Introduction
Overview of the Class

Kengo Nakajima
Information Technology Center

Technical & Scientific Computing II (4820-1028)
Seminar on Computer Science II (4810-1205)
Parallel FEM

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

Technical & Scientific Computing II
Seminar on Computer Science II
科学技術計算II・コンピュータ科学特別講義II

- Parallel FEM
  - **Introduction to Parallel Programming by MPI**
  - Data Structure for Parallel FEM
  - How to develop parallel codes
  - Exercise on Oakleaf-FX (Fujitsu PRIMEHPC FX10
  - Parallel version of "fem3d" (3D static linear-elastic FEM code) in Summer Semester
    - Technical & Scientific Computing I, Seminar on Computer Science I

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.

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.

This Class ...

Science

Modeling

Algorithm

Software

Hardware

- Parallel FEM using MPI

- Science: 3D Solid Mechanics
- Modeling: FEM
- Algorithm: Iterative Solvers etc.

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

Road to Programming for “Parallel” Scientific Computing

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

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

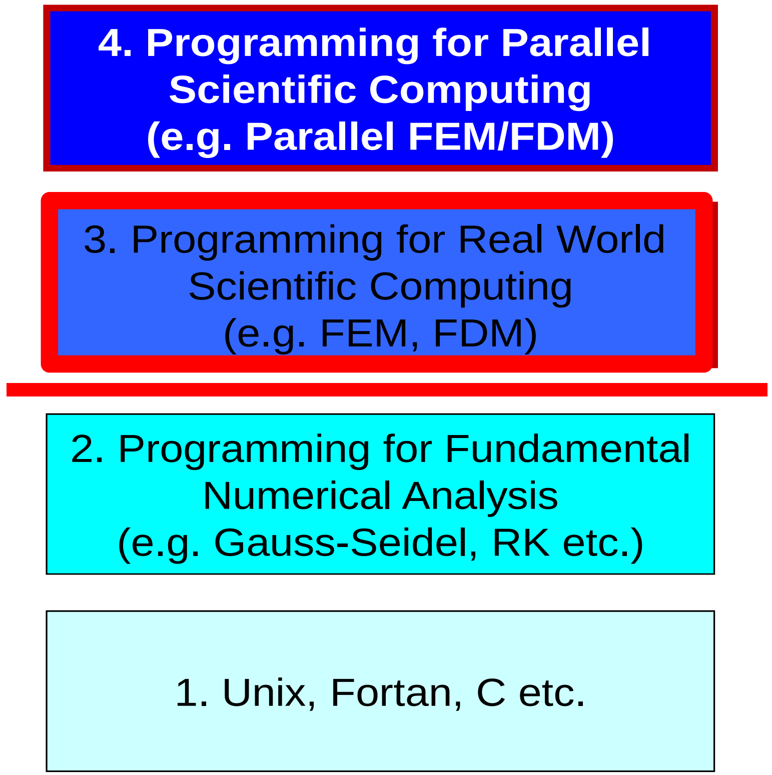
<u>Big gap here !!</u>

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



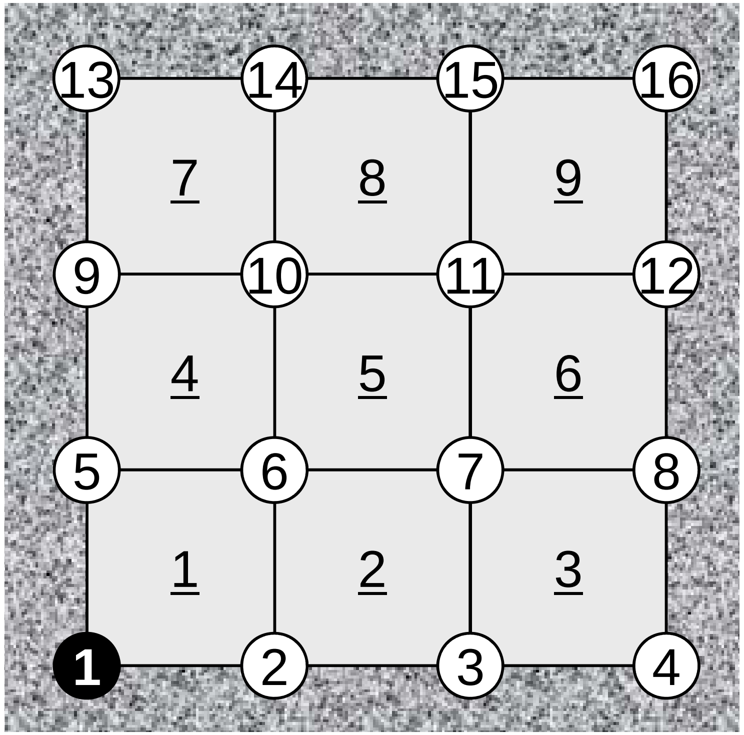
- How to understand the code ?
  - Reading the application code !!
  - It seems primitive, but very effective.
  - In this class, "reading the source code" is encouraged.
  - 3: Summer, 4: Fall/Winter Semesters

Finite-Element Method (FEM)

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

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

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



Galerkin FEM procedures

- Apply Galerkin procedures to each element:

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

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

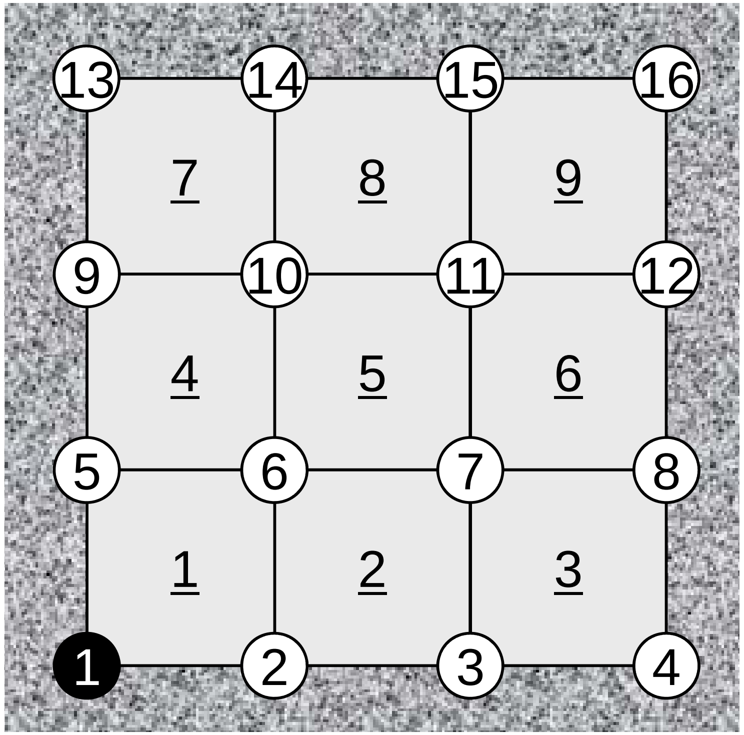
$\{\phi\}$: T at each vertex

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

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

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

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

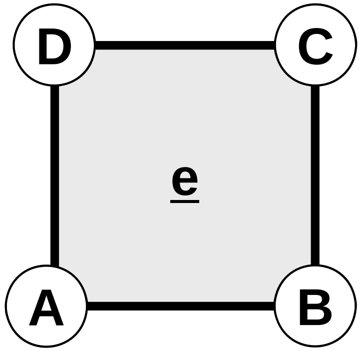


Element Matrix

- Apply the integration to each element and form "element" matrix.

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

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

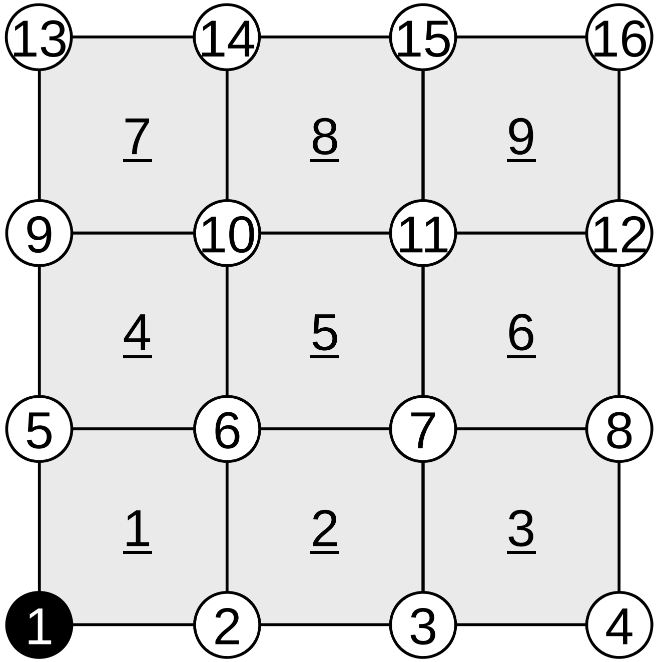


$$[k^{(e)}]\{\phi^{(e)}\}=\{f^{(e)}\}$$

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

Global (Overall) Matrix

Accumulate each element matrix to “global” matrix.

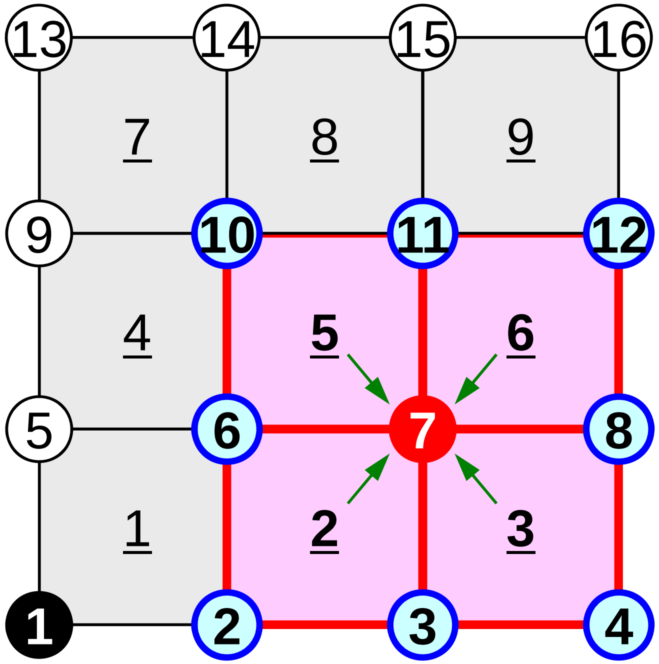


$$[K]\{\Phi\}=\{F\}$$

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

To each node …

Effect of surrounding elem's/nodes are accumulated.



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

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

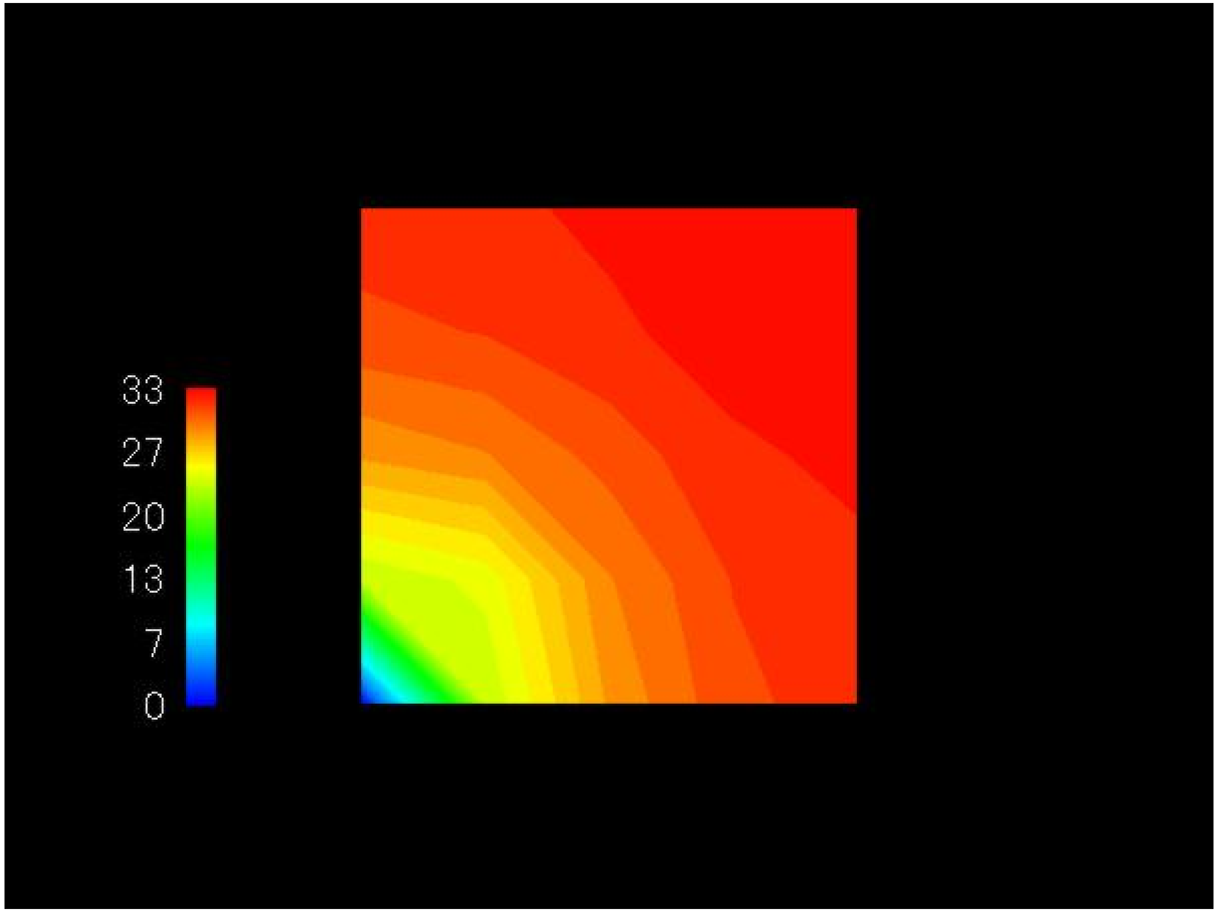
Solve the obtained global eqn's

under certain boundary conditions

(Φ_1=0 in this case)

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

Result …

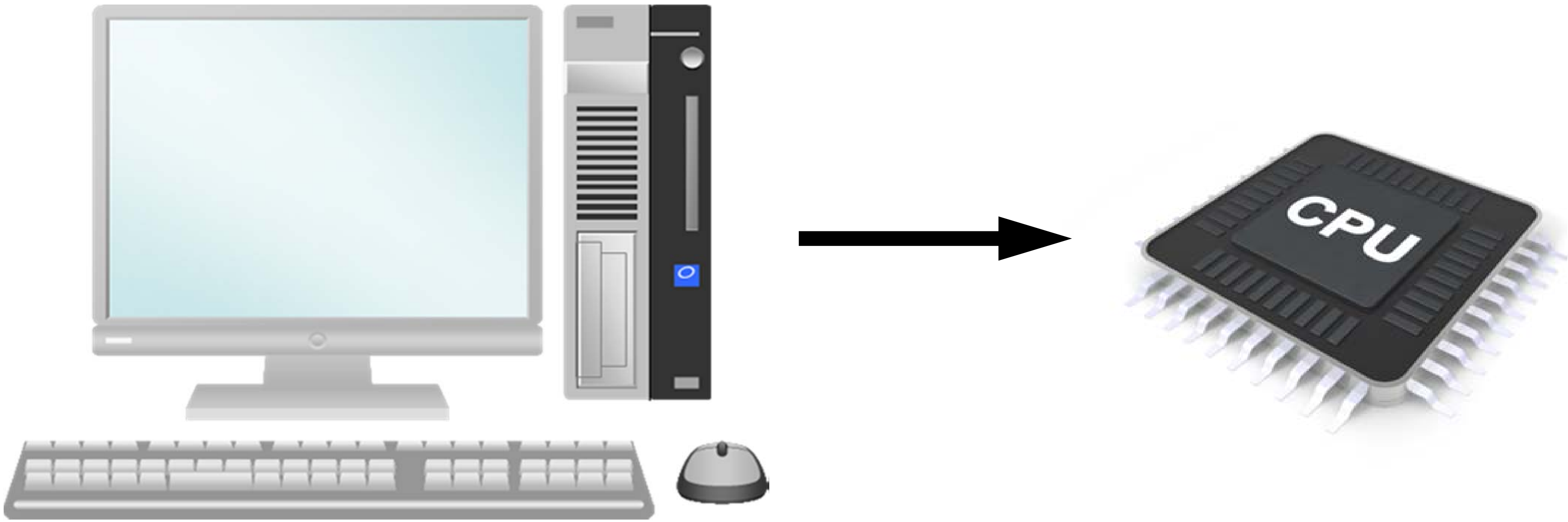


Features of FEM applications

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

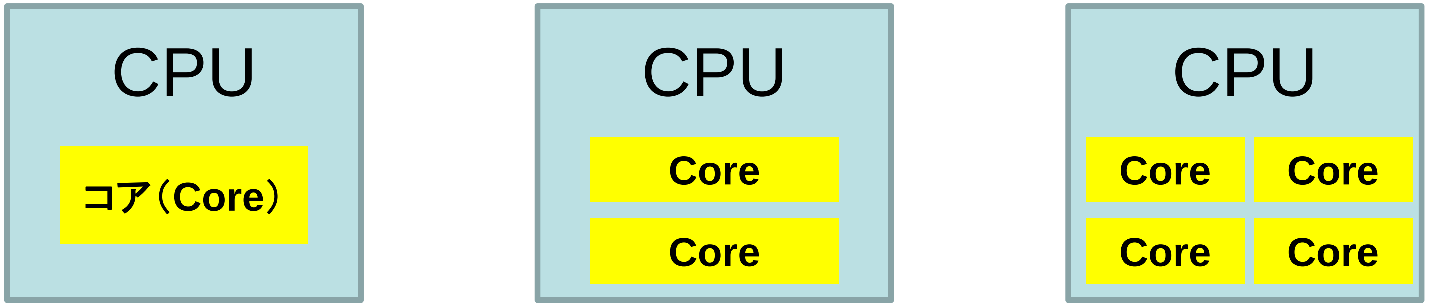
- Target: Parallel FEM
- **Supercomputers and Computational Science**
- Overview of the Class
- Future Issues

Computer & CPU



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

Multicore CPU

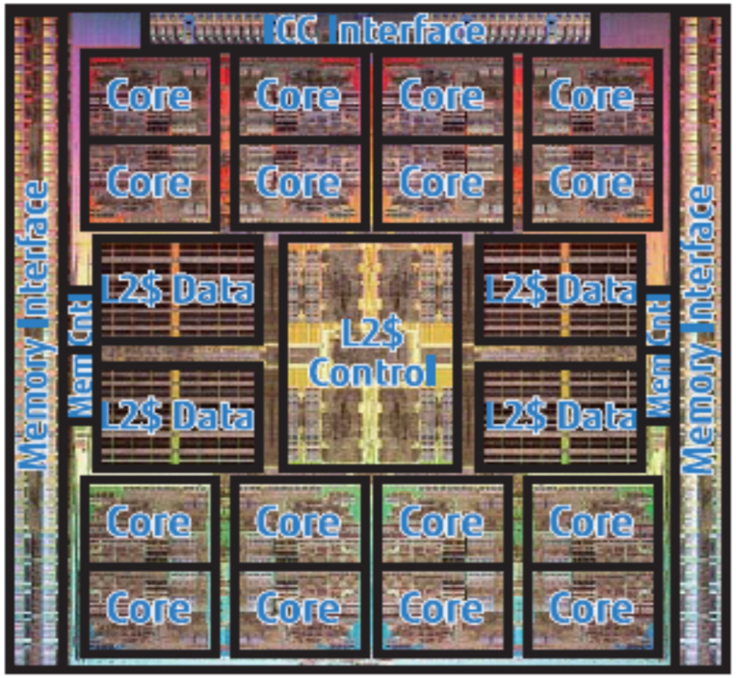


Single Core
1 cores/CPU

Dual Core
2 cores/CPU

Quad Core
4 cores/CPU

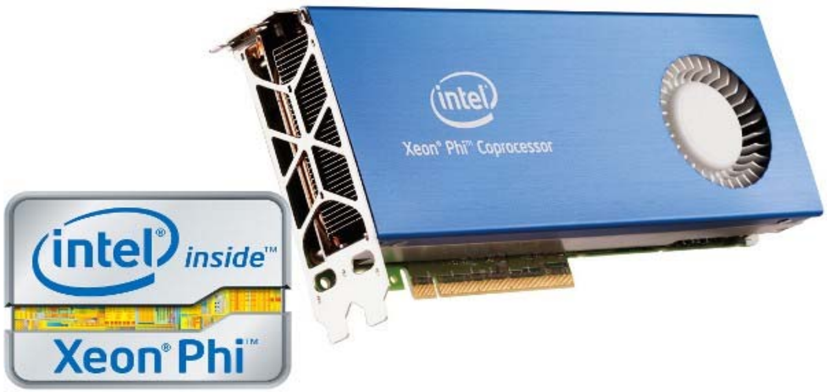
- Core= Central part of CPU
- Multicore CPU's with 4-8 cores are popular



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

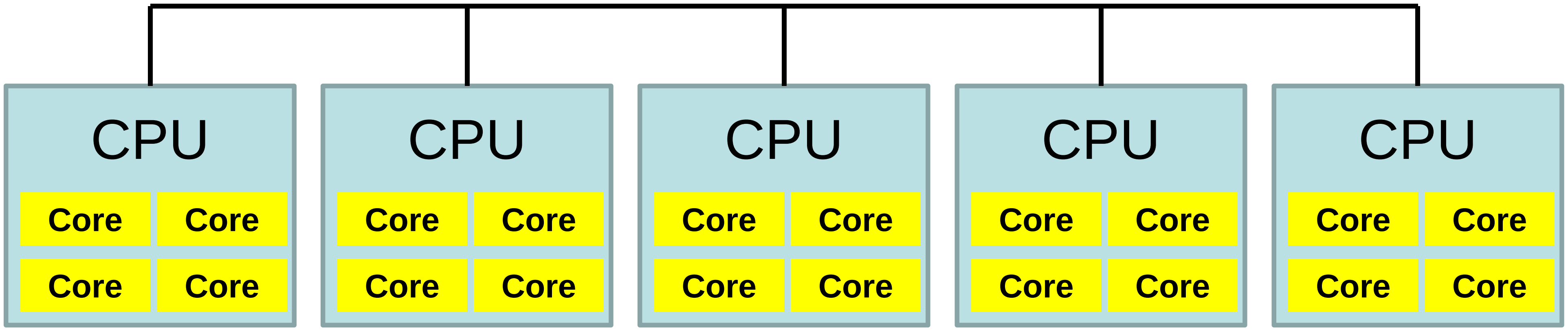
GPU/Manycores

- GPU : Graphic Processing Unit
  - GPGPU: General Purpose GPU
  - $O(10^2)$ cores
  - High Memory Bandwidth
  - Cheap
  - NO stand-alone operations
    - Host CPU needed
  - Programming: CUDA, OpenACC
- Intel Xeon/Phi: Manycore CPU
  - 60 cores
  - High Memory Bandwidth
  - Unix, Fortran, C compiler
  - Currently, host CPU needed
    - Stand-alone will be possible soon

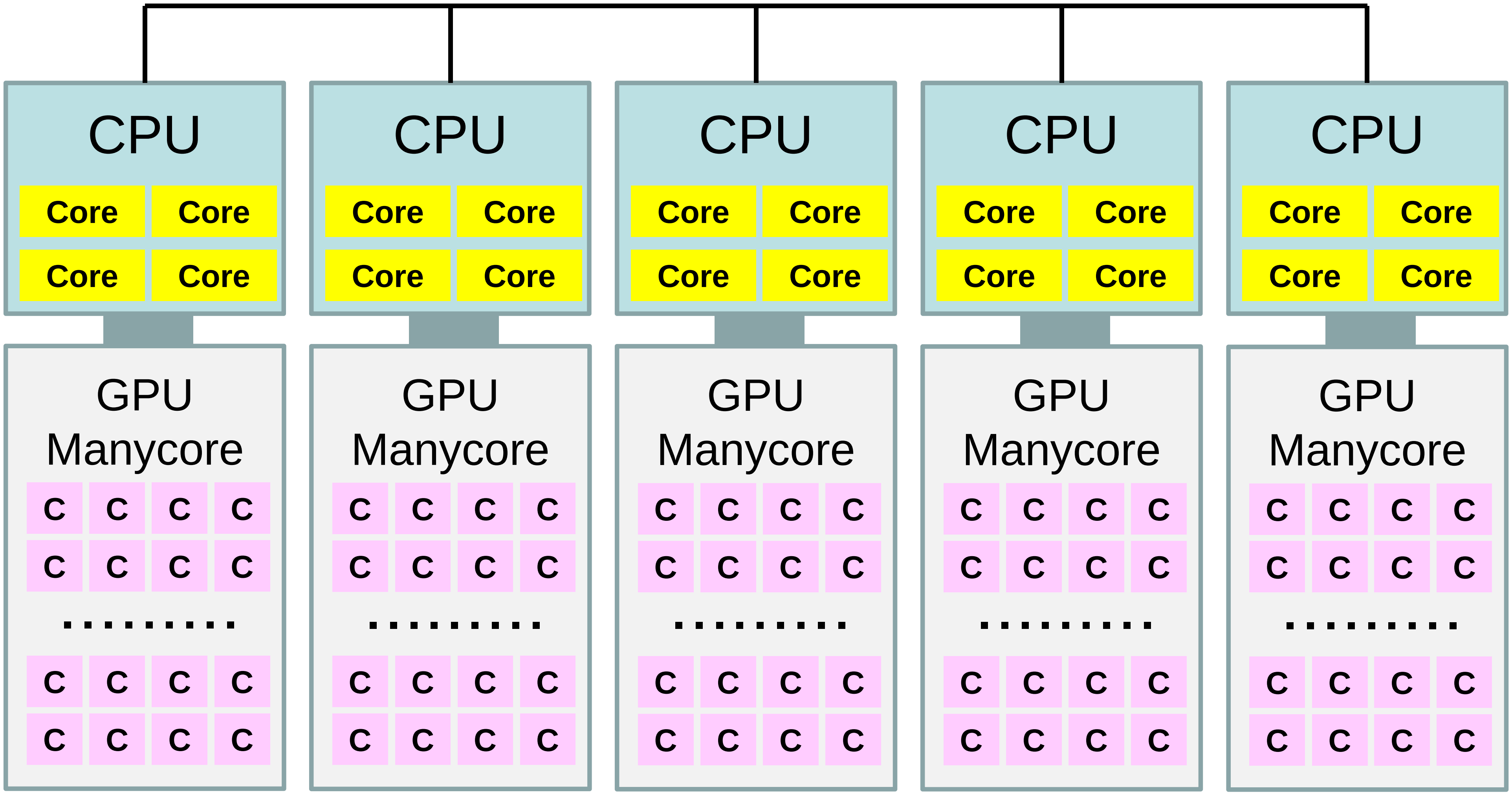


Parallel Supercomputers

Multicore CPU’s are connected through network



Supercomputers with Heterogeneous/Hybrid Nodes



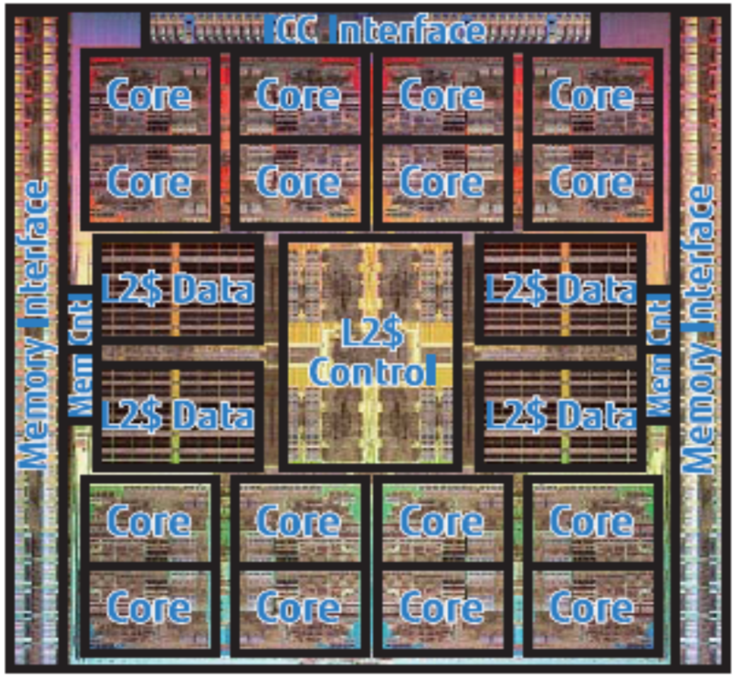
Performance of Supercomputers

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

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

Peak Performance of Oakleaf-FX

Fujitsu PRIMEHPC FX10 at U.Tokyo

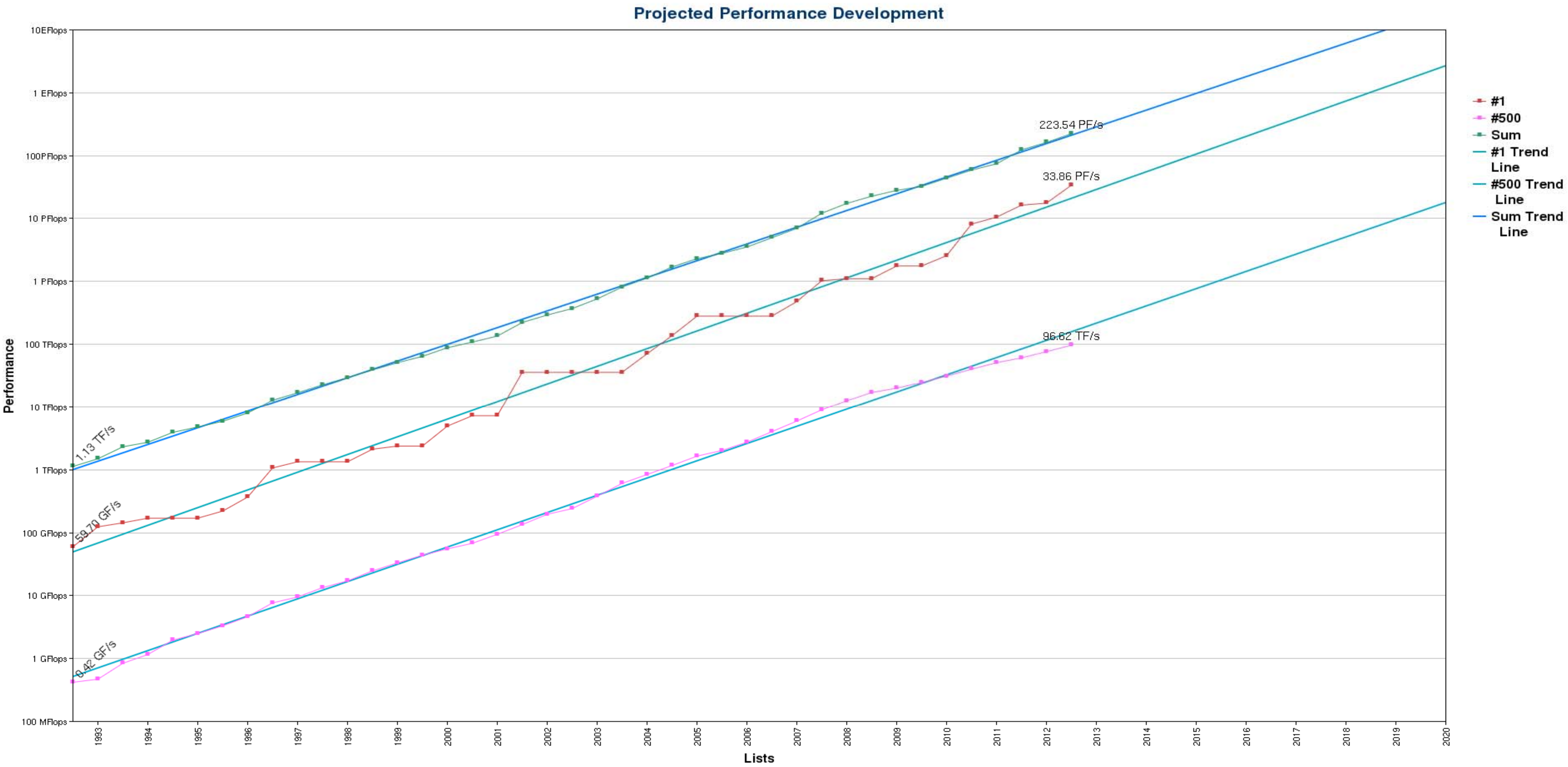


- 1.848 GHz
- 8 FLOP operations per Clock
- Peak Performance (1 core)
  - 1.848×8= 14.78 GFLOPS
- Peak Performance (1 node/16 cores)
  - 236.5 GFLOPS
- Peak Performance of Entire Performance
  - 4,800 nodes, 76,800 cores
  - 1.13 PFLOPS

TOP 500 List

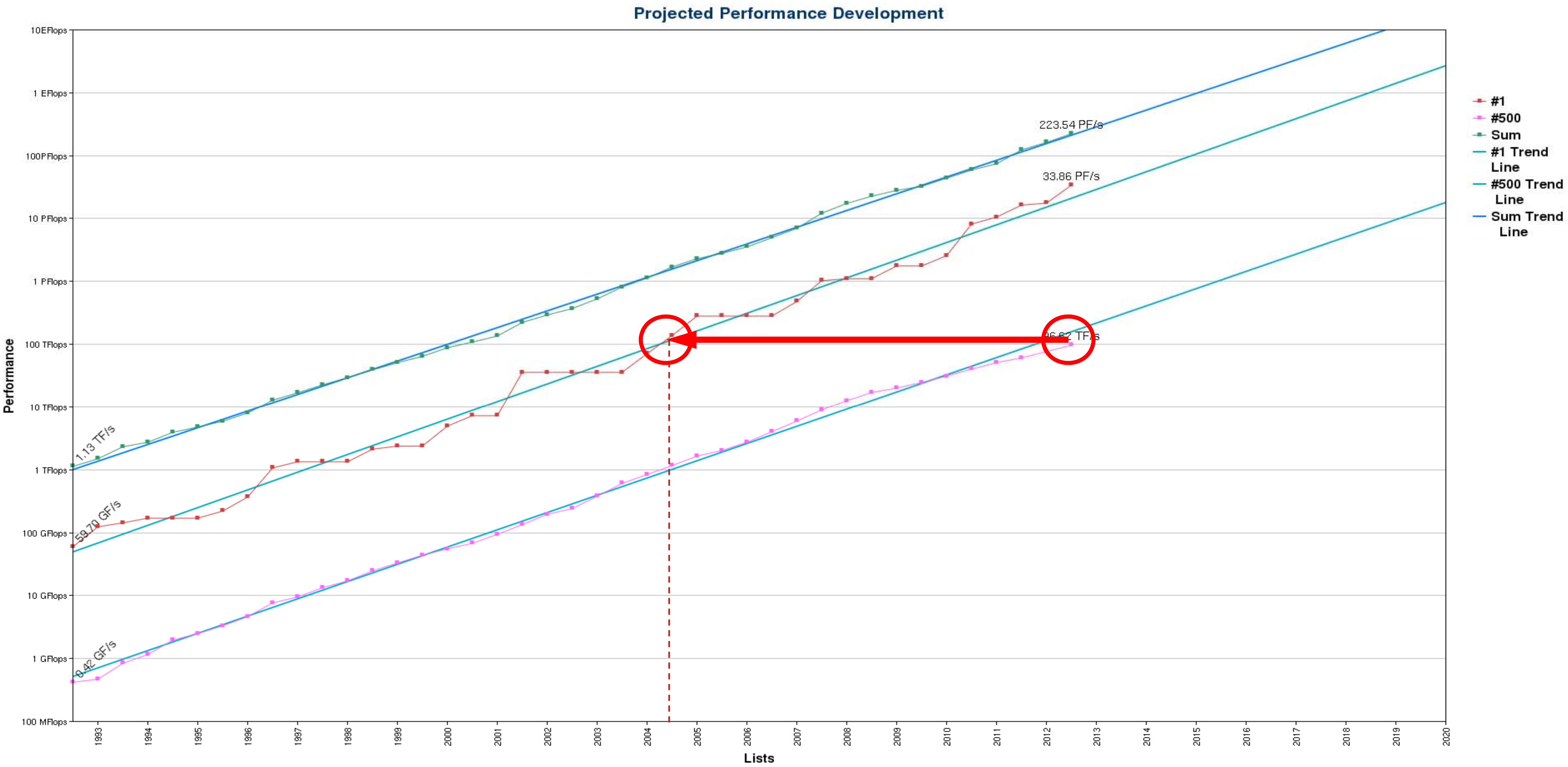
http://www.top500.org/

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



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

http://www.top500.org/



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

http://www.top500.org/

41st TOP500 List (June, 2013)

	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	National Supercomputing Center in Tianjin, China	**Tianhe-2** Intel Xeon E5-2692, TH Express-2, IXeon Phi2013 NUDT	3120000	33863 (= 33.9 PF)	54902	17808
2	Oak Ridge National Laboratory, USA	**Titan** Cray XK7/NVIDIA K20x, 2012 Cray	560640	17590	27113	8209
3	Lawrence Livermore National Laboratory, USA	**Sequoia** BlueGene/Q, 2011 IBM	1572864	17173	20133	7890
4	**RIKEN AICS, Japan**	**K computer, SPARC64 VIIIfx , 2011 Fujitsu**	**705024**	**10510**	**11280**	**12660**
5	Argonne National Laboratory, USA	**Mira** BlueGene/Q, 2012 IBM	786432	85867	10066	3945
6	TACC, USA	**Stampede** Xeon E5-2680/Xeon Phi, 2012 Dell	462462	5168	8520	4510
7	Forschungszentrum Juelich (FZJ), Germany	**JuQUEEN** BlueGene/Q, 2012 IBM	458752	5009	5872	2301
8	DOE/NNSA/LLNL, USA	**Vulcan** BlueGene/Q, 2012 IBM	393216	4293	5033	1972
9	Leibniz Rechenzentrum, Germeny	**SuperMUC** iDataPlex/Xeon E5-2680 2012 IBM	147456	2897	3185	3423
10	National Supercomputing Center in Tianjin, China	**Tianhe-1A** Heterogeneous Node 2010 NUDT	186368	2566	4701	4040

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

http://www.top500.org/

41^{st} TOP500 List (June, 2013)

	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	National Supercomputing Center in Tianjin, China	**Tianhe-2** Intel Xeon E5-2692, TH Express-2, IXeon Phi2013 NUDT	3120000	33863 (= 33.9 PF)	54902	17808
2	Oak Ridge National Laboratory, USA	**Titan** Cray XK7/NVIDIA K20x, 2012 Cray	560640	17590	27113	8209
3	Lawrence Livermore National Laboratory, USA	**Sequoia** BlueGene/Q, 2011 IBM	1572864	17173	20133	7890
4	**RIKEN AICS, Japan**	**K computer, SPARC64 VIIIfx , 2011 Fujitsu**	**705024**	**10510**	**11280**	**12660**
5	Argonne National Laboratory, USA	**Mira** BlueGene/Q, 2012 IBM	786432	85867	10066	3945
6	TACC, USA	**Stampede** Xeon E5-2680/Xeon Phi, 2012 Dell	462462	5168	8520	4510
7	Forschungszentrum Juelich (FZJ), Germany	**JuQUEEN** BlueGene/Q, 2012 IBM	458752	5009	5872	2301
8	DOE/NNSA/LLNL, USA	**Vulcan** BlueGene/Q, 2012 IBM	393216	4293	5033	1972
9	Leibniz Rechenzentrum, Germeny	**SuperMUC** iDataPlex/Xeon E5-2680 2012 IBM	147456	2897	3185	3423
10	National Supercomputing Center in Tianjin, China	**Tianhe-1A** Heterogeneous Node 2010 NUDT	186368	2566	4701	4040
26	**ITC/U. Tokyo Japan**	**Oakleaf-FX SPARC64 IXfx, 2012 Fujitsu**	**76800**	**1043**	**1135**	**1177**

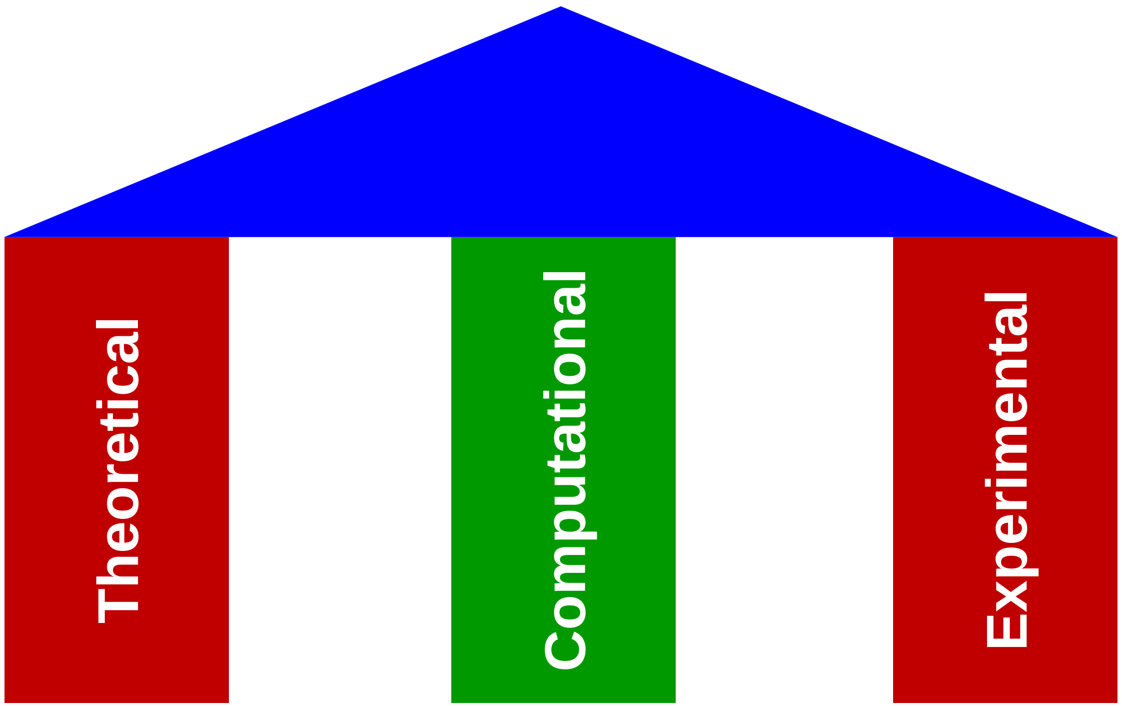
R_{max}: Performance of Linpack (TFLOPS)

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

http://www.top500.org/

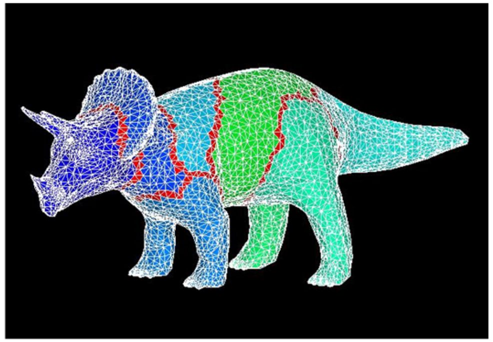
Computational Science
The 3rd Pillar of Science

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

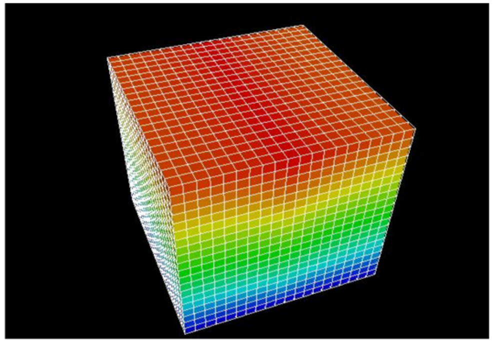


Methods for Scientific Computing

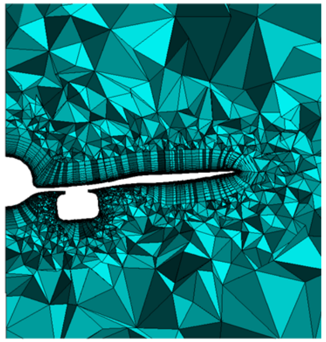
- Numerical solutions of PDE (Partial Diff. Equations)
- Grids, Meshes, Particles
  - Large-Scale Linear Equations
  - Finer meshes provide more accurate solutions



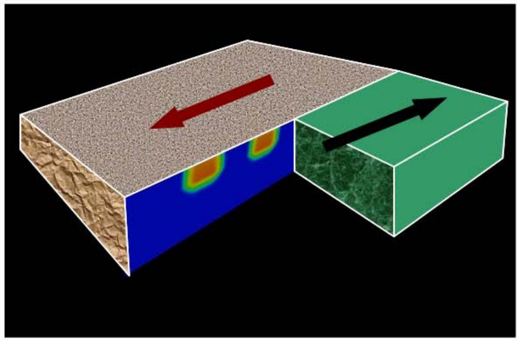
有限要素法
Finite Element Method
FEM



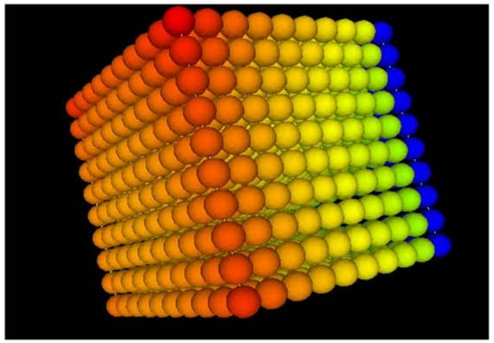
差分法
Finite Difference Method
FDM



有限体積法
Finite Volume Method
FVM



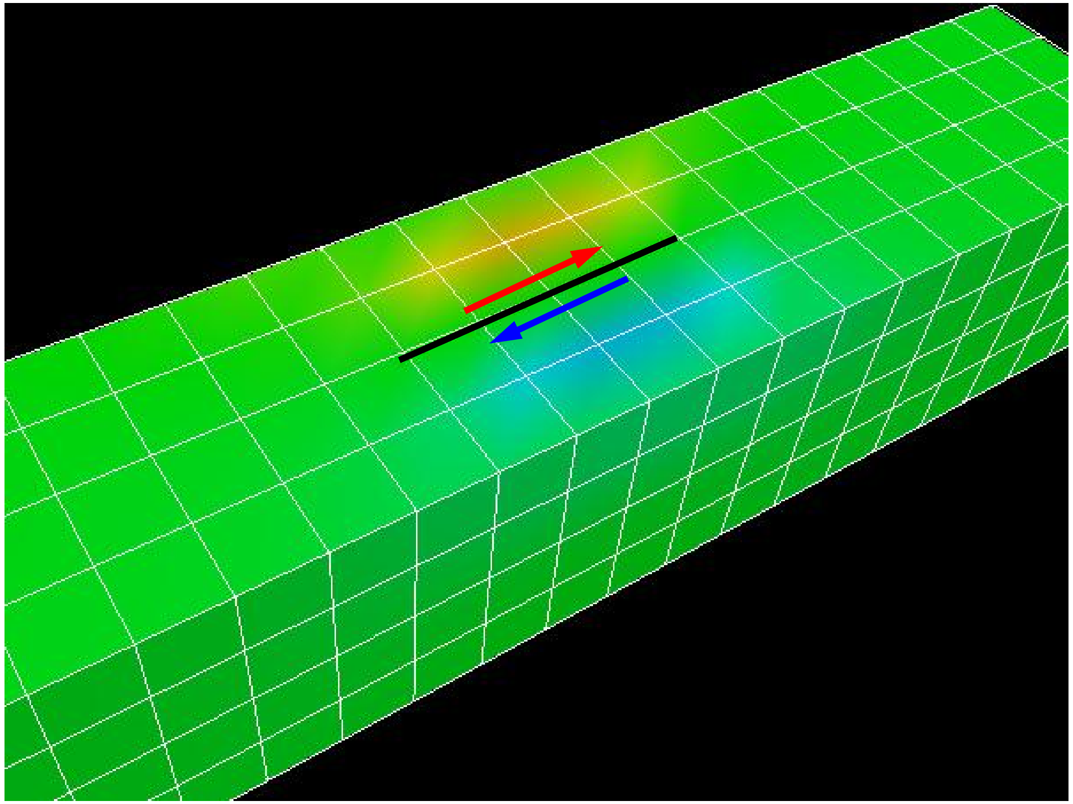
境界要素法
Boundary Element Method
BEM

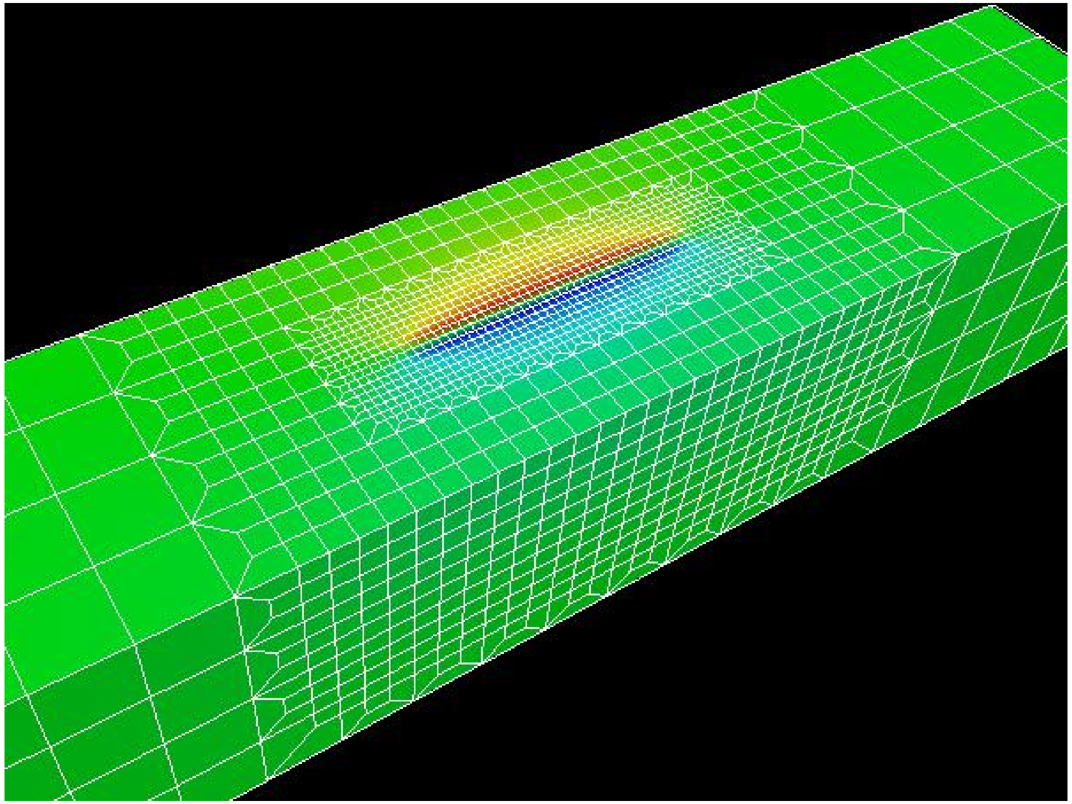


個別要素法
Discrete Element Method
DEM

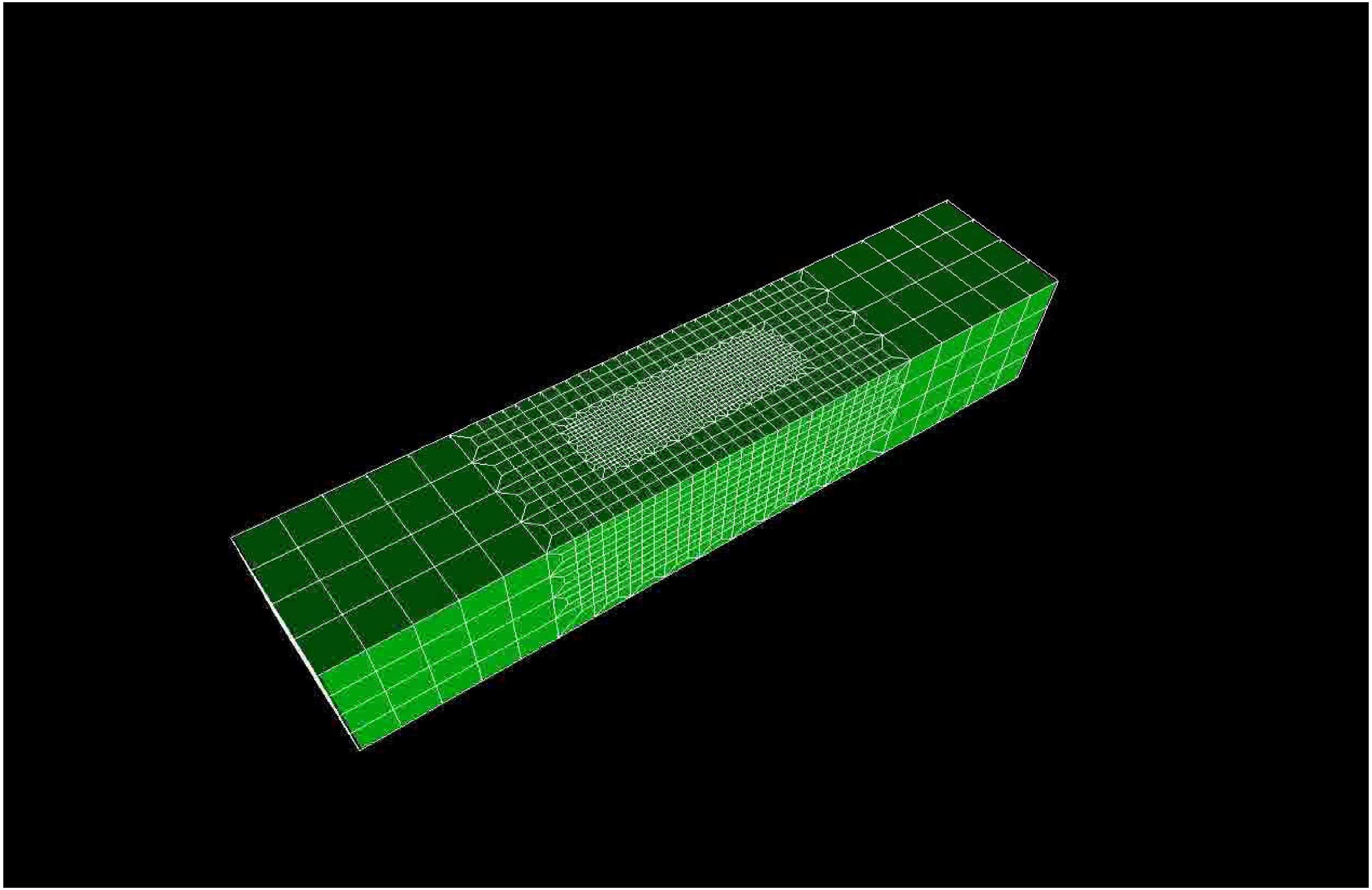
3D Simulations for Earthquake Generation Cycle San Andreas Faults, CA, USA

Stress Accumulation at Transcurrent Plate Boundaries

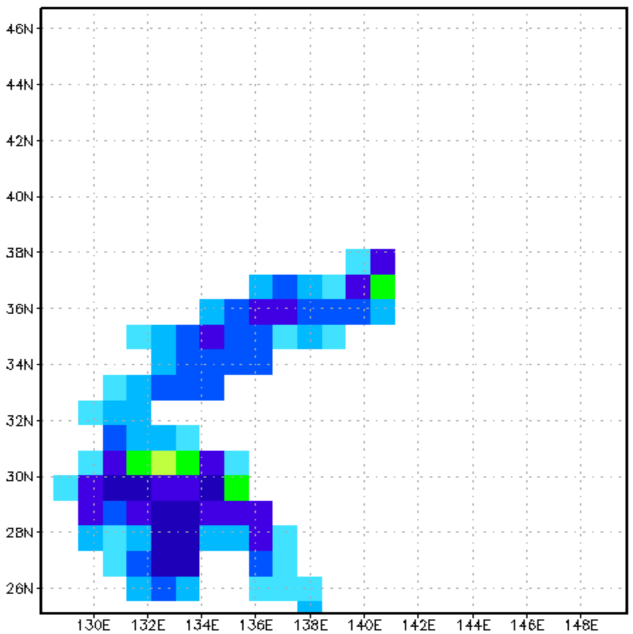




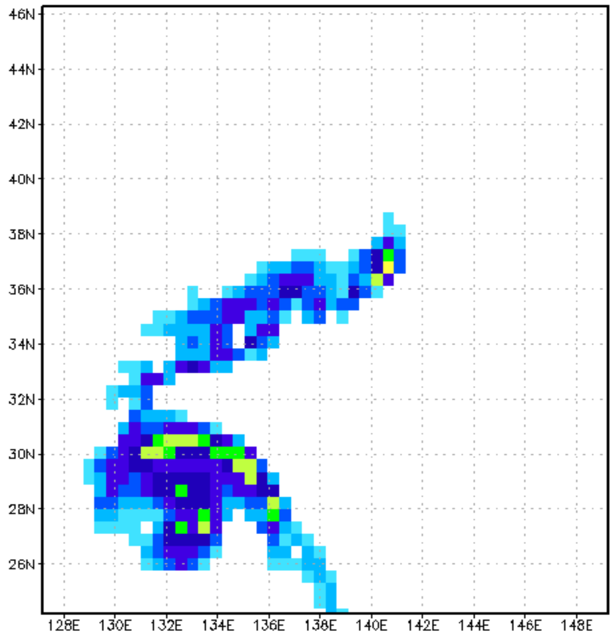
Adaptive FEM: High-resolution needed at meshes with large deformation (large accumulation)



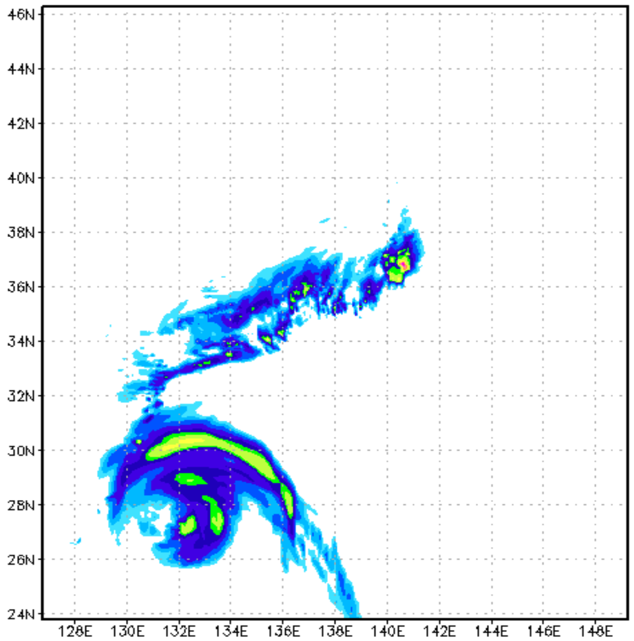
Typhoon Simulations by FDM
Effect of Resolution



Δh=100km



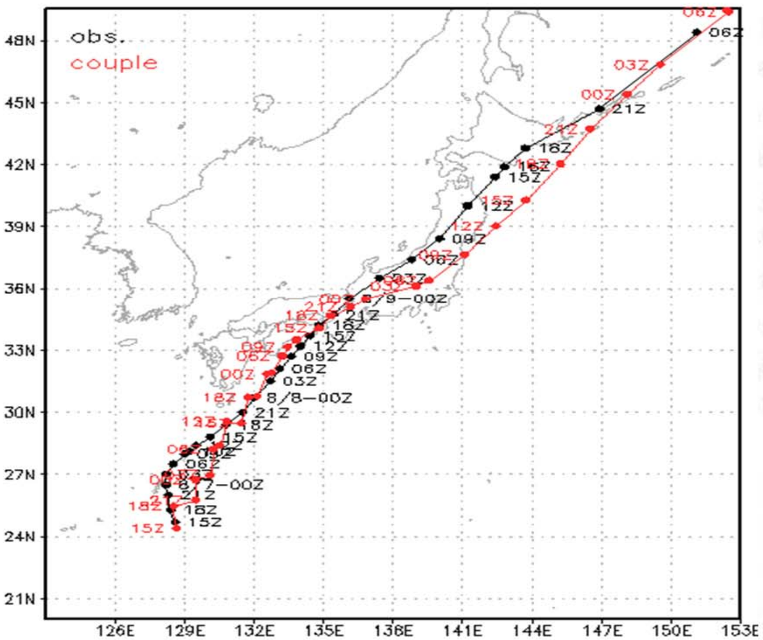
Δh=50km

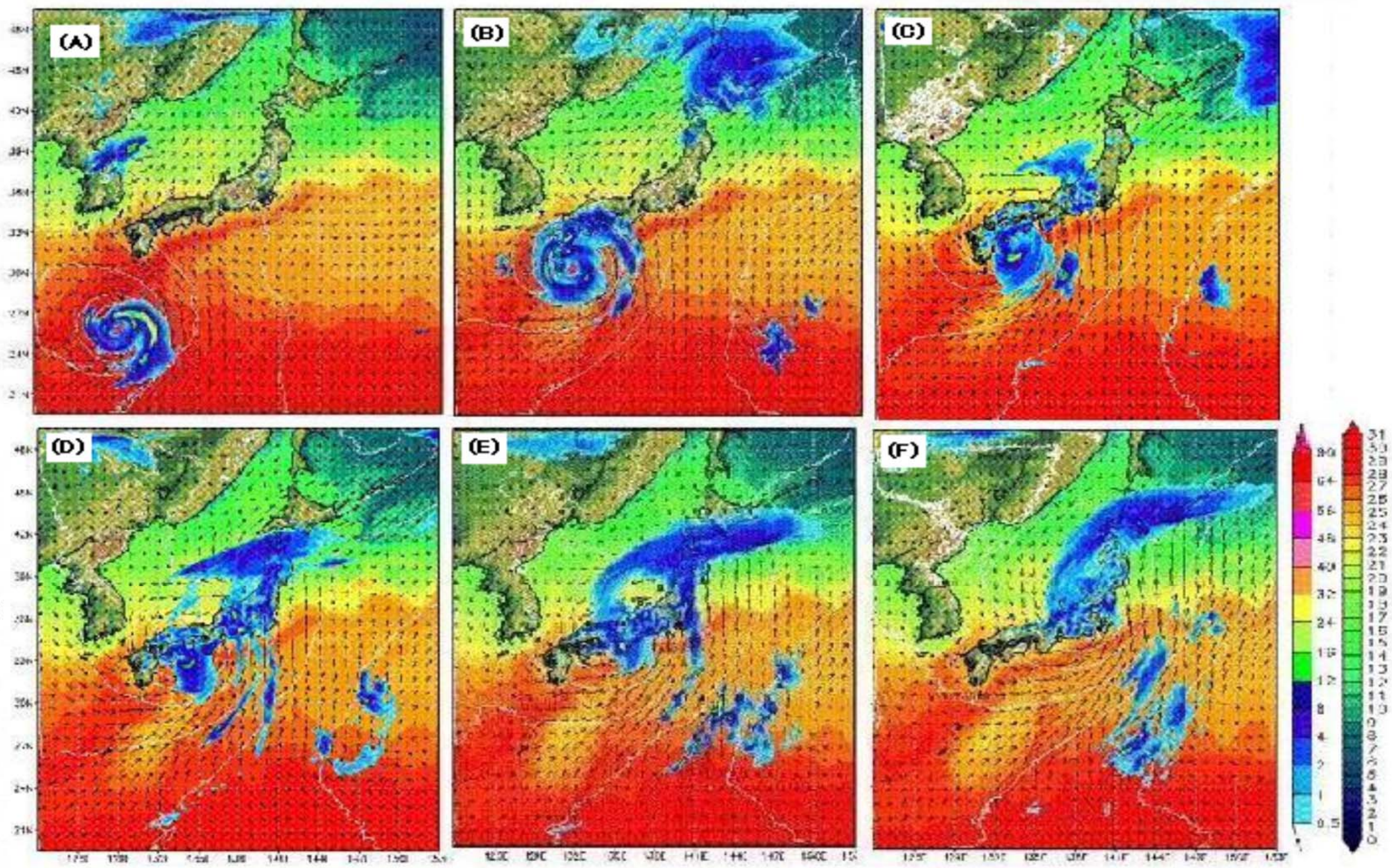


Δh=5km

[JAMSTEC]

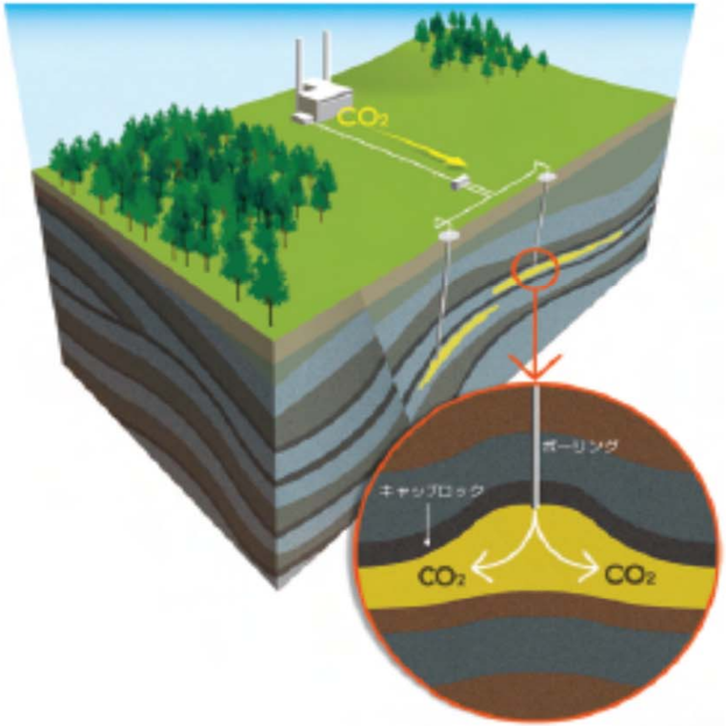
Simulation of Typhoon MANGKHUT in 2003 using the Earth Simulator





[JAMSTEC]

Simulation of Geologic CO_2 Storage



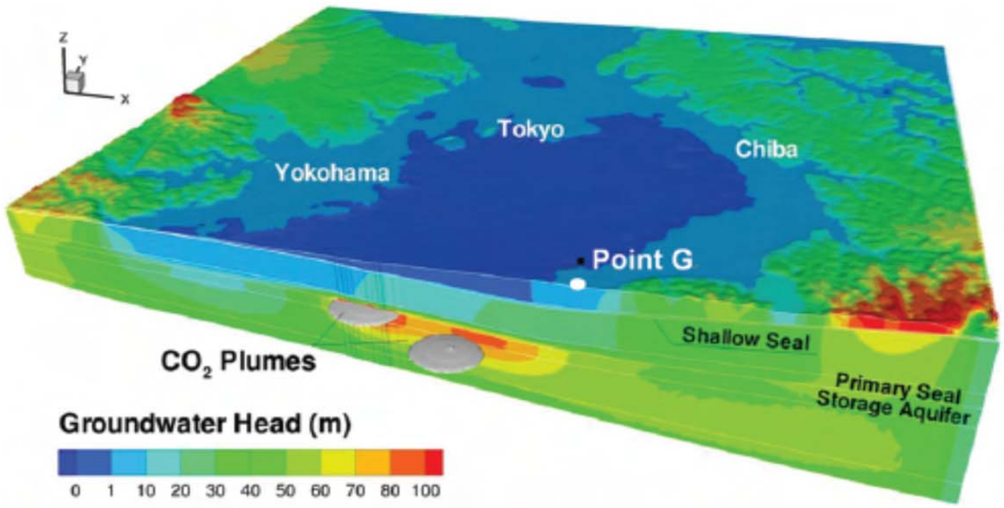
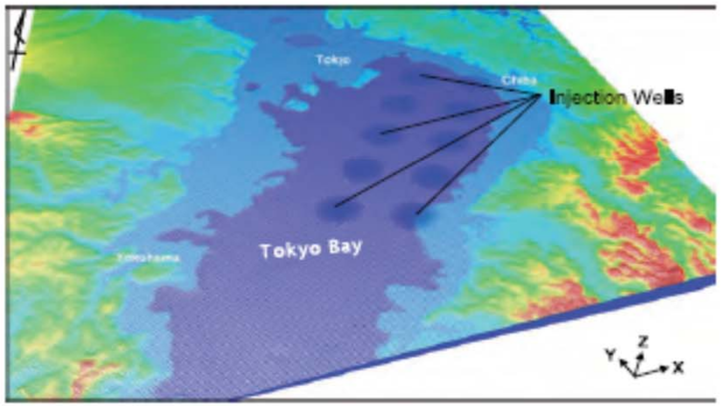
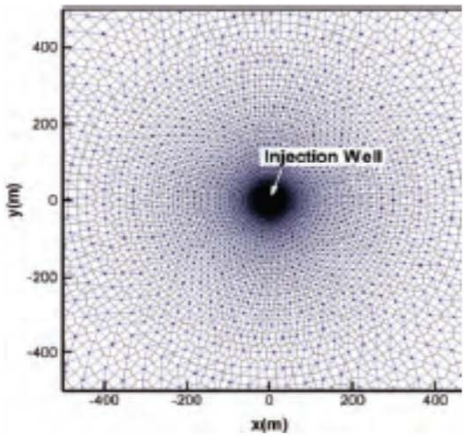
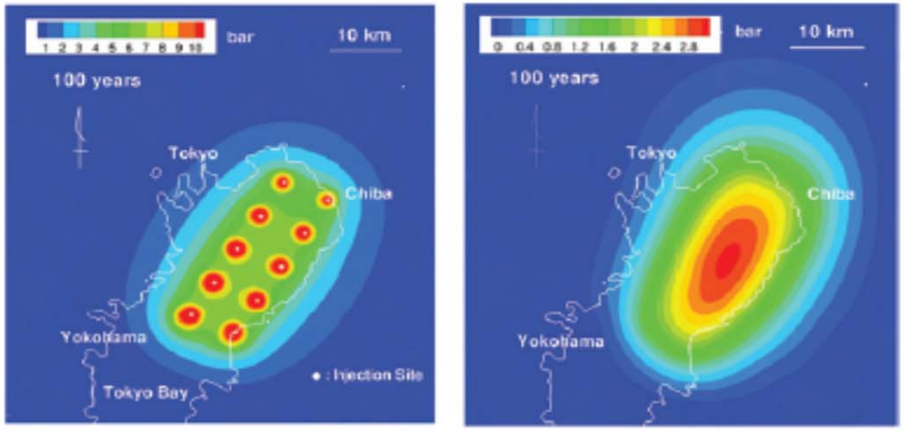


図-4　CO_2 圧入後の地下水圧（全水頭換算）の分布（100 年後）







(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)
  - JAMSTC (Earth Simulator Center) (Software, Hardware)
  - NEC (Software, Hardware)
- 2010 Japan Geotechnical Society (JGS) Award

Science

Modeling

Algorithm

Software

Hardware

Simulation of Geologic CO_2 Storage

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

Concentration of CO_2 in Groundwater

Meshes with higher resolution provide more accurate prediction ⇒ Larger Model/Linear Equations

[Dr. Hajime Yamamoto, Taisei]

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.

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

Prerequisites

- Completed one of the following classes
  - Technical & Scientific Computing I (4820-1027)
  - Seminar on Computer Science I (4810-1204)
- Or, equivalent knowledge and experience in FEM and FEM programming.
  - http://nkl.cc.u-tokyo.ac.jp/13s/
- Knowledge and experiences in fundamental methods for numerical analysis (e.g. Gaussian elimination, SOR)
- Knowledge and experiences in UNIX
- Experiences in programming using FORTRAN or C
- Account for Educational Campuswide Computing System (ECC System) should be obtained in advance:
  - http://www.ecc.u-tokyo.ac.jp/ENGLISH/index-e.html

Grading by Reports ONLY

- MPI (Collective Communication) (S1)
- MPI (1D Parallel FEM) (S2)
- Parallel FEM (S3)

- Sample solutions will be available
- Deadline: February 15th (Sat) 17:00
  - By E-mail: nakajima(at)cc.u-tokyo.ac.jp
  - You can bring hard-copy's to my office ...

Homepage

- **http://nkl.cc.u-tokyo.ac.jp/13w/**
  - General information is available
  - Class materials will be uploaded before Friday evening
  - No hardcopy of course materials are provided (Please print them by yourself)

Schedule

Year	Date	Contents
2013	October 07 (M)	Introduction
	October 15 (T)	Data Structure for Parallel FEM
	October 21 (M)	Oakleaf-FX
	October 28 (M)	Parallel Programming by MPI (1)
	November 05 (T)	Parallel Programming by MPI (2)
	November 11 (M)	Introduction to Tuning/Optimiazation
	November 18 (M)	(No Class)
	November 25 (M)	Example for Report #1
	December 02 (M)	Parallel Programming by MPI (3)
	December 09 (M)	Parallel Programming by MPI (4)
	December 16 (M)	Example for Report #2
2014	January 13 (M)	Parallel 3D FEM (1)
	January 15 (W)	Parallel 3D FEM (2)
	January 20 (M)	Parallel 3D FEM (3)
	January 27 (M)	Recent Topics

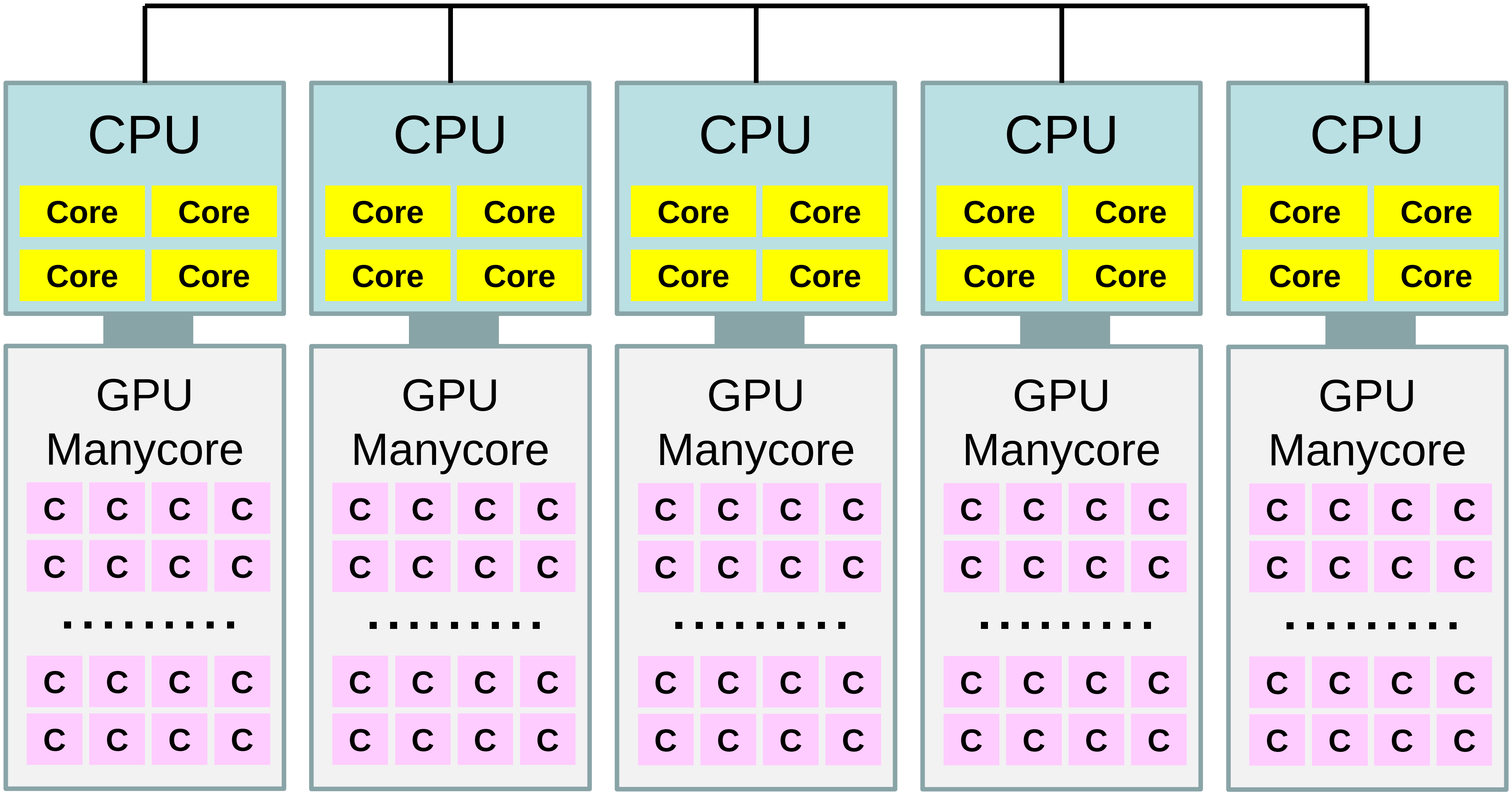
- Target: Parallel FEM
- Supercomputers and Computational Science
- Overview of the Class
- **Future Issues**

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

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

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

Supercomputers with Heterogeneous/Hybrid Nodes

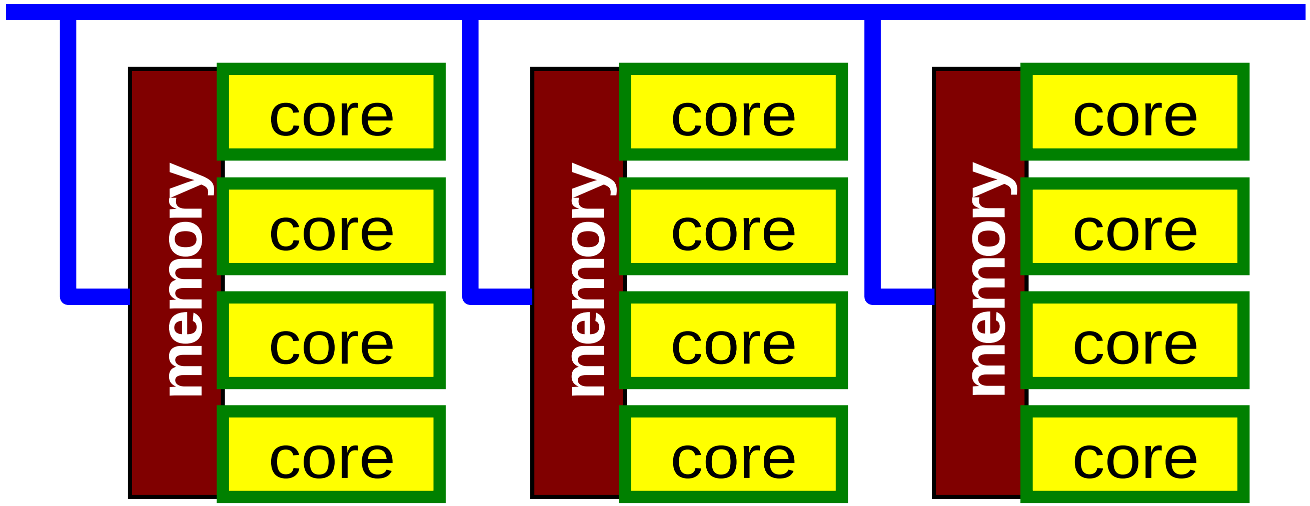


Hybrid Parallel Programming Model is essential for Post-Peta/Exascale Computing

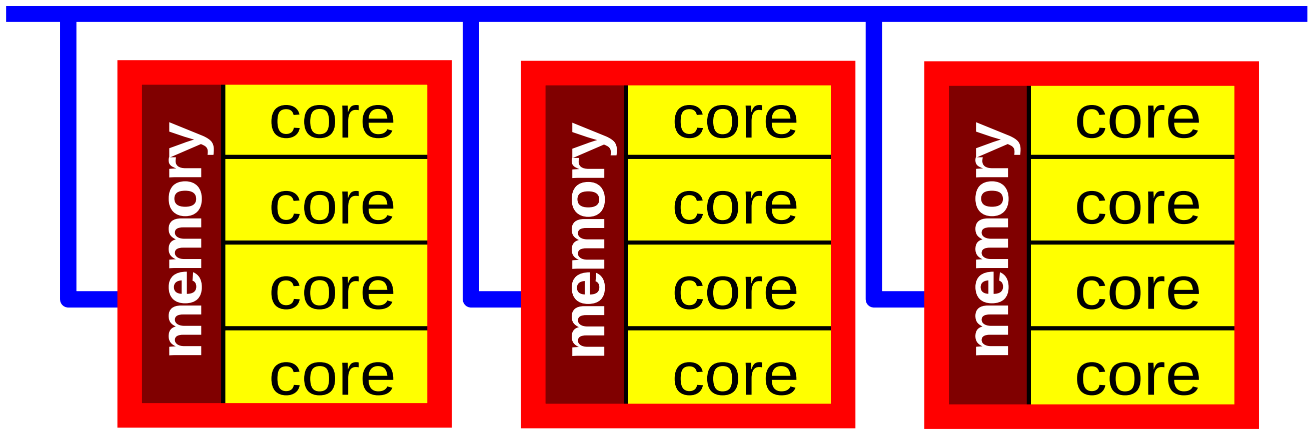
- Message Passing (e.g. MPI) + Multi Threading (e.g. OpenMP, CUDA, OpenCL, OpenACC etc.)
- In K computer and FX10, hybrid parallel programming is recommended
  - MPI + Automatic Parallelization by Fujitsu's Compiler
- Expectations for Hybrid
  - Number of MPI processes (and sub-domains) to be reduced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

Flat MPI vs. Hybrid

Flat-MPI：Each PE -> Independent



Hybrid：Hierarchal Structure



In this class...

- You do not have enough time to learn hybrid parallel programming model.
- But you can easily extend the ideas in materials on MPI and (OpenMP) to hybrid parallel programming models.

- Anyway, MPI is essential for large-scale scientific computing. If you want to something new using supercomputers, you must learn MPI, then OpenMP.
  - You don’t have to be attracted by PGAS (e.g. HPF), automatic parallelization（自動並列化）, etc.

Example of OpnMP/MPI Hybrid

Sending Messages to Neighboring Processes

MPI: Message Passing, OpenMP: Threading with Directives

```
!C
!C- SEND

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

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

Parallel Programming Models

- Multicore Clusters (e.g. K, FX10)
  - MPI + OpenMP and (Fortan/C/C++)
- Multicore + GPU (e.g. Tsubame)
  - 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 (e.g. Stampede)
  - Xeon-Phi needs host CPU (currently)
  - 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)
  - 2018-2020
    - 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)

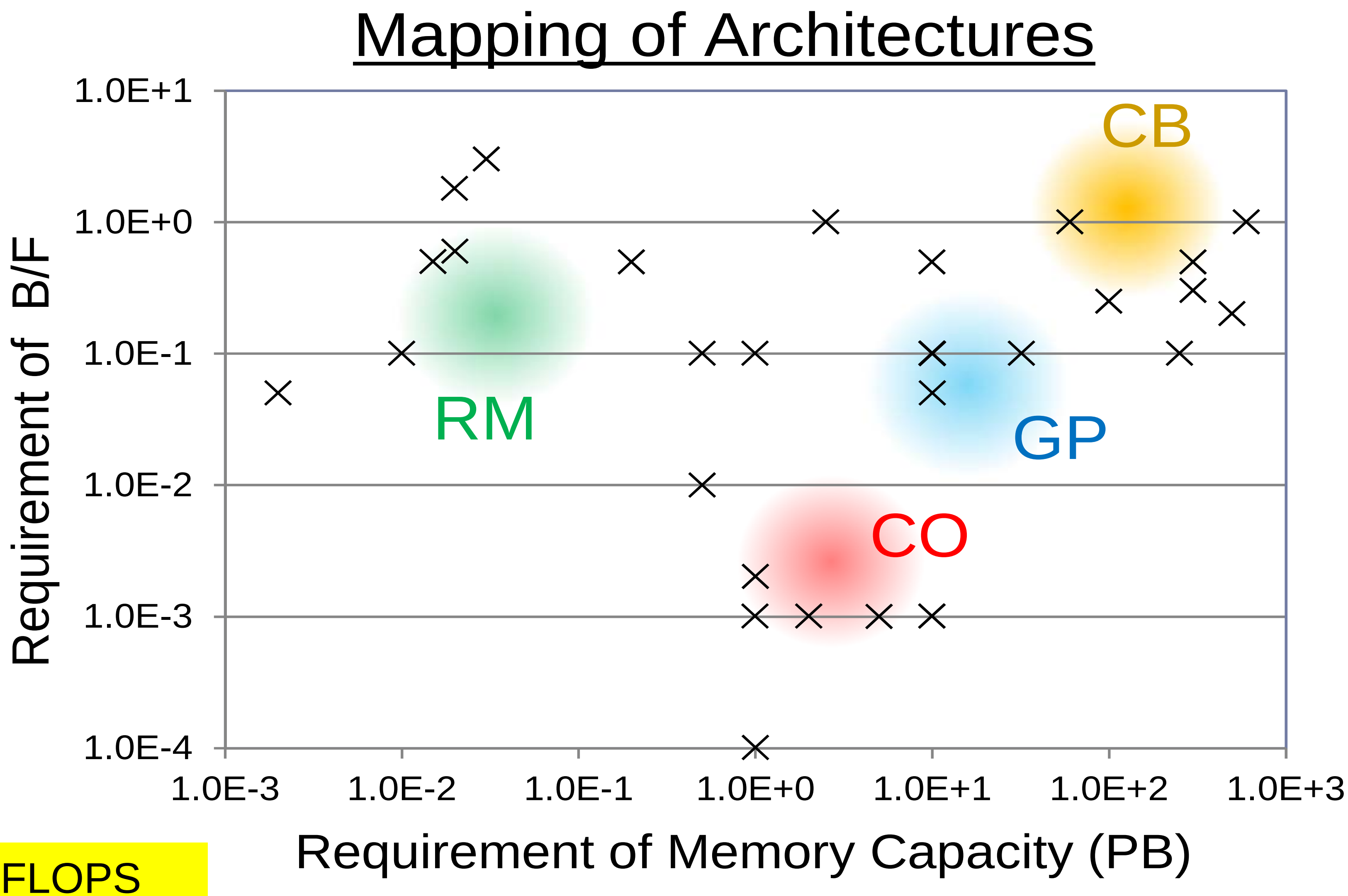
- Not only hardware, but also numerical models and algorithms must be improved:
  - 省電力アルゴリズム（Power-Aware/Reducing）
  - 耐故障アルゴリズム（Fault Resilient）
  - 通信削減アルゴリズム（Communication Avoiding/Reducing）
- Co-Design by experts from various area (SMASH) is important
  - Exascale system will be a special-purpose system, not a general-purpose one.

SDHPC

Workshop on Strategic Development of High Performance Computers

- Series of domestic meetings towards development of post-peta/exascale systems in Japan
  - Architectures, System Software, Compiler, Applications, Algorithms
  - Academia & Industries
- Since August 2010-
  - 10^{th} Workshop, July 30^{th}, 2013 in Kita-Kyushu
- **White paper/roadmap for strategic direction/development of HPC in Japan, published in March 2012**

System Requirement for Target Sciences by 2020

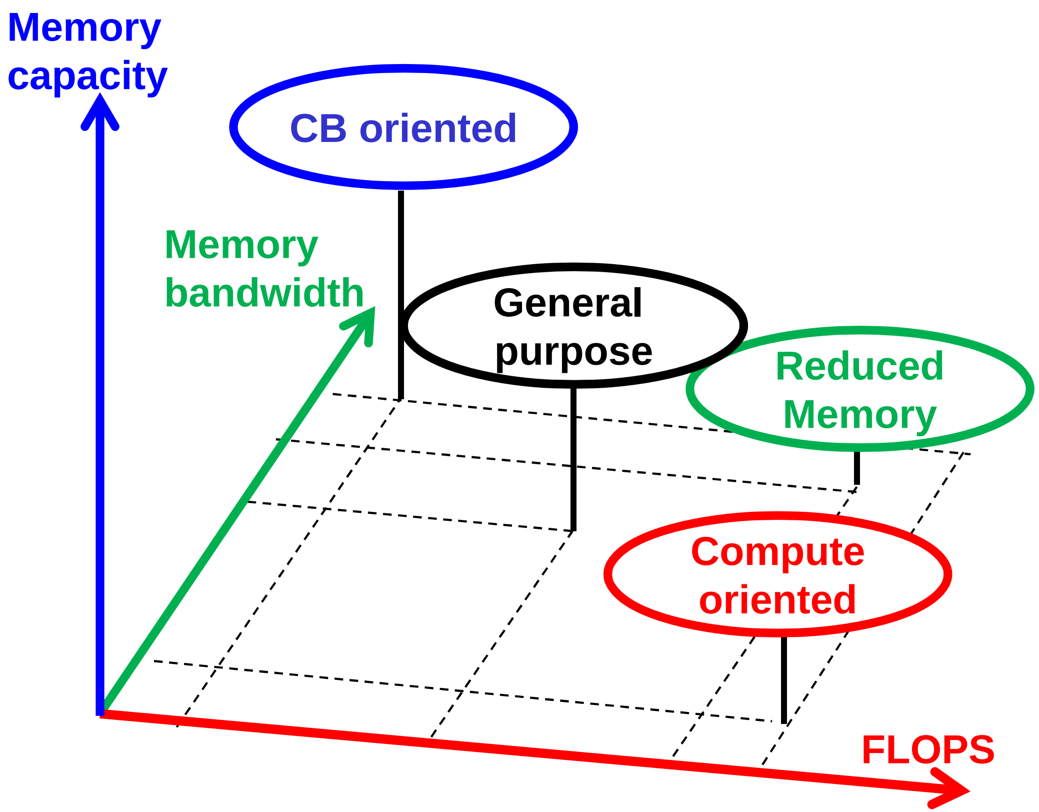


- 800 – 2500 PFLOPS
- 10TB – 500 PB
- B/F: 10^{-3}-10^{0}

Source: Masaaki Kondo's presentation at IESP Kobe meeting, 2012

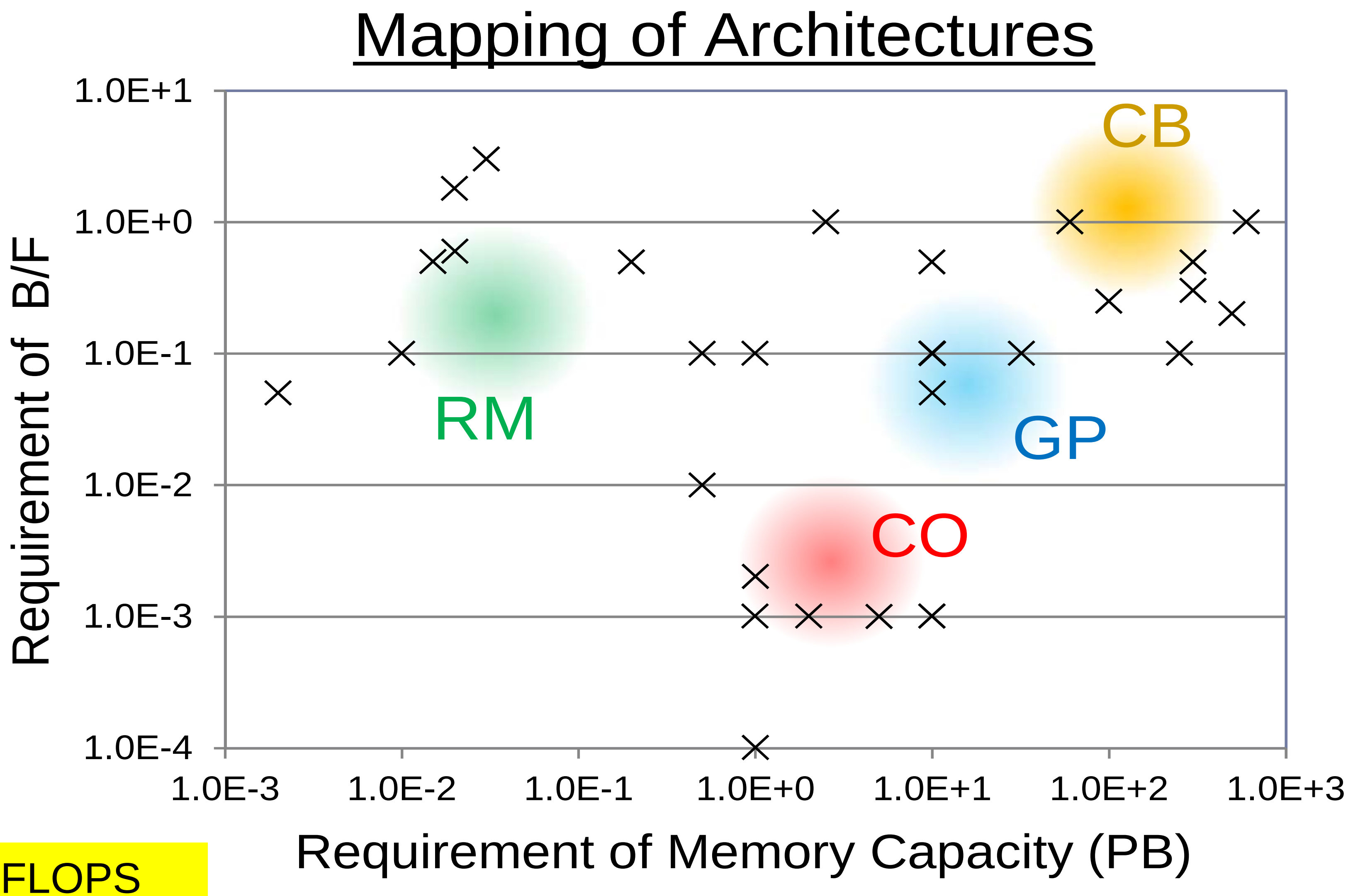
Candidate of the Post Peta-scale Architectures

- Four types of architectures are considered
  - General Purpose (GP)
    - Ordinary CPU-based MPPs
    - e.g.) K-Computer, GPU, Blue Gene, x86-based PC-clusters
  - Capacity-Bandwidth oriented (CB)
    - With expensive memory-I/F rather than computing capability
    - e.g.) Vector machines
  - Reduced Memory (RM)
    - With embedded (main) memory
    - e.g.) SoC, MD-GRAPE4, Anton
  - Compute Oriented (CO)
    - Many processing units
    - e.g.) ClearSpeed, GRAPE-DR



Source: Masaaki Kondo's presentation at IESP Kobe meeting, 2012

System Requirement for Target Sciences by 2020



- 800 – 2500 PFLOPS
- 10TB – 500 PB
- B/F: 10^{-3}-10^{0}

Source: Masaaki Kondo's presentation at IESP Kobe meeting, 2012

Feasibility Study of Future HPC R&D in Japan

- 2-year project for feasibility study of advanced HPC funded by Japanese Government (FY.2012 & 2013)
  - SDHPC, White Paper
- 4 Interdisciplinary Research Teams are selected
  - Hardware, Software, Application, Algorithm
    - Academia, Industry
  - Tohoku U., U. Tsukuba, **U.Tokyo**, RIKEN/Titech
  - Approx. 5M USD/yr. (1.0M-1.5M USD/yr./team)
- Keywords
  - Science-Driven, Co-Design

MEXT MINISTRY OF EDUCATION, CULTURE, SPORTS, SCIENCE AND TECHNOLOGY-JAPAN

- Results of this feasibility study will be the proposal for funding on development exascale system(s) in Japan.
  - In May 2013, MEXT announced development of Exascale System (FY. 2014-2020, 1B USD). ... Olympic in 2020 also.

Feasibility Study on Future HPC R&D in Japan

Program promotion board

Member: The head of each team and other specialists

Role: To check the progress of the each team and to coordinate the collaboration among the teams

1 application study team

RIKEN AICS and TITECH
Collaboration with application filelds

- Identification of scientific and social issues to be solve in the future
- Drawing Science road map until 2020
- Selection of the applications that plays key roles in the roadmap
- Review of the architectures using those applications

3 system study teams

CB
Tohoku Univ. and NEC

RM+CO
U. of Tsukuba, Titech, and Hitachi

GP
U. [illegible]yo, Kyushu U., Fujitsu, Hitachi, and NEC

- Design of computer systems solving scientific and social issues
- Identification of R&D issues to realize the systems
- Review of the system using the application codes
- Estimation of the system's cost

Towards Next-generation General Purpose Supercomputer

Feature of Target System:

- ✓Deployment around 2018-2020
- ✓Power consumption 30MW, 2000 m^2 constraints
- ✓Extension of K/FX10.
- ✓Co-Design, Memory-Bound Applications

PI: Yutaka Ishikawa, U. of Tokyo
- Organization
- System Software Stack
- Performance Prediction and Tuning

Applications

System Software Stack
(MPI, parallel file I/O, PGAS,
Batch Job Scheduler, Debugging and Tuning Tools)

Commodity-based Supercomputer

Next-Gen General Purpose Supercomputer

Co-PI: Yuichi Nakamura, NEC
- System Software Stack

Co-PI: Tsuneo Iida, Hitachi
- Storage Architecture and System Software Stack

Co-PI: Kei Hiraki, U. of Tokyo
- Architecture Evaluation, Compiler, and Low power technologies

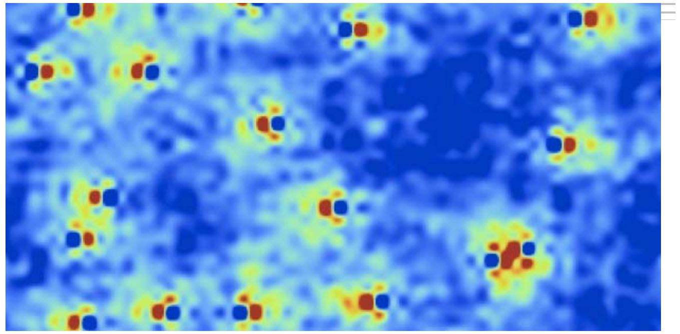
Co-PI: Mutsu Aoyagi, Kyushu U.
- Network Evaluation Environment

Co-PI: Naoki Shinjo, Fujitsu
- Processor, Node, Interconnect Architecture and System Software Stack

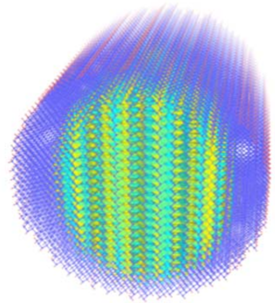
〔Ishikawa, 2012〕

Target Applications considered in FY2012

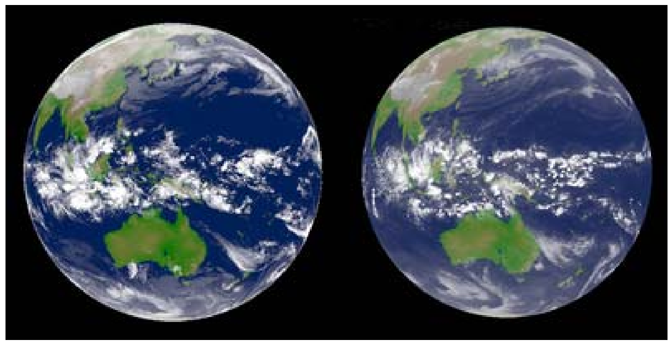
- ALPS(Algorithms and Libraries for Physics Simulations)
  - Providing high-end simulation codes for strongly correlated quantum mechanical systems
  - Total Memory: 10～100PB, low latency and high radix network



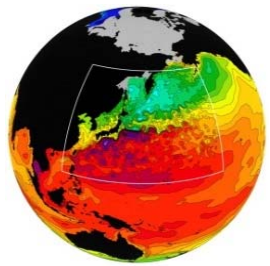
- RSDFT (Real-Space Density Functional Theory)
  - A DFT(Density Functional Theory) code with real space discretized wave functions and densities for molecular dynamics simulations using the Car-Parrinello type approach
  - Total Memory: 1PB
  - 1EFLOPS (B/F = 0.1+)



- NICAM (Nonhydrostatic ICosahedral Atmospheric Model)
  - A Global Cloud Resolving Model (GCRM)
  - Total Memory:1 PB, Memory Bandwidth: 300 PB/sec
  - 100 PFLOPS (B/F = 3)

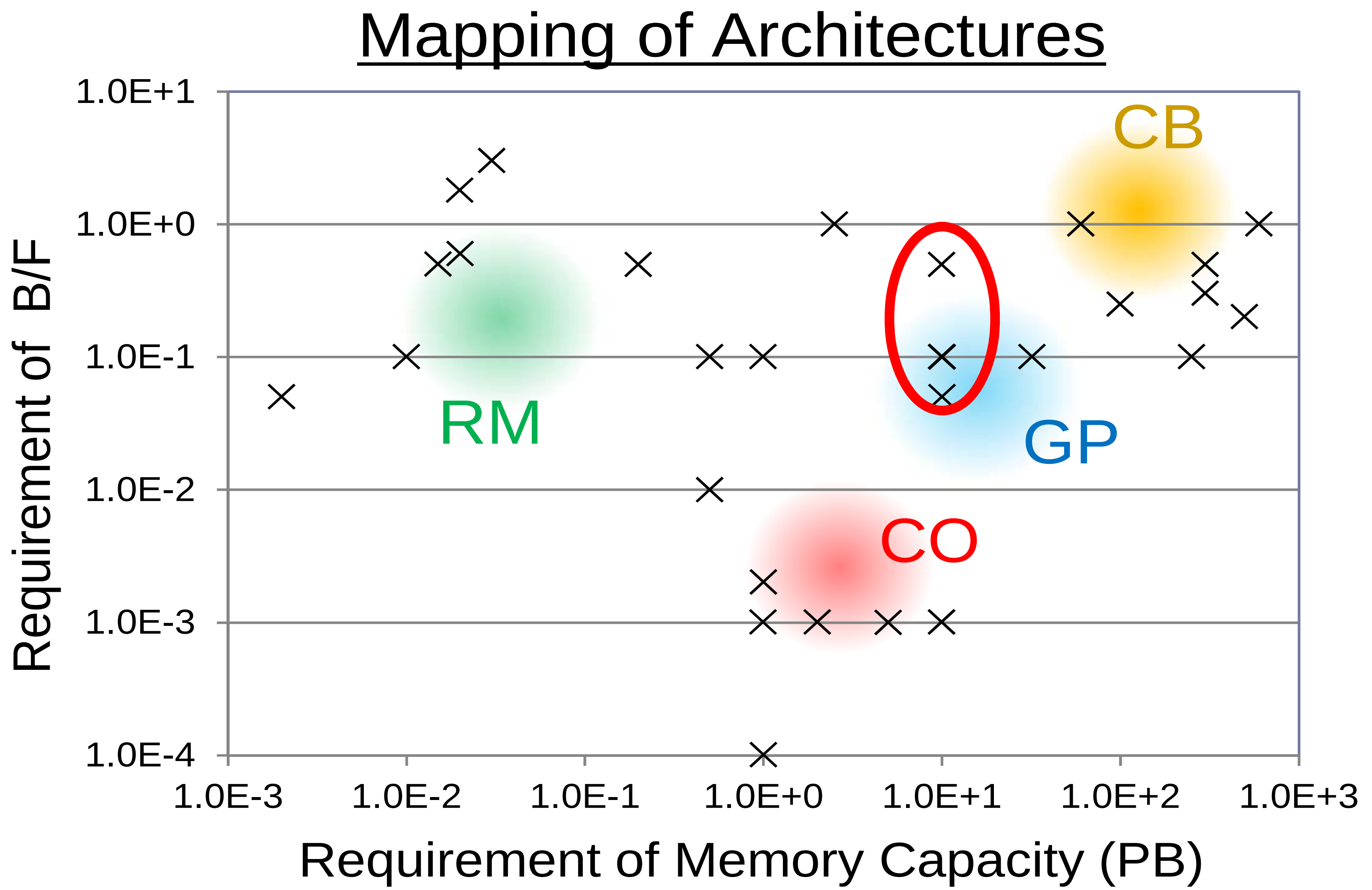


- COCO (CCSR Ocean Component Model)
  - ocean general circulation model developed at Center for Climate System Research (CCSR), the University of Tokyo
  - Total Memory: 320 TB, Memory Bandwidth:150 PB/sec.
  - 50 PFLOPS (B/F=3)



〔Ishikawa, 2012〕

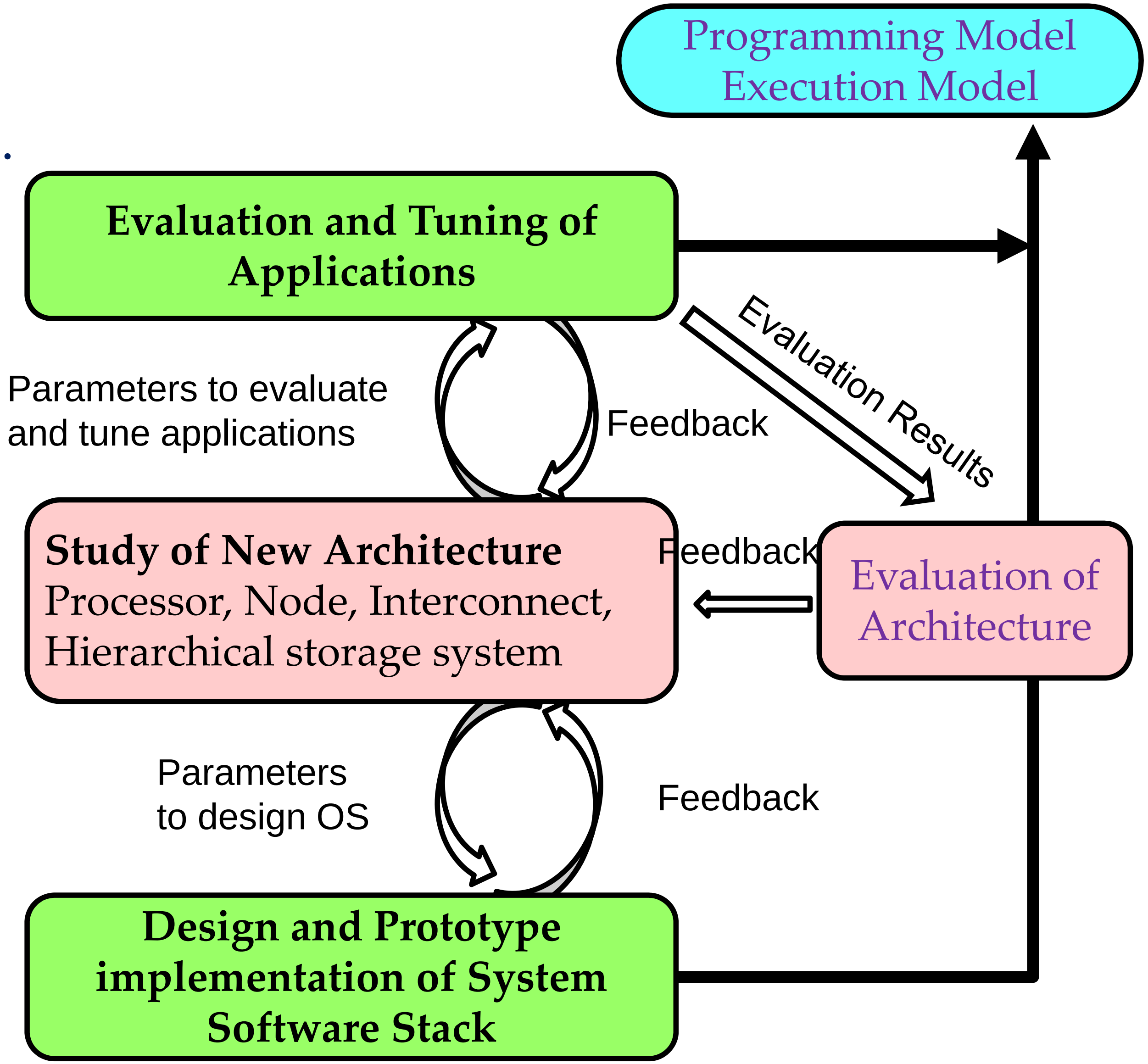
Target Applications considered in FY2012



Source: Masaaki Kondo's presentation at IESP Kobe meeting, 2012

Co-design Strategy

- Four teams, architecture design, application tuning, architecture evaluation, and system software design, are intensively cooperative.
- Every two months, the architecture design team provides architectural parameters
  - To evaluate and tune applications
  - To design and implement system software stack
  - To evaluate architecture
- In FY2013 More applications will be used to evaluate the architecture
- **Good system for Linpack is not necessarily a good one for certain applications**



〔Ishikawa, 2012〕