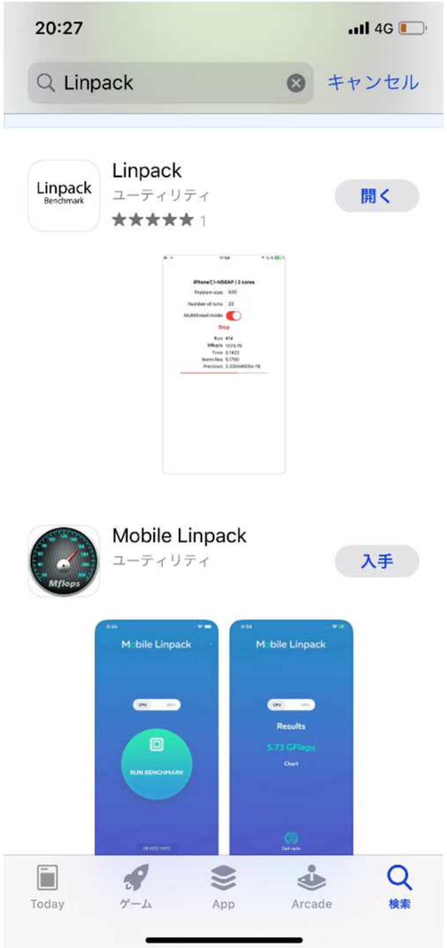
Linpack on My iPhone XS

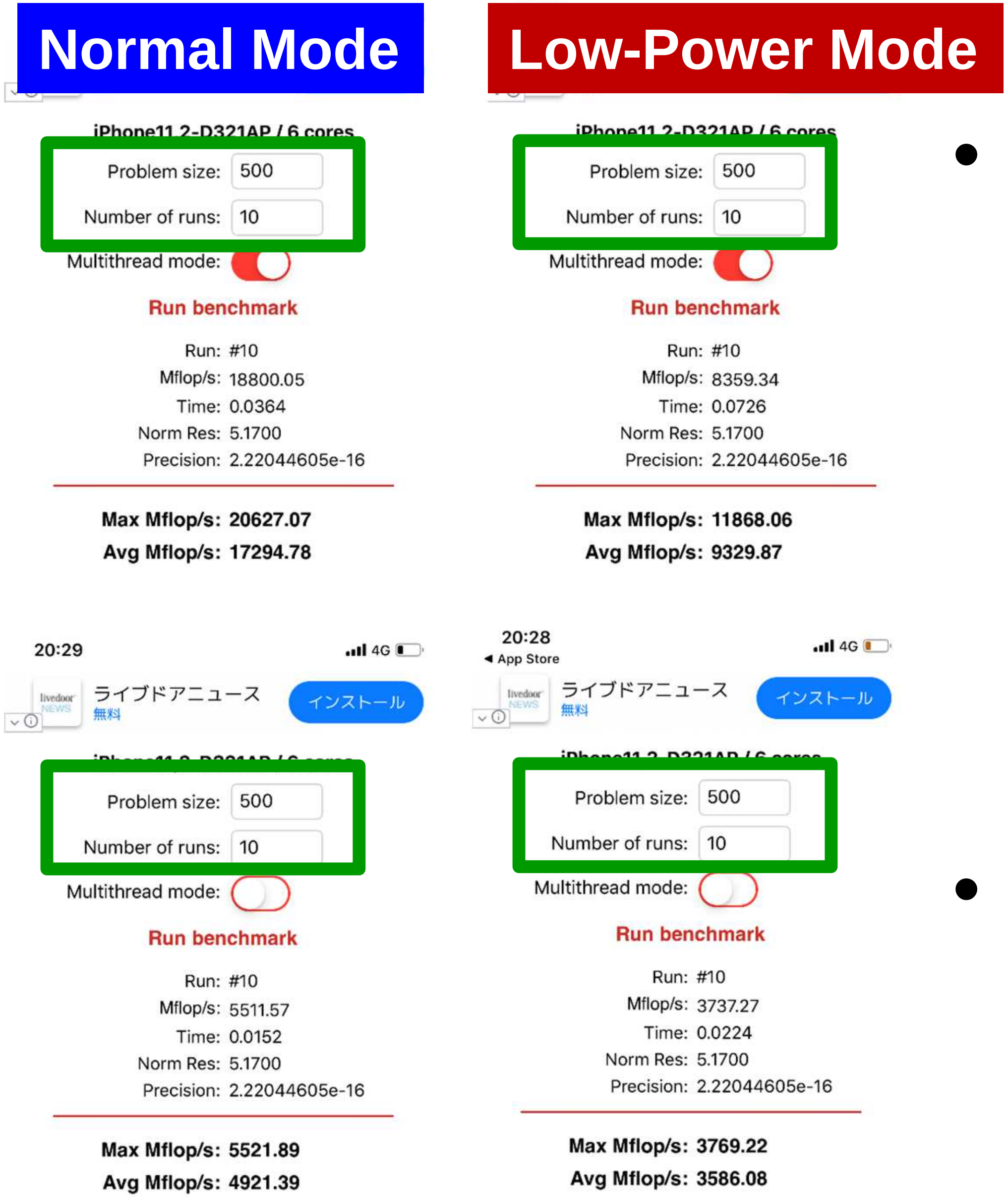




- Performance of my iPhone XS is about 20,000 Mflops
  - OBCX: 4.84 Tflops
- Cray-1S
  - Supercomputer of my company in 1985 with 80 Mflops
  - I do not know the price, but we had to pay 10 USD for 1 sec. computing !

Linpack on My iPhone XS





- 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 !!

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

56^{th} TOP500 List (Nov, 2020) Oakbridge-CX (OBCX) is 69th

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	**Supercomputer Fugaku, 2020, Japan** RIKEN Center for Computational Science (R-CCS)	Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu	7,630,848	442,010 (= 442.01 PF)	537,212	**29.899**
2	**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	200,795	**10,096**
3	**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**
4	**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**
5	**Selene, 2020, USA** NVIDIA Corporation	DGX A100 SuperPOD, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband	555,520	63,460	79,125	**2,646**
6	**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**
7	**JUWELS Booster Module, 2020, Germany** Forschungszenturm Juelich (FZJ)	Bull Sequana XH2000, AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR Infiniband	277,760	27,580	34,568.6	**1,344**
8	**HPC5, 2020, Italy** Eni S.p.A (Ente Nazionale Idrocarburi)	PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, Dell EMC	669,760	35,450	51,720.8	**2,252**
9	**Frontera, 2019, USA** Texas Advanced Computing Center	Dell C6420, Xeon Platinum 8280 28c 2.7GHz, Mellanox Infiniband HDR	448,448	23,516	38,746	
10	**Dammam-7, 2020, Saudi Arabia** Saudi Aramco	Cray CS-Storm, Xeon Gold 6248 20C, 2.5GHz, NVIDIA V100 SXM2, Infiniband HDR 100, HPE	387,872	21,230	27,154	**2,384**
22	**Oakforest-PACS, 2016, Japan** Joint Center for Advanced High Performance Computing	Fujitsu PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path	556,104	13,556	24,913	2,719

HPCG Ranking (November, 2020)

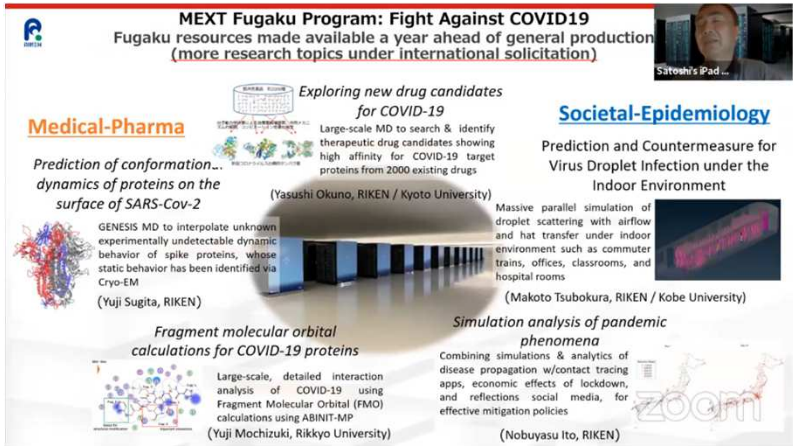
	Computer	Cores	HPL Rmax (Pflop/s)	TOP500 Rank	HPCG (Pflop/s)
1	Fugaku	7,630,848	442,010	1	16.004
2	Summit	2,414,592	148.600	2	2.926
3	Sierra	1,572,480	94.640	3	1.796
4	Selene	555,520	63,460	5	1.622
5	JUWELS Booster Module	449,280	44,120	7	1.275
6	Dammam-7	672,520	22,400	10	0.881
7	HPC5	669,760	35,450	8	0.860
8	TOKI-SORA (JAXA, Japan)	276,480	16,592	19	0.614
9	Trinity	979,072	20,158	13	0.546
10	Plasma Simulator (NIFS, Japan, SX-Aurora TSUBASA)	34,560	7,892	33	0.529
16	Oakforest-PACS	556,104	13.555	22	0.385

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

Green 500 Ranking (November, 2020)

http://www.top500.org/

	TOP 500 Rank	System	Accelerator	Cores	HPL Rmax (Pflop/s)	Power (kW)	GFLOPS/W
1	172	NVIDIA DGX SuperPOD, USA	NVIDIA A100	19,840	2,356	90	26.195
2 (1)	332	MN-3, Preferred Networks, Japan	MN-Core	1,664	1,653	65	*26.039
3	7	JUWELS Booster Module, Germany	NVIDIA A100	449,280	44,120	1,764	25.008
4	148	Spartan2, France	NVIDIA A100	23,040	2,566	106	24.262
5 (7)	5	Selena, NVIDIA, USA	NVIDIA A100	555,520	63,460	2,646	23.983
6 (4)	241	A64FX Prototype, Fujitsu, Japan		36,864	1.999	118	16.876
7 (5)	29	AiMOS, USA	NVIDIA V100	130,000	8.339	512	16.285
8 (6)	8	HPC5, Italy	NVIDIA V100	669,760	35.450	2,252	15.740
9 (7)	460	Satori, USA	NVIDIA V100	34,040	1.464	94	15.574
10 (9)	1	Fugaku, Fujitsu, Japan		7,630,848	442,010	29,899	*15.418
(13)	Nov.'17	Reedbush-L, U.Tokyo, Japan	NVIDIA P100	16,640	806	79	10.167
(19)		Reedbush-H, U.Tokyo, Japan	NVIDIA P100	17,760	802	94	8.576

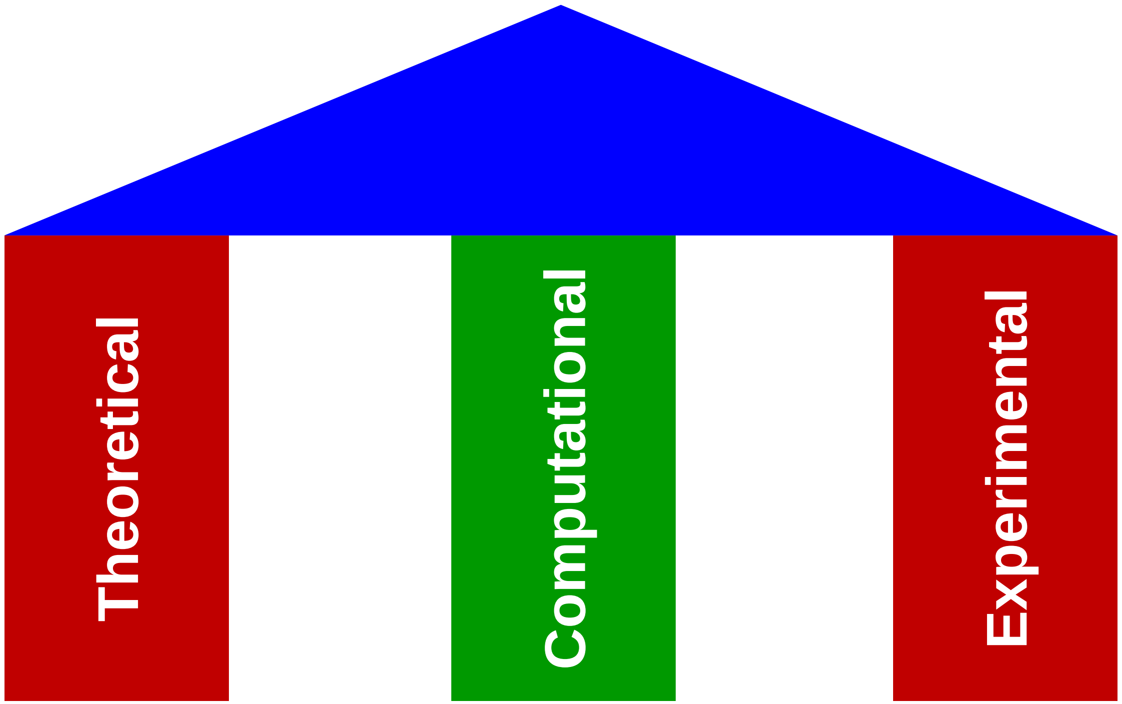


Benchmark	Category
TOP 500	Computing
HPCG	Computing
HPL-AI	AI/ML
Graph 500	Big Data

Computational Science

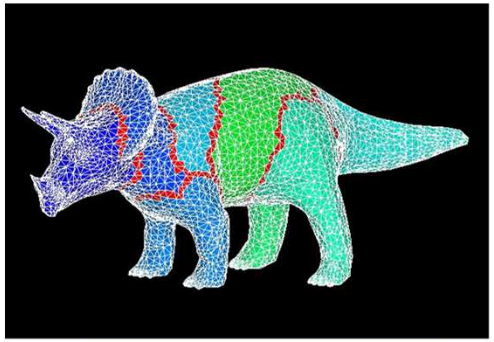
The 3rd Pillar of Science

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

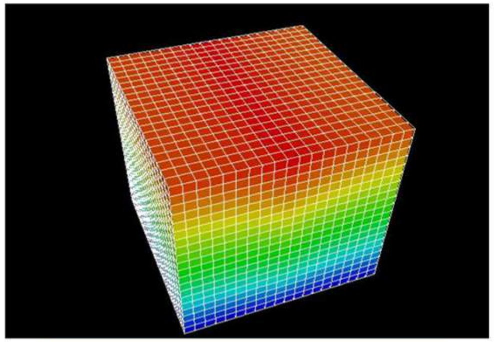


Methods for Scientific Computing

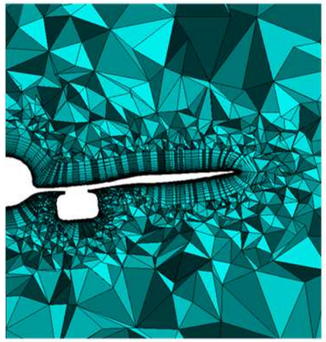
- Numerical solutions of PDE (Partial Diff. Equations)
- Grids, Meshes, Particles
  - Large-Scale Linear Equations
  - Finer meshes provide more accurate solutions



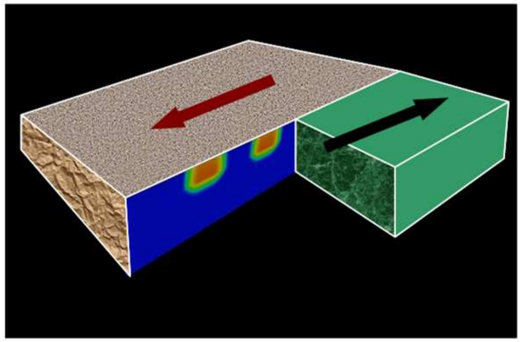
有限要素法
Finite Element Method
FEM



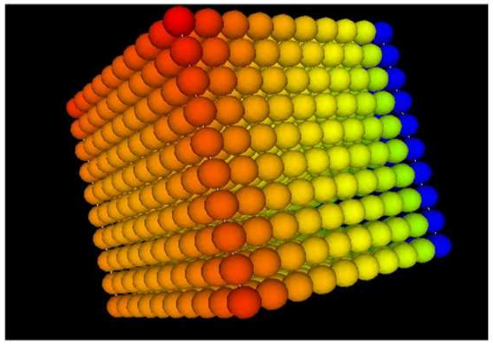
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



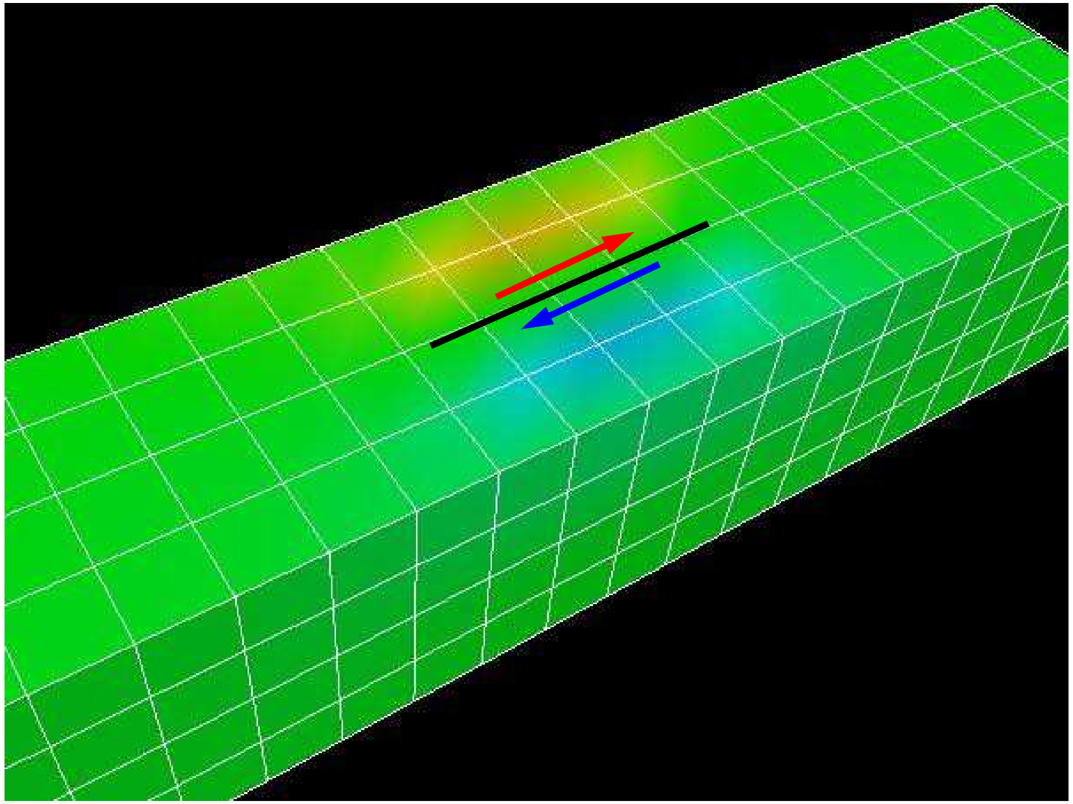
境界要素法
Boundary Element Method
BEM

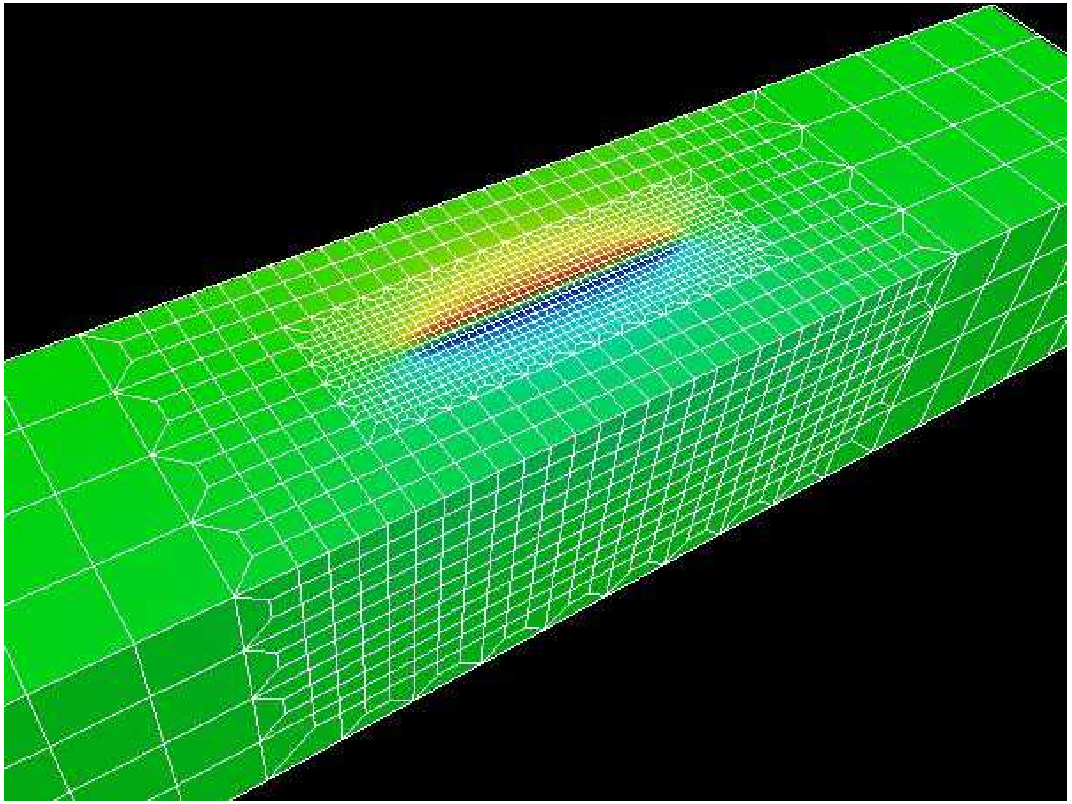


個別要素法
Discrete Element Method
DEM

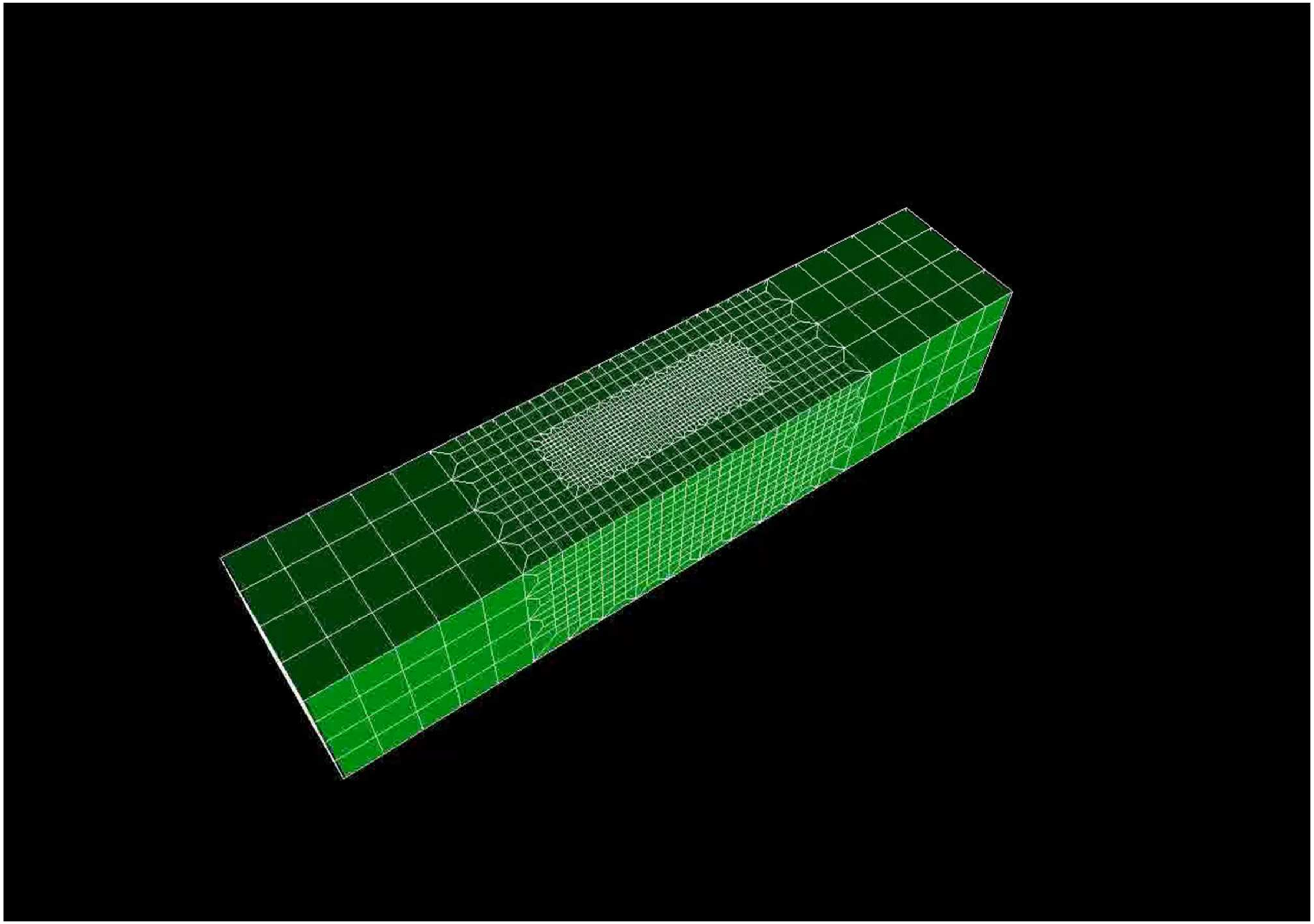
3D Simulations for Earthquake Generation Cycle
San Andreas Faults, CA, USA

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

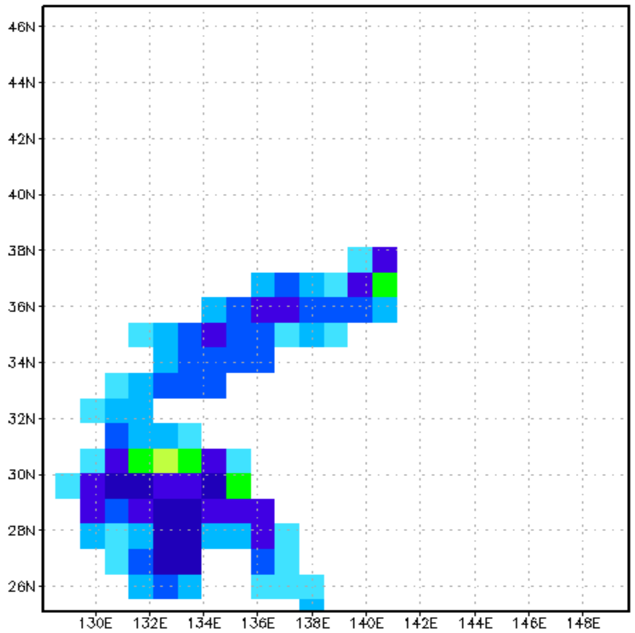




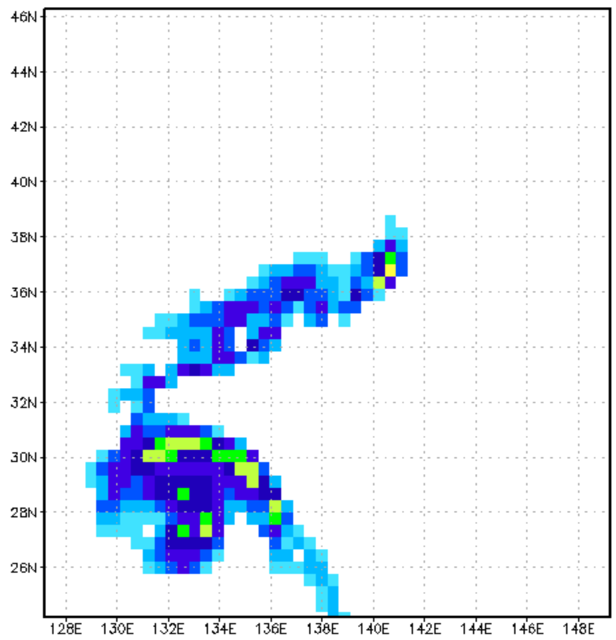
Adaptive FEM: High-resolution needed at meshes with large deformation (large accumulation)



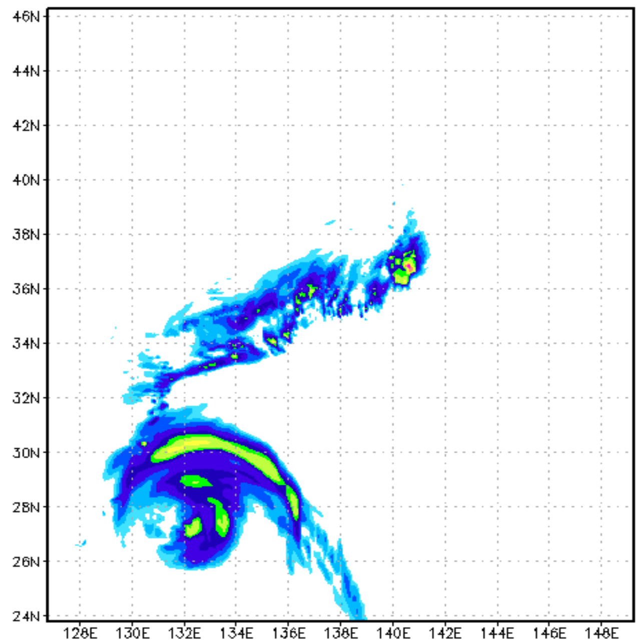
Typhoon Simulations by FDM
Effect of Resolution



Δh=100km



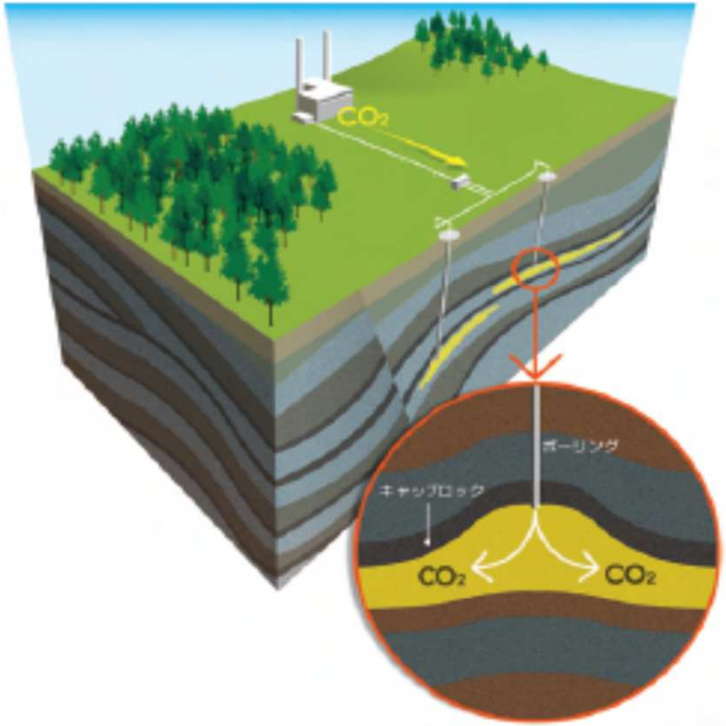
Δh=50km



Δh=5km

[JAMSTEC]

Simulation of Geologic CO_2 Storage



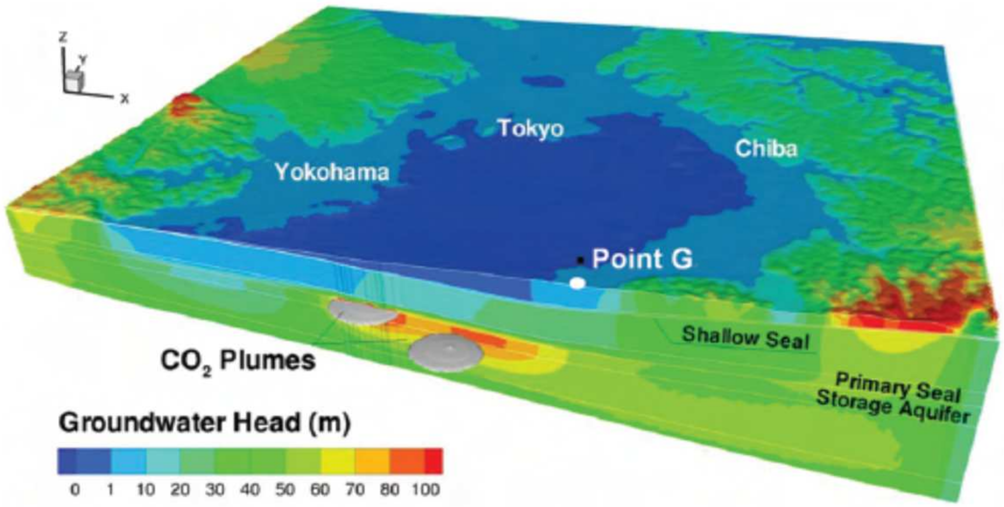
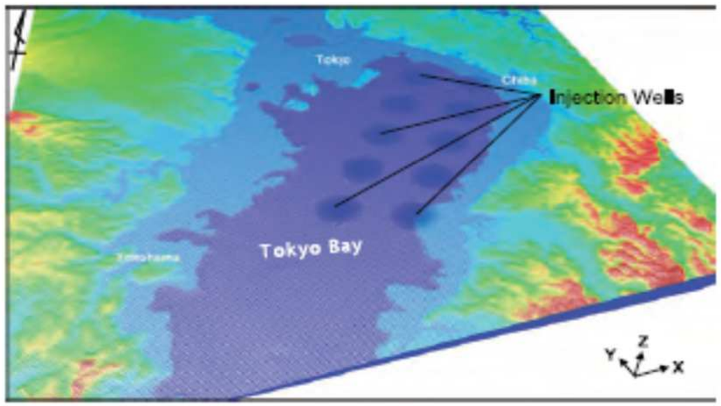
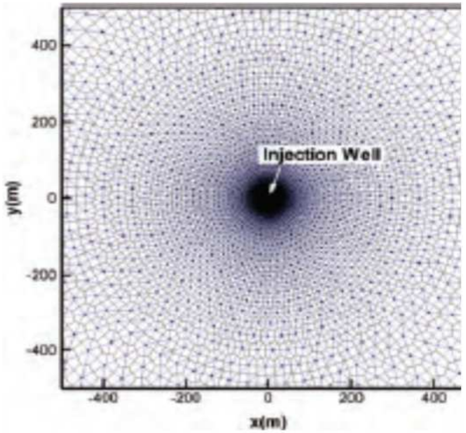
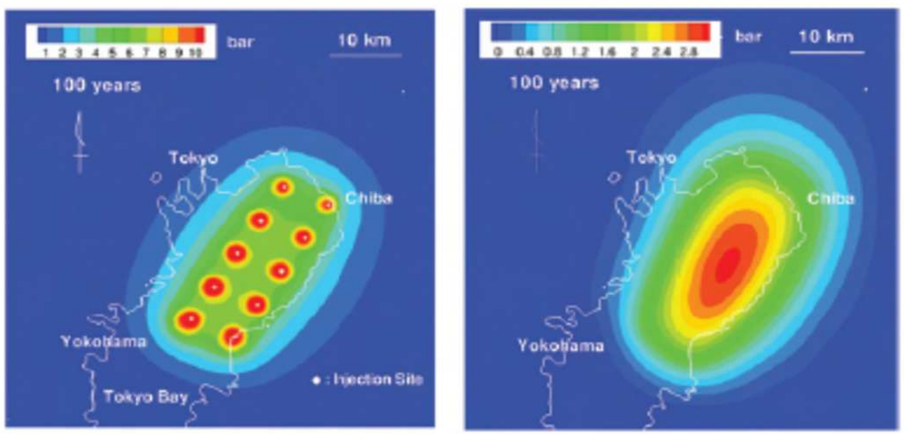


図 -4　CO_2 圧入後の地下水圧（全水頭換算）の分布（100 年後）







(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

Science

Modeling

Algorithm

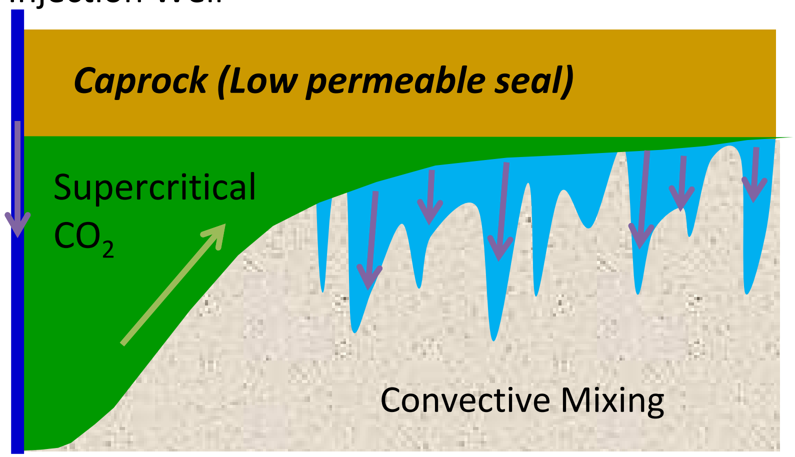
Software

Hardware

Simulation of Geologic CO_2 Storage

- Science
  - Behavior of CO_2 in supercritical state at deep reservoir
- PDE's
  - 3D Multiphase Flow (Liquid/Gas) + 3D Mass Transfer
- Method for Computation
  - TOUGH2 code based on FVM, 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

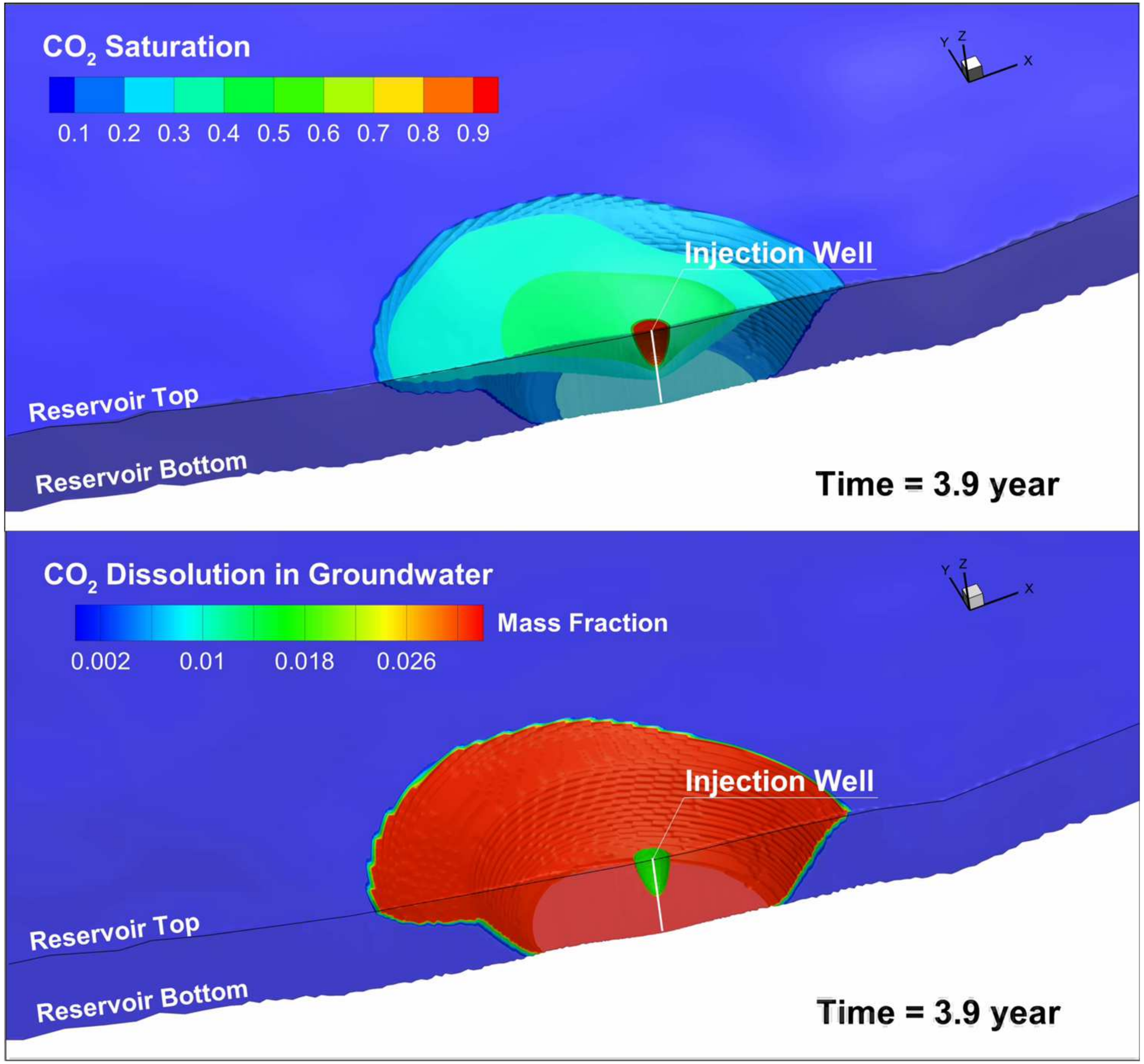


- **Buoyant scCO_2 overrides onto groundwater**
- **Dissolution of CO_2 increases water density**
- **Denser fluid laid on lighter fluid**
- **Rayleigh-Taylor instability invokes convective mixing of groundwater**

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

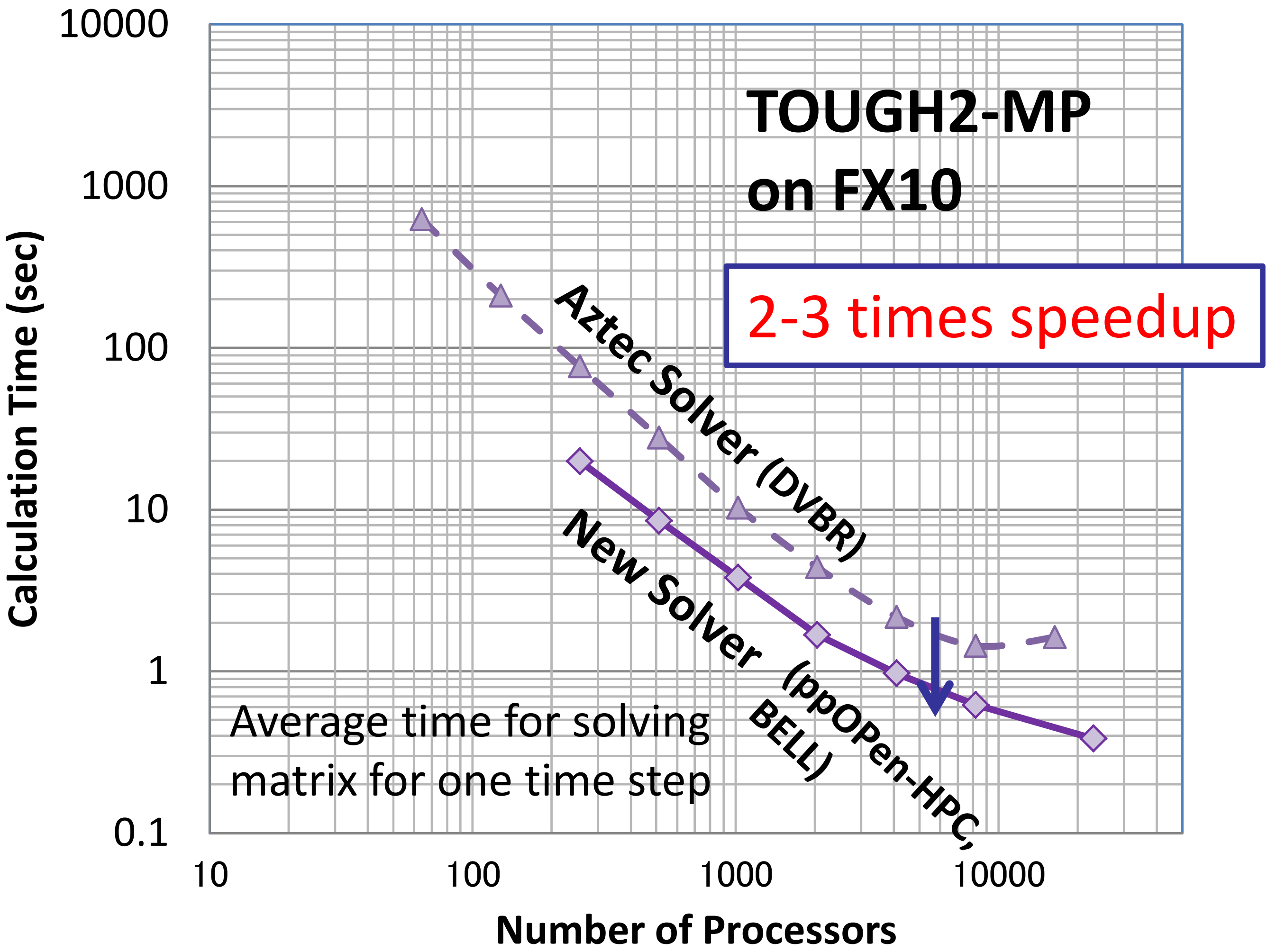
- Permeability: 100 md
- Porosity: 20%
- Pressure: 3MPa
- Temperature: 100°C
- Salinity: 15wt%

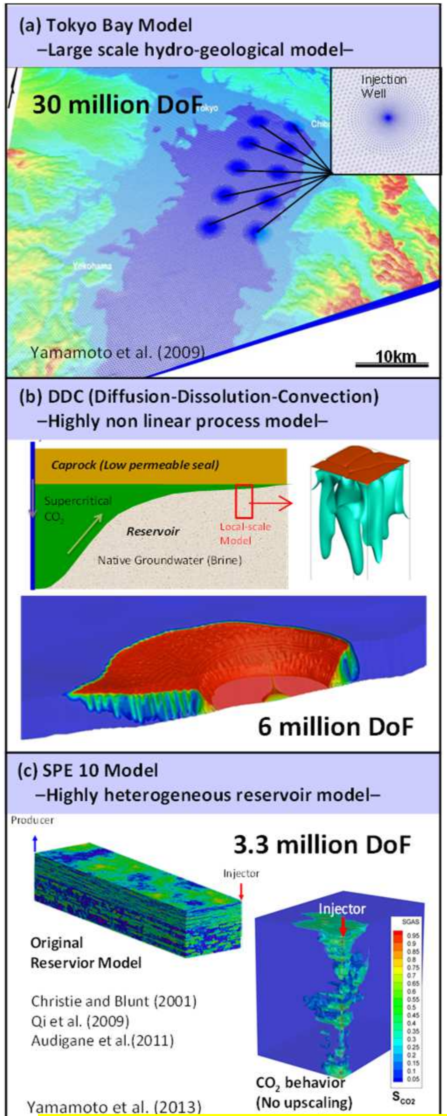
[Dr. Hajime Yamamoto, Taisei]

- 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

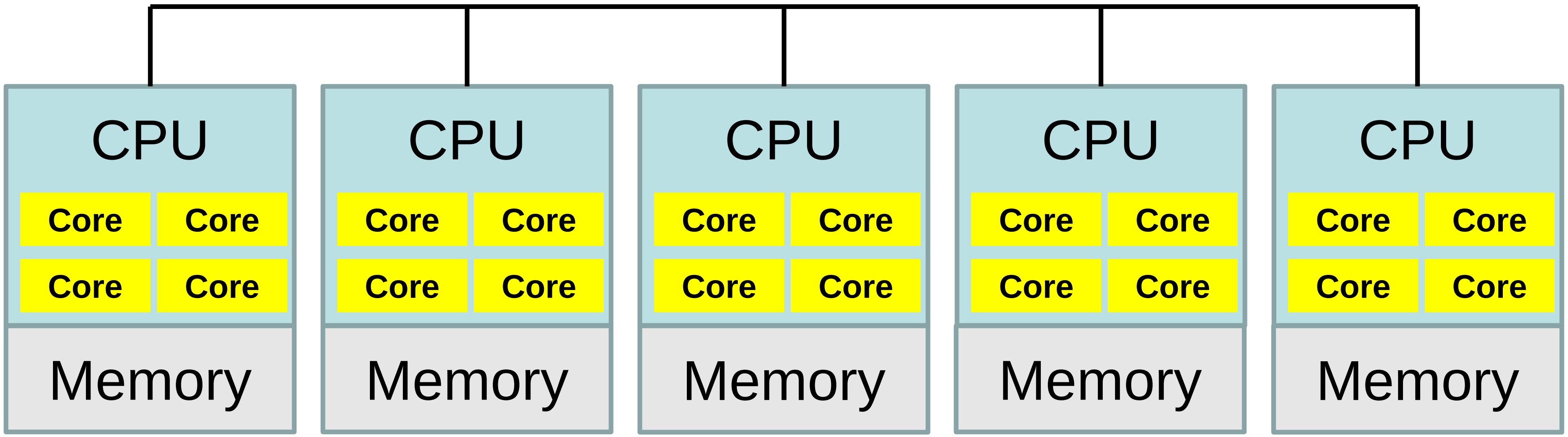
Motivation for Parallel Computing, again

- 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

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

Our Current Target: Multicore Cluster

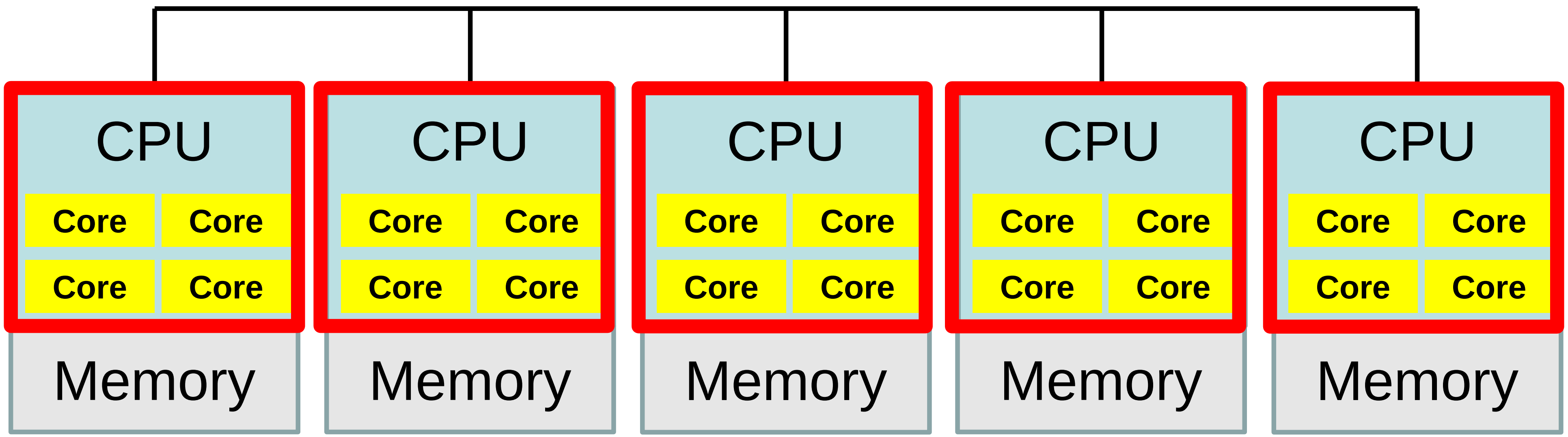
Multicore CPU's are connected through network



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

Our Current Target: Multicore Cluster

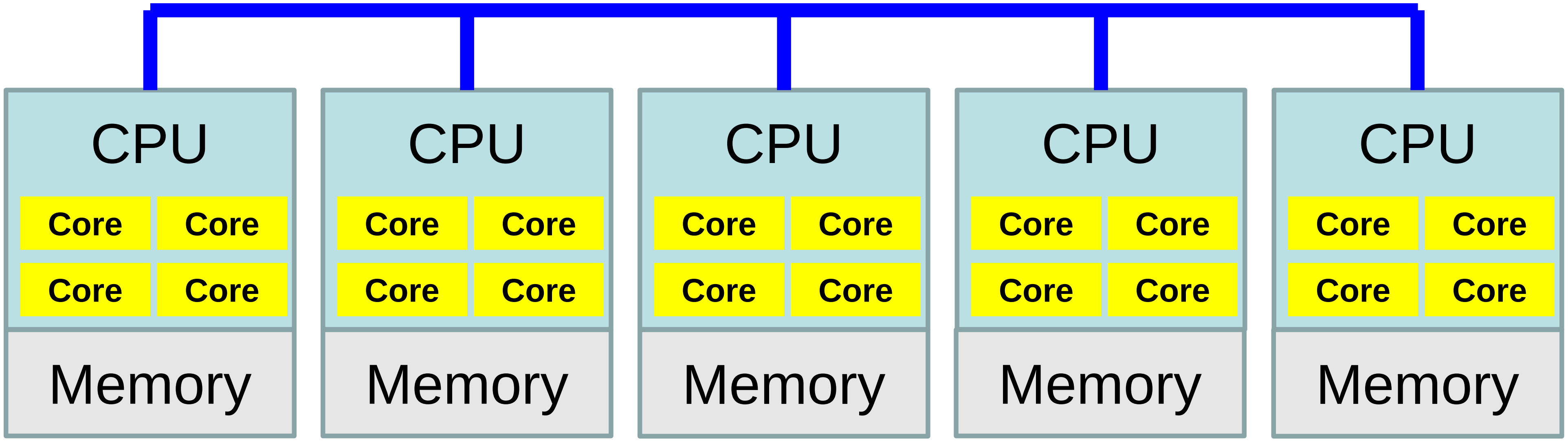
Multicore CPU's are connected through network



- 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

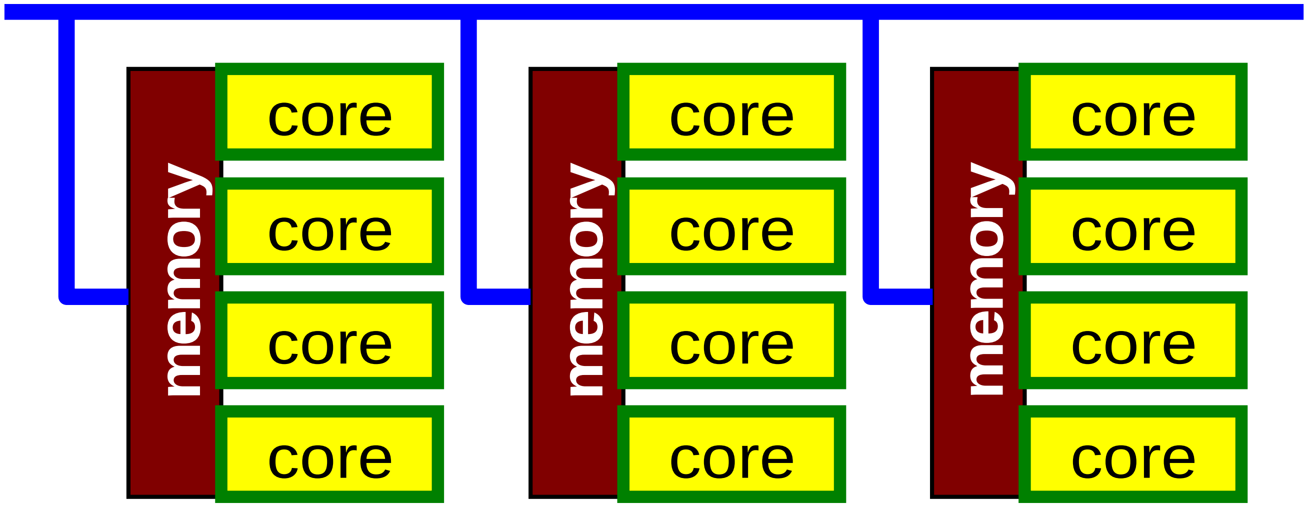


- OpenMP
  - ✓ Multithreading
  - ✓ Intra Node (Intra CPU)
  - ✓ Shared Memory
- MPI (after October)
  - ✓ Message Passing
  - ✓ Inter Node (Inter CPU)
  - ✓ Distributed Memory

Flat MPI vs. Hybrid

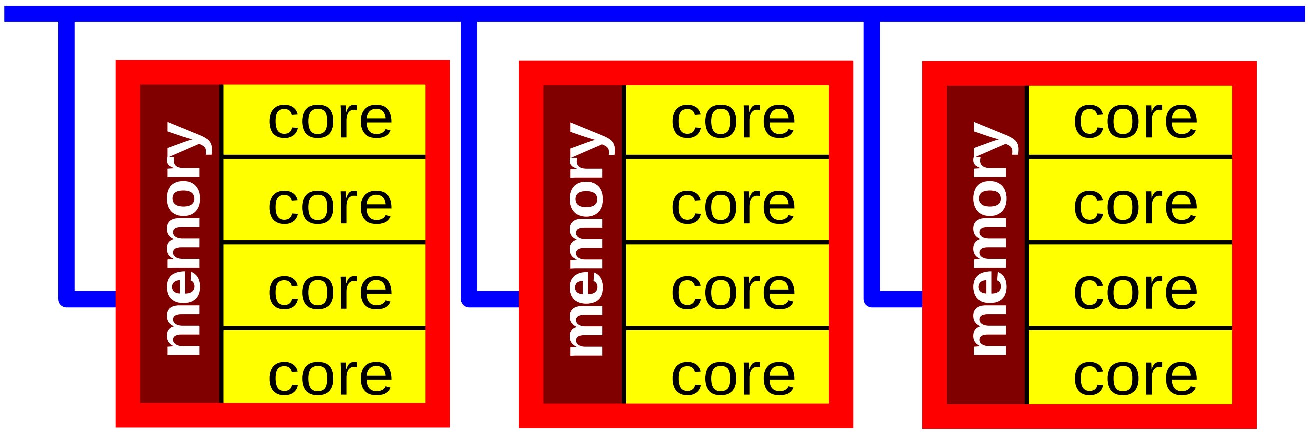
Flat-MPI：Each Core -> Independent

- MPI only
- Intra/Inter Node



Hybrid：Hierarchal Structure

- OpenMP
- MPI



Example of OpnMP/MPI Hybrid

Sending Messages to Neighboring Processes

MPI: Message Passing, OpenMP: Threading with Directives

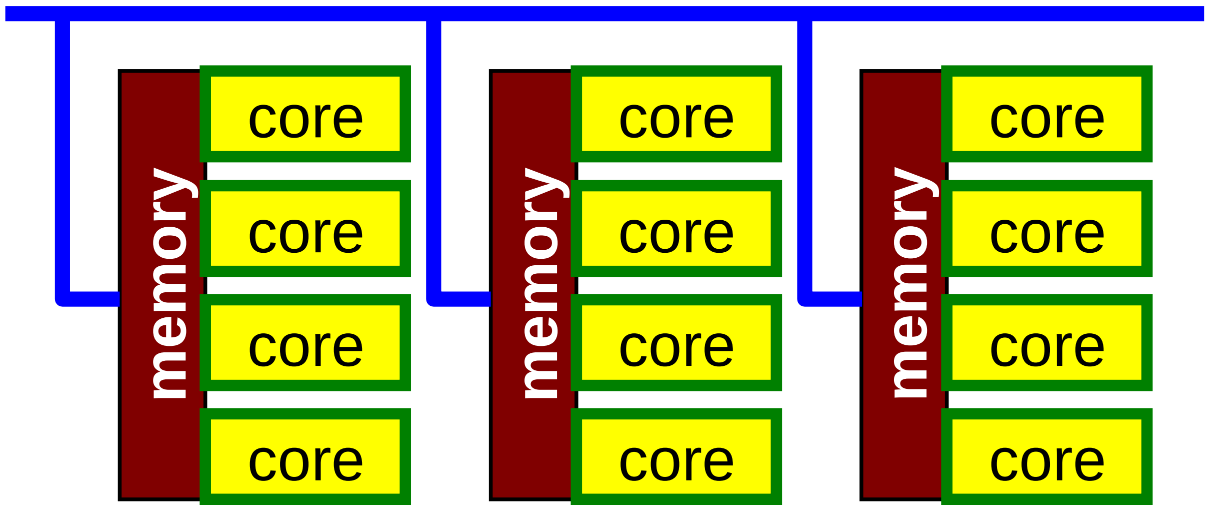
```
!C
!C- SEND

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

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

Overview of This Class (1/3)

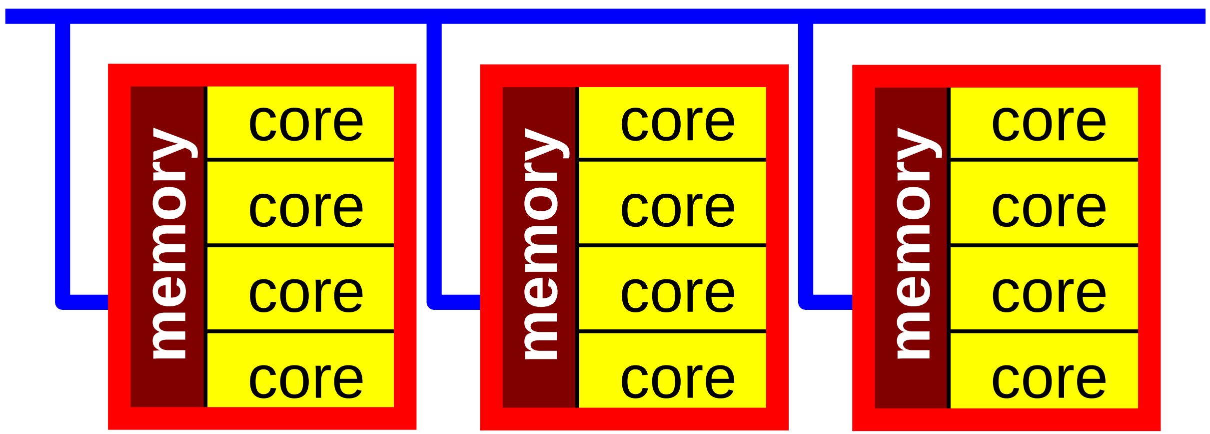
- http://nkl.cc.u-tokyo.ac.jp/21s/
- In order to make full use of modern supercomputer systems with multicore/manycore architectures, hybrid parallel programming with message-passing and multithreading is essential.
- While MPI is widely used for message-passing, OpenMP for CPU and OpenACC for GPU are 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 Oakbridge-CX (OBCX) Supercomputer with Intel Cascade Lake (CLX) at the University of Tokyo.**
  - **Because of limitation of time, we are (mainly) focusing on multithreading by OpenMP.**

Flat MPI vs. Hybrid

Flat-MPI：Each PE -> Independent



Hybrid：Hierarchal 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 **related to OpenMP** 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	ID	Title
Apr-07 (W)	CS-01a	Introduction-a
Apr-14 (W)	CS-01b	Introduction-b (Introduction-a and –b are same)
Apr-21 (W)	CS-02	FVM (1/3)
Apr-28 (W)	CS-03	FVM (2/3)
May-05 (W)	**(no class)**	**(National Holiday)**
May-12 (W)	CS-04	FVM (3/3) , OpenMP (1/3)
May-19 (W)	CS-05	OpenMP (2/3), Login to OBCX
May-26 (W)	CS-06	OpenMP (3/3)
Jun-02 (W)	CS-07	Reordering (1/3)
Jun-09 (W)	CS-08	Reordering (2/3)
Jun-16 (W)	CS-09	Reordering (3/3)
Jun-23 (W)	CS-10	Tuning
Jun-30 (W)	CS-11	Parallel Code by OpenMP (1/3)
Jul-07 (W)	CS-12	Parallel Code by OpenMP (2/3)
Jul-14 (W)	CS-13	Parallel Code by OpenMP (3/3), Q/A

“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 or 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.**
- Experiences in Programming by C/C++/Fortran
- User account of ECCS2016 must be obtained (later)
  - https://www.ecc.u-tokyo.ac.jp/en/newaccount.html

Strategy

- If you can develop programs by yourself, it is ideal... but difficult.
  - You can focus on "reading", not developing by yourself
  - Programs are in C and Fortran
    - Lectures are done by ...
- Lecture Materials
  - available at **NOON Moday** through WEB.
    - http://nkl.cc.u-tokyo.ac.jp/21s/
  - NO hardcopy is provided
- Starting at 08:30
  - You can enter the building/ZOOM classroom after 08:00
- In the Classroom …
  - Taking seats from the front row
  - Terminals must be shut-down after class

Grades

- 1 Report on programming
  - Assignment will be given in early July
  - Deadline will be 17:00, August 23 (M)

If you have any questions, please feel free to contact me !

- **NO specific office hours, appointment by e-mail**
  - Although my office is moving to Kashiwa II campus, I do not go there before October 2021
  - Zoom meeting
  - e-mail: nakajima(at)cc.u-tokyo.ac.jp

- http://nkl.cc.u-tokyo.ac.jp/21s/
- http://nkl.cc.u-tokyo.ac.jp/seminars/multicore/ 日本語資料（一部）

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

Some Technical Terms

- Processor, Core
  - Processing Unit (H/W), Processor=Core for single-core proc's
- Process
  - Unit for MPI computation, nearly equal to "core"
  - Each core (or processor) can host multiple processes (but not efficient)
- PE (Processing Element)
  - PE originally mean "processor", but it is sometimes used as "process" in this class. Moreover it means "domain" (next)
    - In multicore proc's: PE generally means "core"
- Domain
  - domain=process (=PE), each of "MD" in "SPMD", each data set
  - Domain Decomposition

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

- 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)

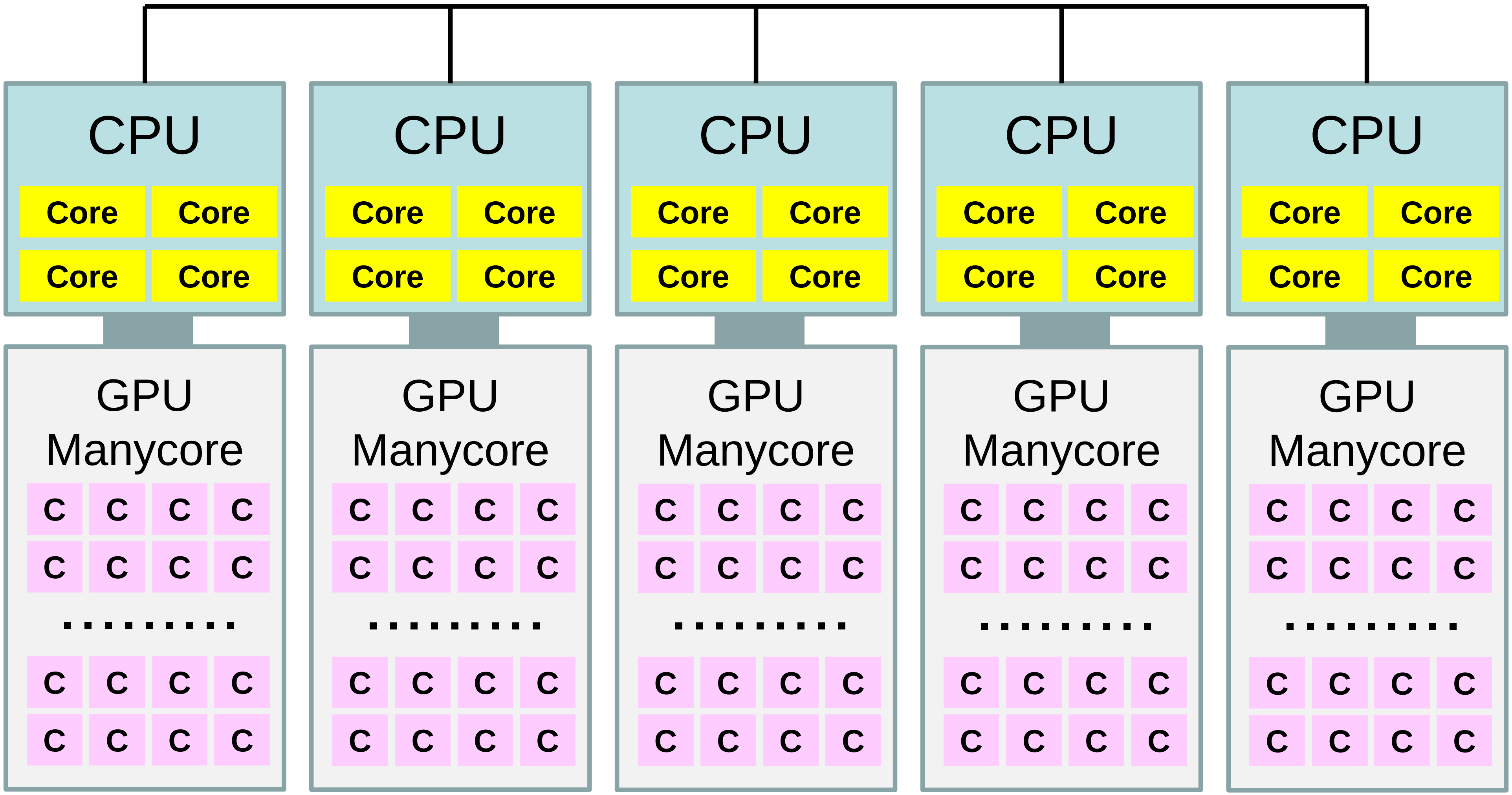
Key-Issues towards Appl./Algorithms on Exa-Scale Systems

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

60

Supercomputers with Heterogeneous/Hybrid Nodes



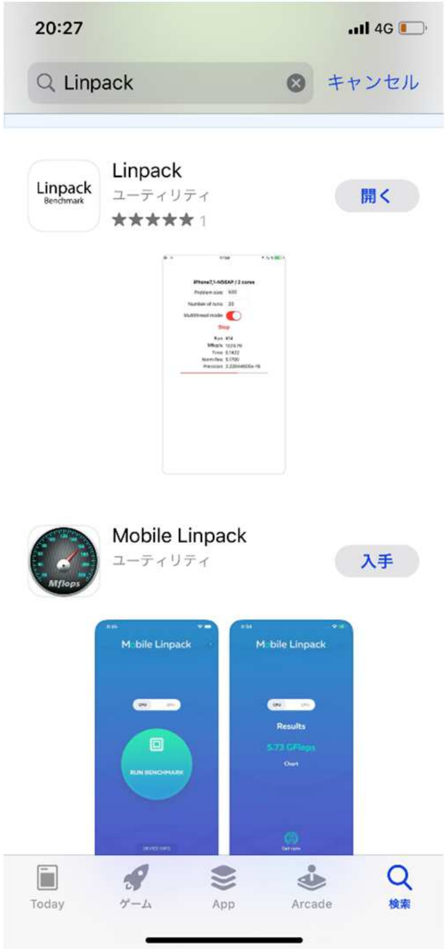
56^{th} TOP500 List (Nov, 2020) Oakbridge-CX (OBCX) is 69th

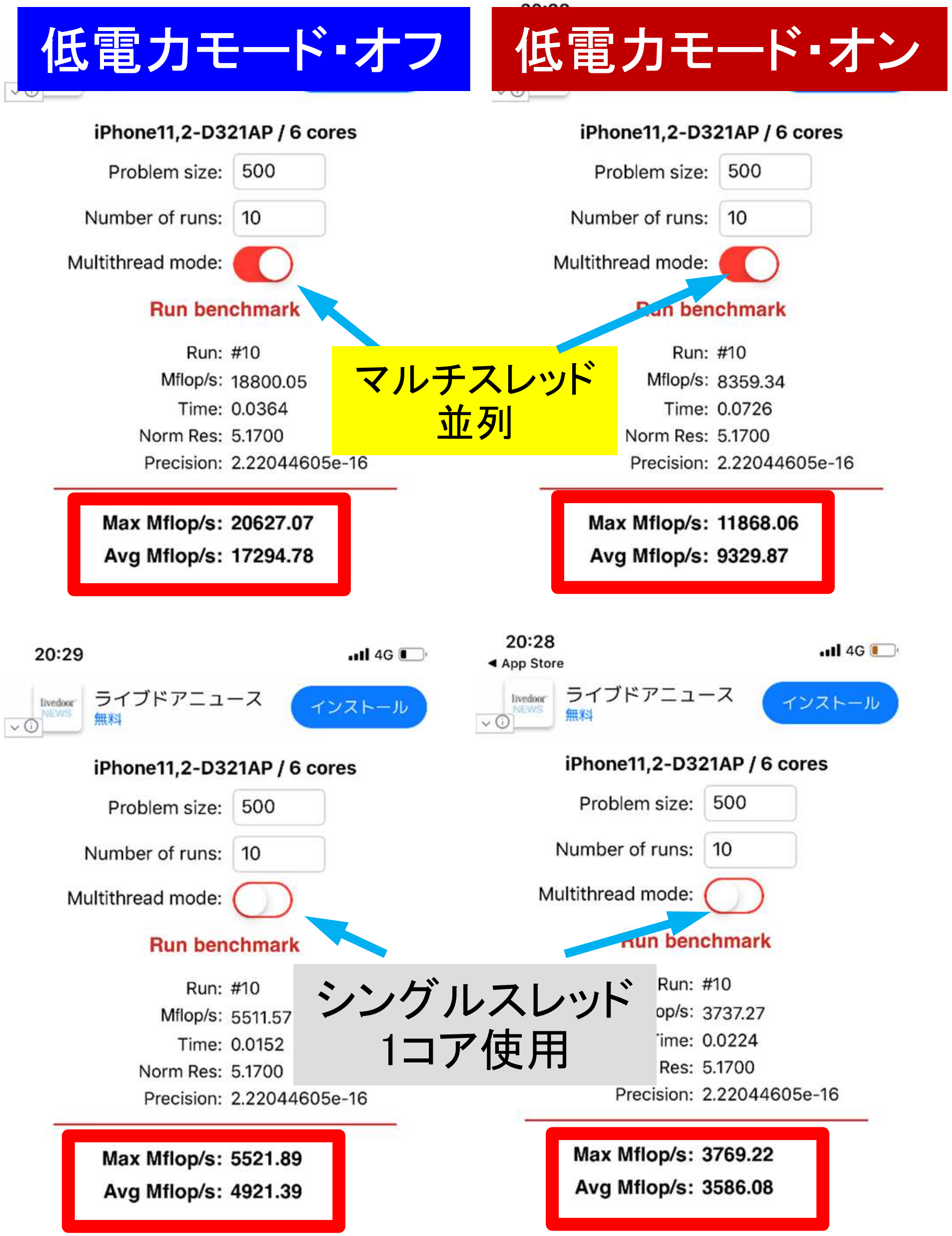
R_{max}: Performance of Linpack (TFLOPS)
R_{peak}: Peak Performance (TFLOPS),
Power: kW

http://www.top500.org/

	Site	Computer/Year Vendor	Cores	R_{max} (TFLOPS)	R_{peak} TFLOPS)	Power (kW)
1	**Supercomputer Fugaku, 2020, Japan** RIKEN Center for Computational Science (R-CCS)	Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu	7,630,848	442,010 (= 442.01 PF)	537,212	29.899
2	**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	200,795	10,096
3	**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
4	**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
5	**Selene, 2020, USA** NVIDIA Corporation	DGX A100 SuperPOD, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband	555,520	63,460	79,125	2,646
6	**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
7	**JUWELS Booster Module, 2020, Germany** Forschungszenturm Juelich (FZJ)	Bull Sequana XH2000, AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR Infiniband	277,760	27,580	34,568.6	1,344
8	**HPC5, 2020, Italy** Eni S.p.A (Ente Nazionale Idrocarburi)	PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, Dell EMC	669,760	35,450	51,720.8	2,252
9	**Frontera, 2019, USA** Texas Advanced Computing Center	Dell C6420, Xeon Platinum 8280 28c 2.7GHz, Mellanox Infiniband HDR	448,448	23,516	38,746	
10	**Dammam-7, 2020, Saudi Arabia** Saudi Aramco	Cray CS-Storm, Xeon Gold 6248 20C, 2.5GHz, NVIDIA V100 SXM2, Infiniband HDR 100, HPE	387,872	21,230	27,154	2,384
22	**Oakforest-PACS, 2016, Japan** Joint Center for Advanced High Performance Computing	Fujitsu PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path	556,104	13,556	24,913	2,719

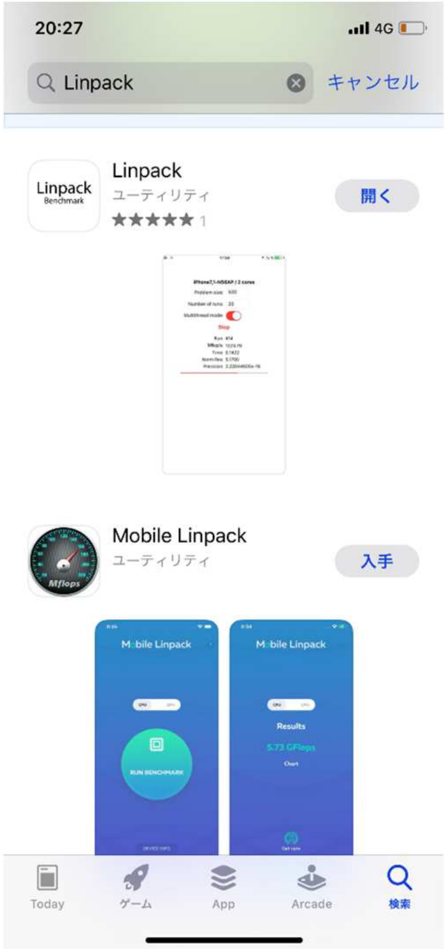
Linpack on My iPhone XS





- Performance of my iPhone XS is about 20,000 Mflops
- 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



- 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 !!