

はじめに, 並列有限要素法とは

2011年度冬学期

中島研吾

科学技術計算Ⅱ (4820-1028)・コンピュータ科学特別講義(4810-1205)
(並列有限要素法)

概要

- はじめに
- MPIとは
- 並列有限要素法とは？

本講義の目的(1/3)

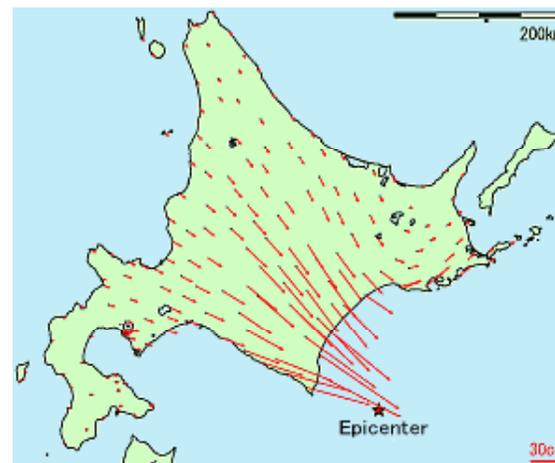
- 科学技術計算Ⅱ(4820-1028)
 - 情報理工学系数理情報学専攻
- コンピュータ科学特別講義Ⅱ(4810-1205)
 - 情報理工学系コンピュータ科学専攻
- 科学技術計算Ⅰ・コンピュータ科学特別講義Ⅰ(有限要素法)
(夏学期)に引き続き以下の講義, 実習を実施:
 - MPIによる並列計算プログラミング入門
 - 並列有限要素法のためのデータ構造
 - 並列プログラムの作成法
 - T2Kオープンスパコンによるプログラミング実習
- 夏学期に扱った三次元弾性静解析プログラム「fem3d」の並列化をMPIによって実施する

並列計算の意義

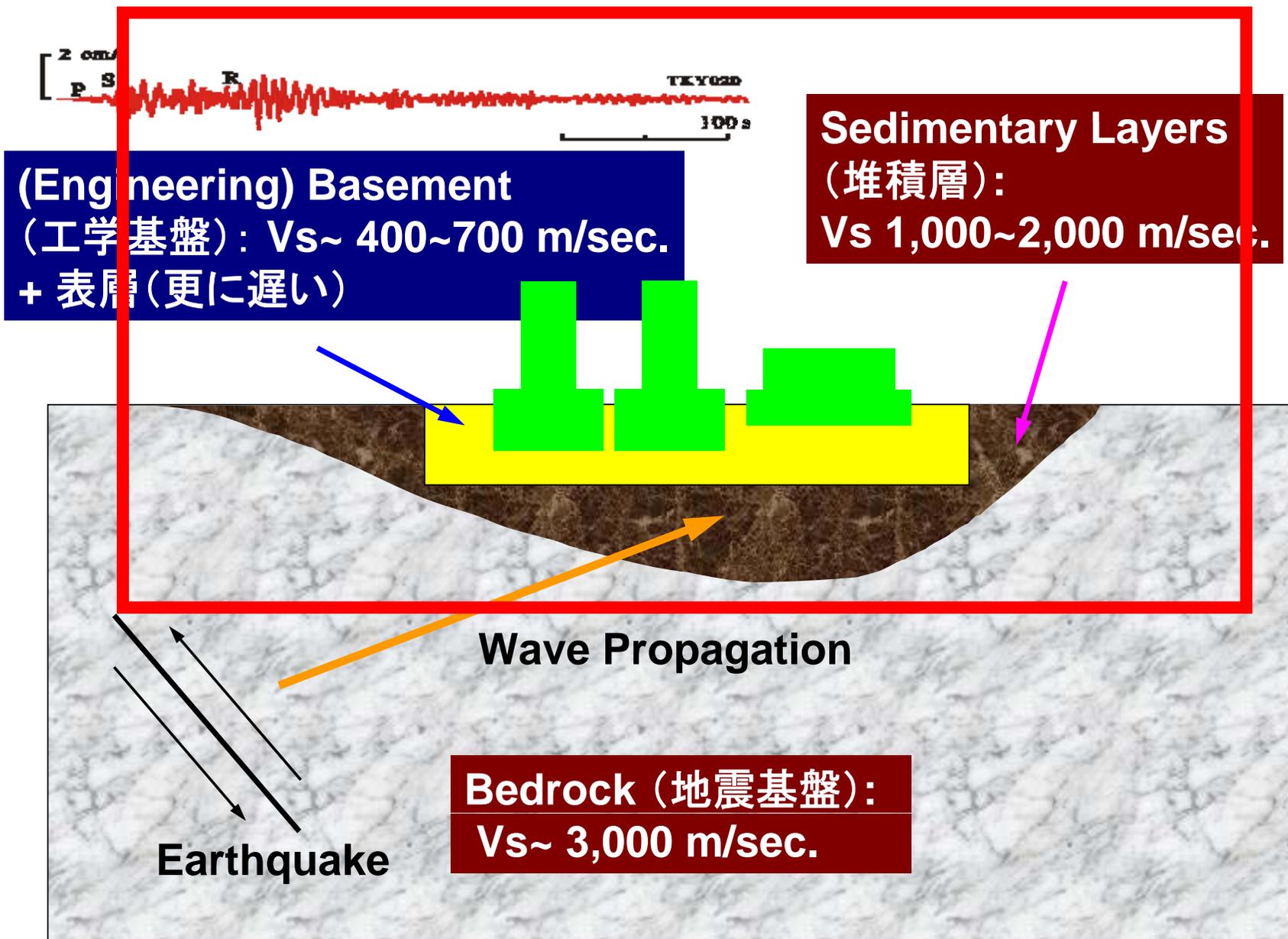
- たくさんの計算機を使って
- より速く
- より大規模に
- より複雑に
 - 連成, 連結シミュレーション

2003年 十勝沖地震

長周期地震波動(表面波)のために苫小牧の
石油タンクがスロッシングを起こし火災発生

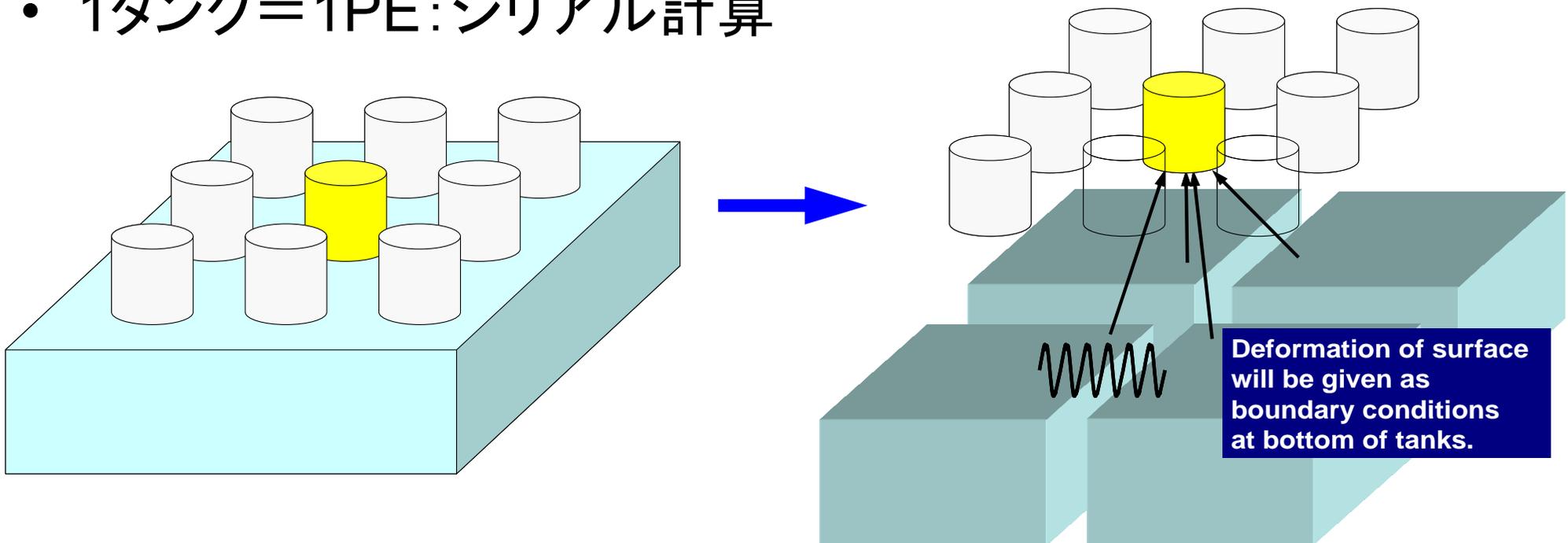


地盤・石油タンク振動連成シミュレーション



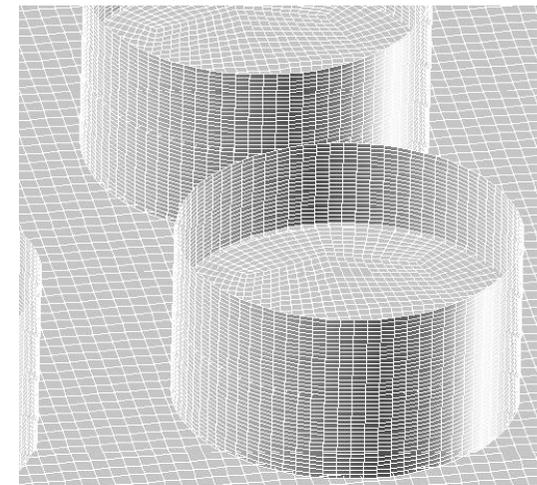
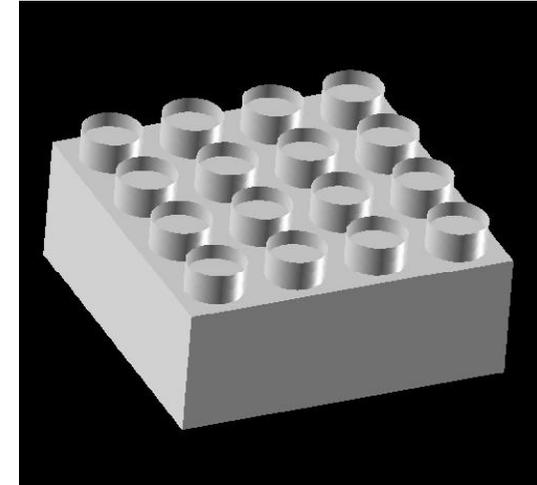
対象とするアプリケーション

- 地盤・石油タンク振動
 - 地盤⇒タンクへの「一方向」連成
 - 地盤表層の変位 ⇒ タンク底面の強制変位として与える
- このアプリケーションに対して、連成シミュレーションのためのフレームワークを開発，実装
- 1タンク=1PE:シリアル計算

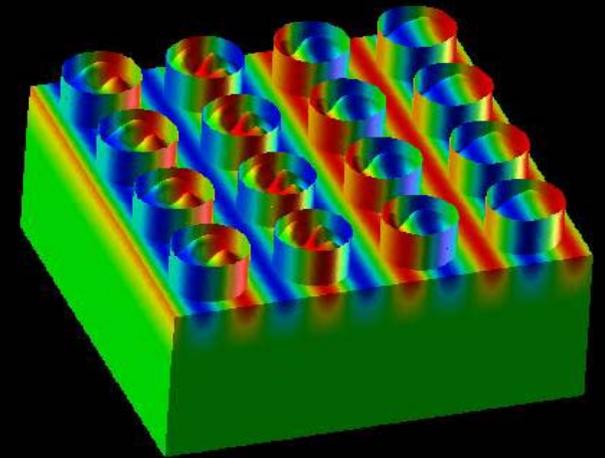
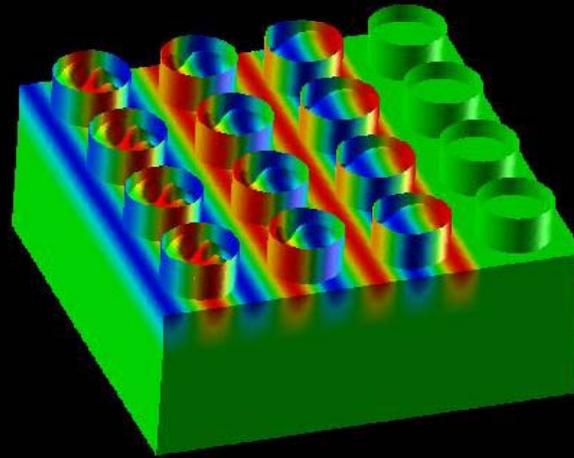
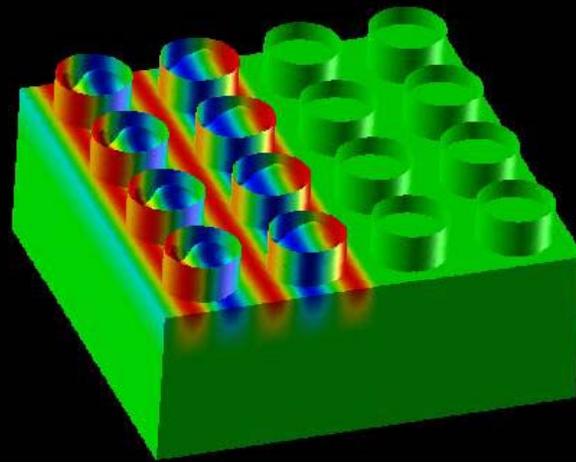
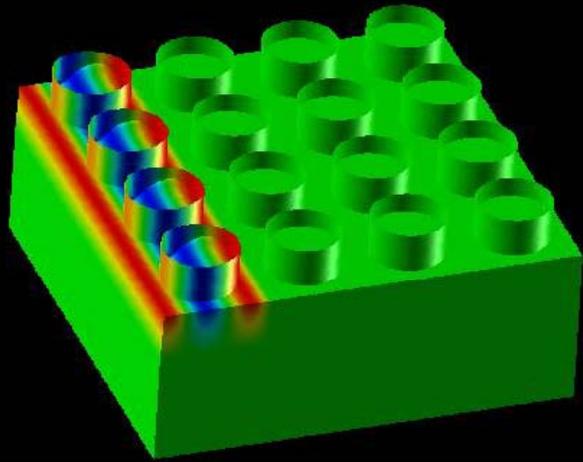


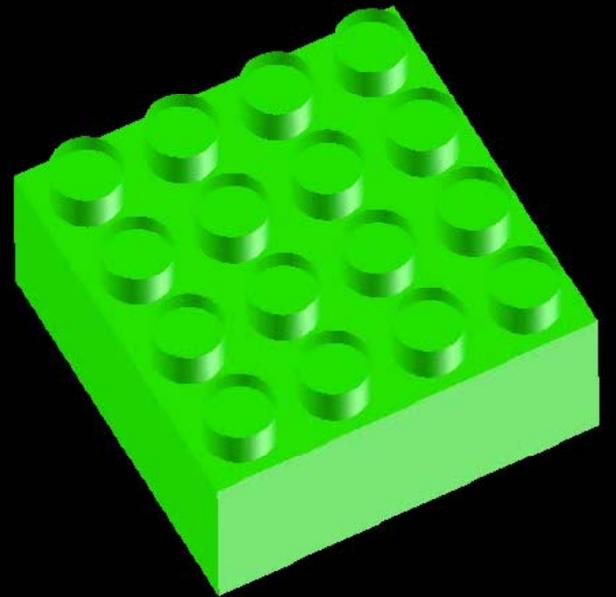
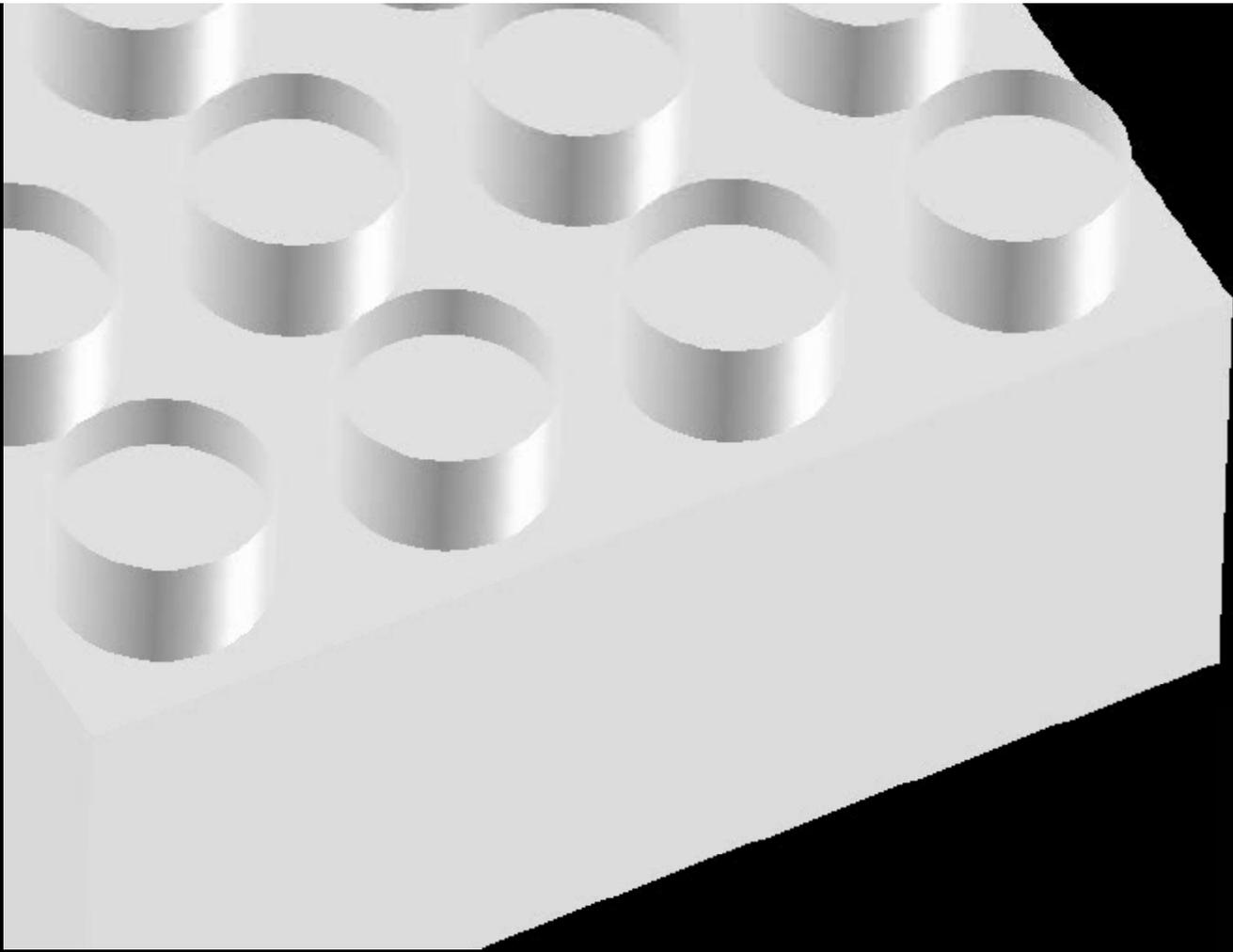
地盤，タンクモデル

- 地盤モデル：FORTRAN
 - 並列FEM, 三次元弾性動解析
 - 前進オイラー陽解法, EBE
 - 各要素は一辺2mの立方体
 - 240m × 240m × 100m
- タンクモデル：C
 - シリアルFEM(EP), 三次元弾性動解析
 - 後退オイラー陰解法, スカイライン法
 - シェル要素+ポテンシャル流(非粘性)
 - 直径:42.7m, 高さ:24.9m, 厚さ:20mm, 液面:12.45m, スロッシング周期:7.6sec.
 - 周方向80分割, 高さ方向:0.6m幅
 - 60m間隔で4 × 4に配置
- 合計自由度数:2,918,169



計算結果





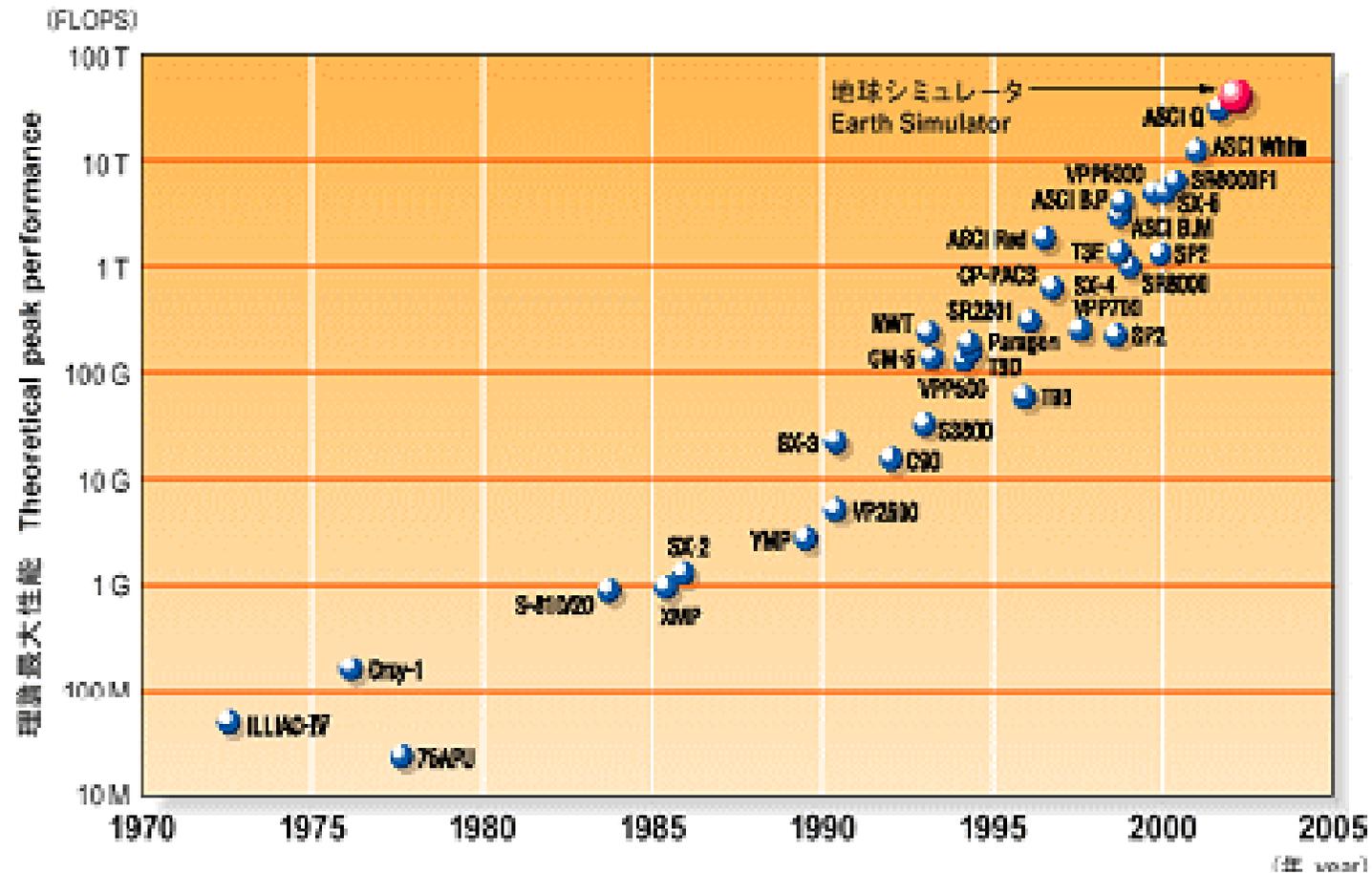
本講義の歴史

- 2008年度: コンピュータ科学専攻
 - 冬学期: コンピュータ科学特別講義 I
 - 現在の I, II を一学期にまとめた内容, FEM + 並列FEM
- 2009年度: コンピュータ科学専攻
 - 夏学期: コンピュータ科学特別講義 I
 - 冬学期: コンピュータ科学特別講義 II
- 2010年度: 数理情報学専攻
 - 夏学期: 科学技術計算 I
 - 冬学期: 科学技術計算 II
- 2011年度: 数理情報学・コンピュータ科学専攻
 - 夏学期: 科学技術計算 I, コンピュータ科学特別講義 I
 - 冬学期: 科学技術計算 II, コンピュータ科学特別講義 II

計算機ハードウェアの歴史

- プロセッサは1.5年に2倍の割合で処理速度が増加している
 - Moore's Law: 集積度が18ヶ月から24ヶ月で2倍
- 並列計算機の発達
- 1983年: 1 GFLOPS, 1996年: 1 TFLOPS, 2002年: 36 TFLOPS, 2005年: 280TFLOPS, 2008年: 1,000TFLOPS
 - MFLOPS: Millions of Floating Point Operations per Second. (1秒間に 10^6 回の浮動小数点処理)
 - GFLOPS: 10^9 回, TFLOPS: 10^{12} 回, PFLOPS: 10^{15} 回

計算機ハードウェア発達の歴史



TOP 500 List

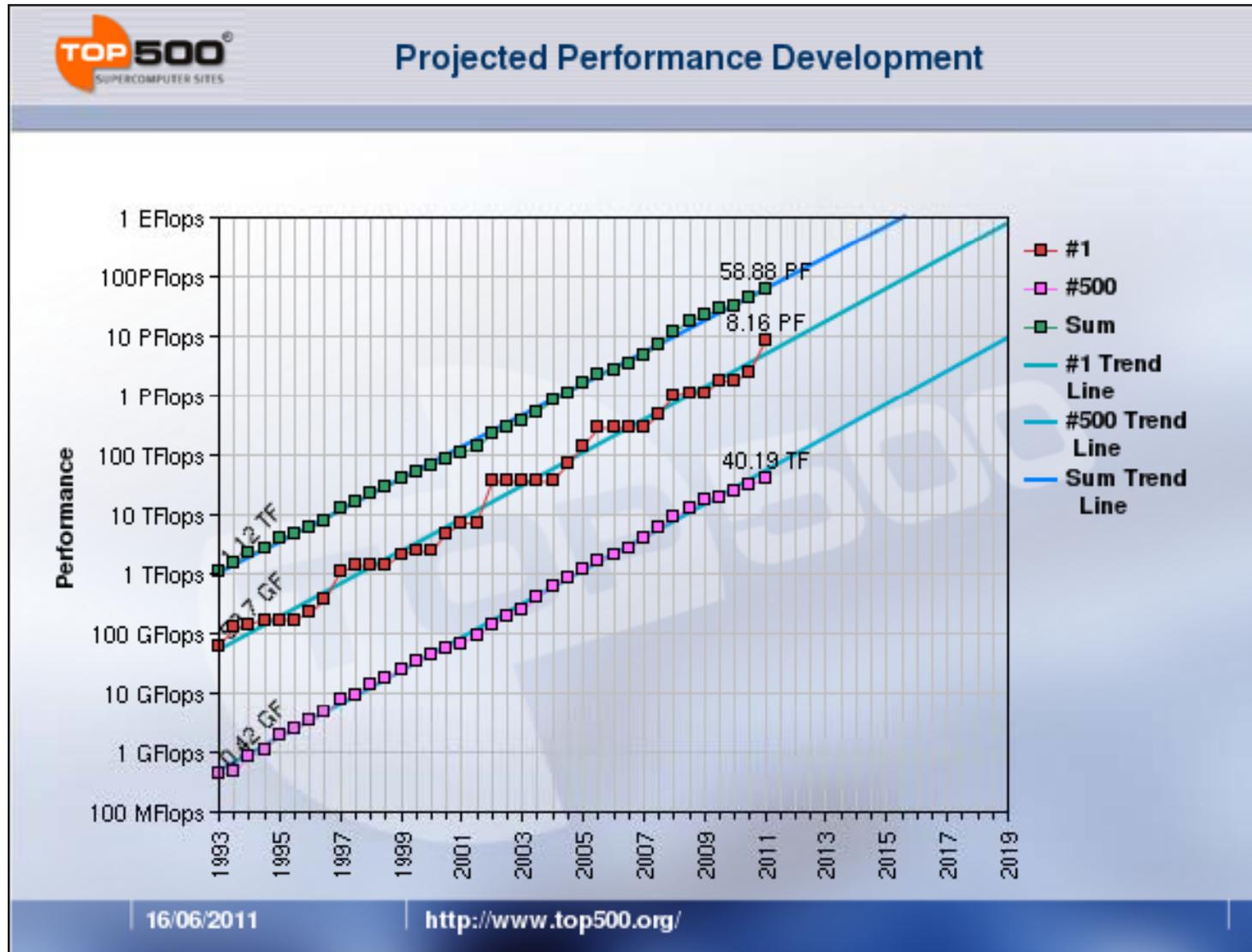
<http://www.top500.org/>

- 年2回更新
- LINPACKと言われるベンチマークテストを実施する。
 - 密行列を係数とする連立一次方程式を解く
 - ベクトル機でもスカラー機でも性能が出やすい
- 実際のアプリケーションではこれほどの性能は出ない
 - 差分法, スペクトル法系の手法: ピーク性能の60%程度
 - AFES on the Earth Simulator (Old): 26 TFLOPS (ピーク性能の65%)
 - 有限要素法
 - GeoFEM on the Earth Simulator (512ノード) (Old): 10 TFLOPS (30%)
 - スカラー機ではこれほど出ない: 5%~10%
- 最近では消費電力も考慮したGreen500というランキングもある

	Site	Computer/Year Vendor	Cores	R_{\max}	R_{peak}	Power
1	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect / 2011 Fujitsu	548352	8162.00	8773.63	9898.56
2	National Supercomputing Center in Tianjin China	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C / 2010 NUDT	186368	2566.00	4701.00	4040.00
3	DOE/SC/Oak Ridge National Laboratory United States	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz / 2009 Cray Inc.	224162	1759.00	2331.00	6950.60
4	National Supercomputing Centre in Shenzhen (NSCS) China	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU / 2010 Dawning	120640	1271.00	2984.30	2580.00
5	GSIC Center, Tokyo Institute of Technology Japan	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows / 2010 NEC/HP	73278	1192.00	2287.63	1398.61
6	DOE/NNSA/LANL/SNL United States	Cielo - Cray XE6 8-core 2.4 GHz / 2011 Cray Inc.	142272	1110.00	1365.81	3980.00
7	NASA/Ames Research Center/NAS United States	Pleiades - SGI Altix ICE 8200EX/8400EX, Xeon HT QC 3.0/Xeon 5570/5670 2.93 Ghz, Infiniband / 2011 SGI	111104	1088.00	1315.33	4102.00
8	DOE/SC/LBNL/NERSC United States	Hopper - Cray XE6 12-core 2.1 GHz / 2010 Cray Inc.	153408	1054.00	1288.63	2910.00
9	Commissariat a l'Energie Atomique (CEA) France	Tera-100 - Bull bullx super-node S6010/S6030 / 2010 Bull SA	138368	1050.00	1254.55	4590.00
10	DOE/NNSA/LANL United States	Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband / 2009 IBM	122400	1042.00	1375.78	2345.50

R_{\max} : 実効性能 (GFLOPS), R_{peak} : ピーク性能 (GFLOPS)

<http://www.top500.org/>



- PFLOPS: Peta ($=10^{15}$) Floating OPerations per Sec.
- Exa-FLOPS ($=10^{18}$) will be attained in 2019

次世代スーパーコンピュータ「京」

- 理化学研究所計算科学研究機構
(神戸)
 - <http://www.aics.riken.jp/>
- 京速計算機
 - 「京」=「兆」の10,000倍 = $10 \times 10^{15} = 10$ Peta FLOPS
- 2011年6月現在世界最速(未完成)



広範な分野での利活用

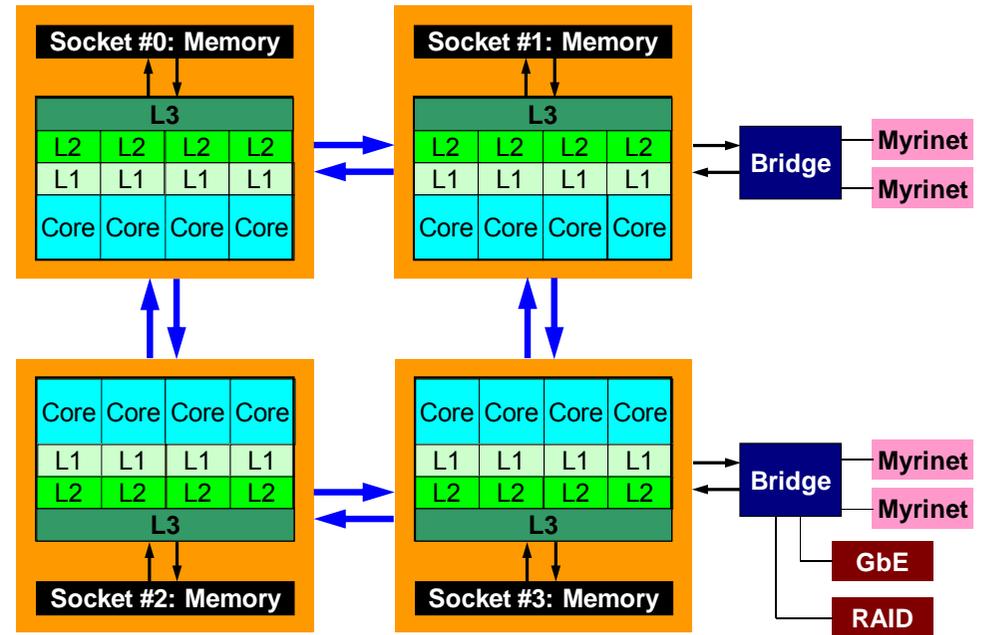
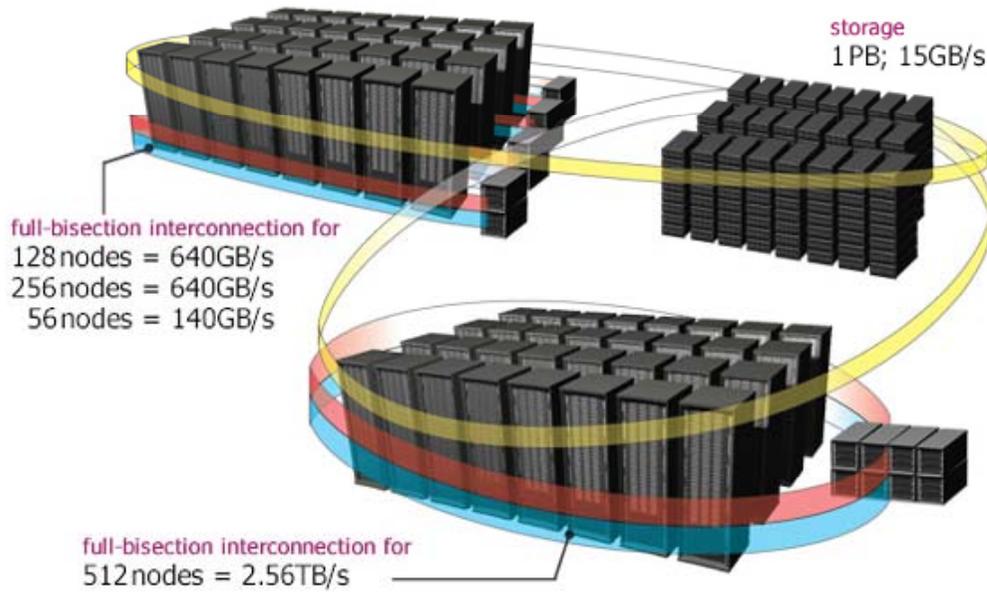
一次世代スーパーコンピュータが拓く世界ー



T2Kオープンスパコン(筑波, 東大, 京都)

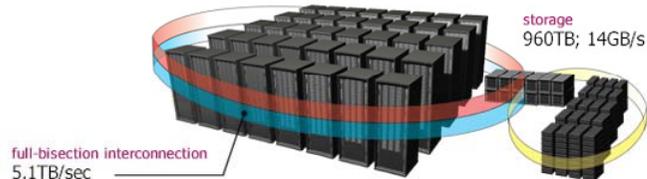
University of Tokyo

nodes = 952 Rpeak = 140.1TFlops Memory = 31TB



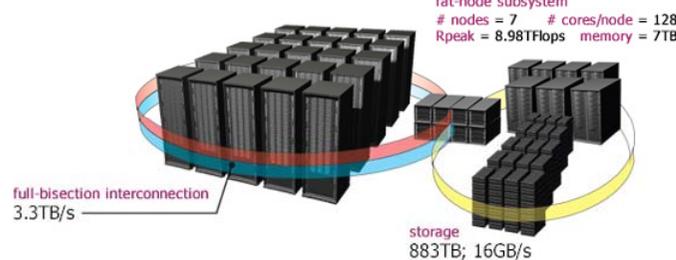
University of Tsukuba

nodes = 648 Rpeak = 95.4TFlops Memory = 20TB



Kyoto University

nodes = 416 Rpeak = 61.2TFlops Memory = 13TB



専用計算機

- GRAPE (GRAvity PipE)
 - 宇宙物理学におけるN対N問題用専用ハードウェア: MD, 境界要素法等
 - コストパフォーマンス
- MDGRAPE
 - MD (Molecular Dynamics) 専用のGRAPE
 - MDGRAPEを通常のクラスタのAcceleratorのように使用することもできる。

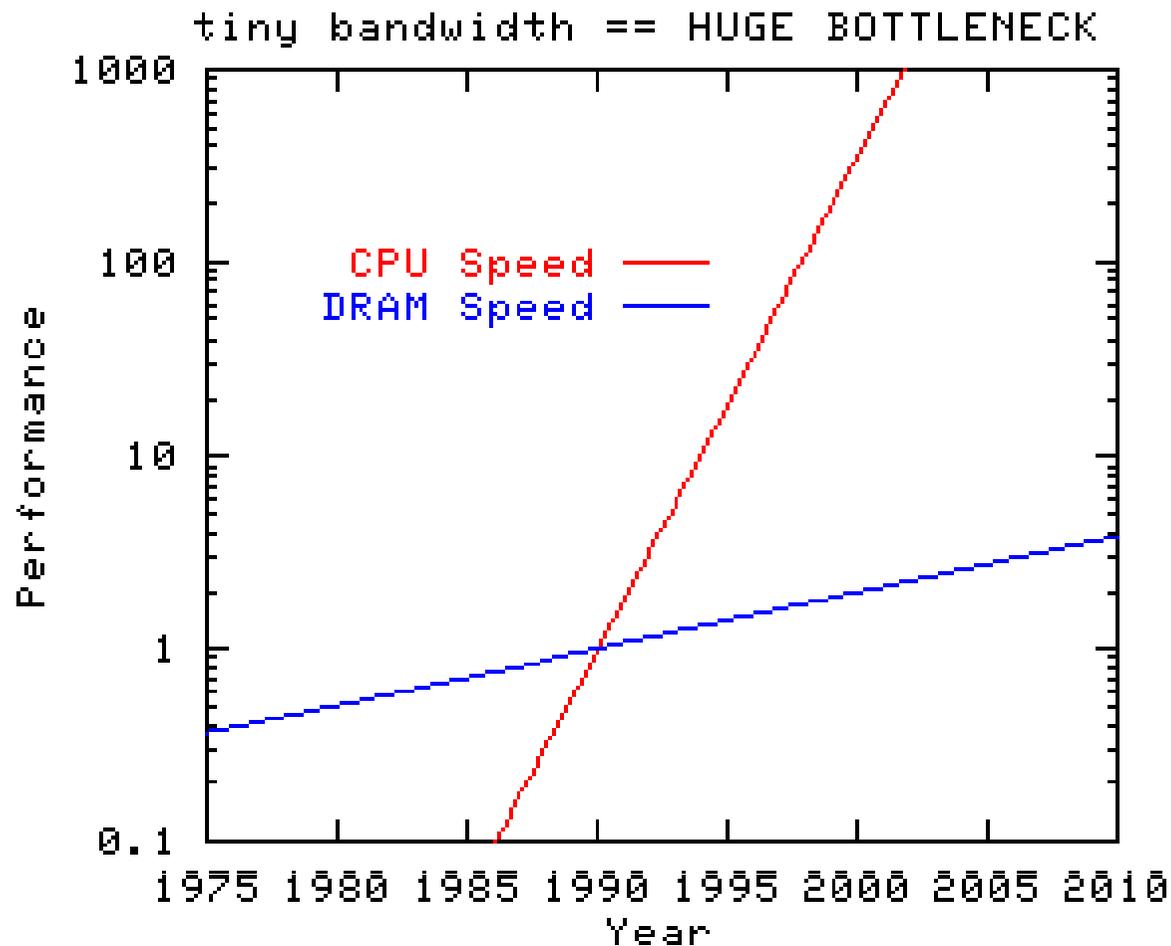
様々な技術的課題

- コアの性能→今後10年間でそれほど変化無し
- 電力消費量
 - Exascale(2019年頃)だと2GW(電気代2,000億円/年)
 - これを100MW未満に抑えるための技術革新必要
 - マルチコア化, メニーコア化
- メモリー性能の低下
 - GPGPU: 高いメモリバンド幅
 - 用途は限られる, プログラミングの困難さ
- 設置場所
- 故障
 - Exascale: 「億」規模のコア数



マイクロプロセッサの動向

CPU性能, メモリバンド幅のギャップ



<http://www.streambench.org/>

疎行列ソルバーの性能: 三次元弾性問題

ICCG法, T2K-SR11000 1ノード: メモリバンド幅が効く

Hitachi SR11000/J2

Power 5+ 2.3GHz x 16

147.2 GFLOPS/node

100 GB/s for STREAM/Triad

L3 cache: 18MB/core

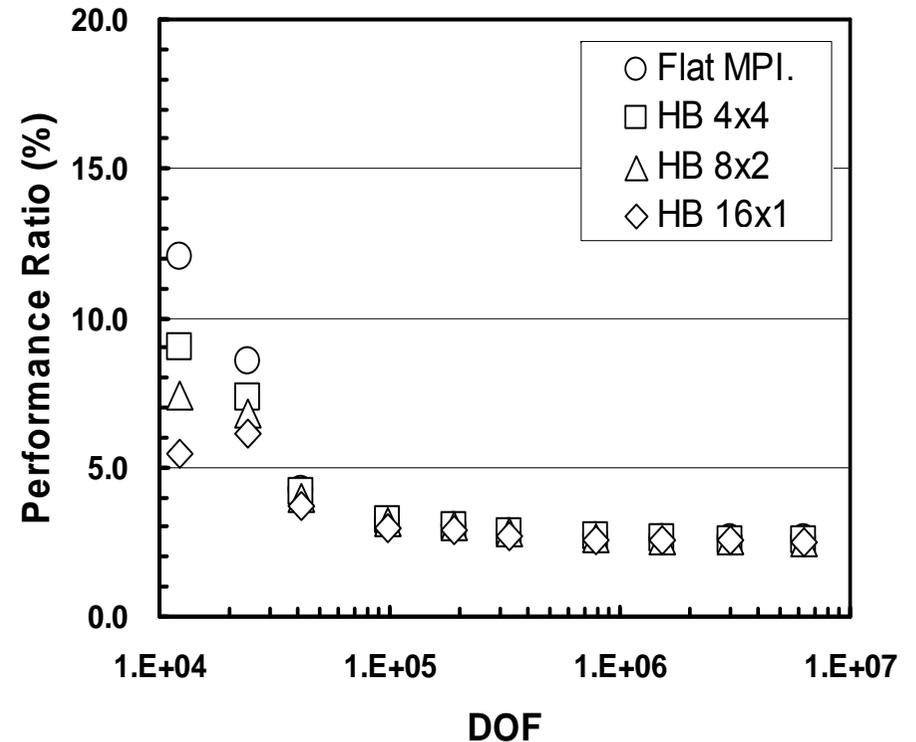
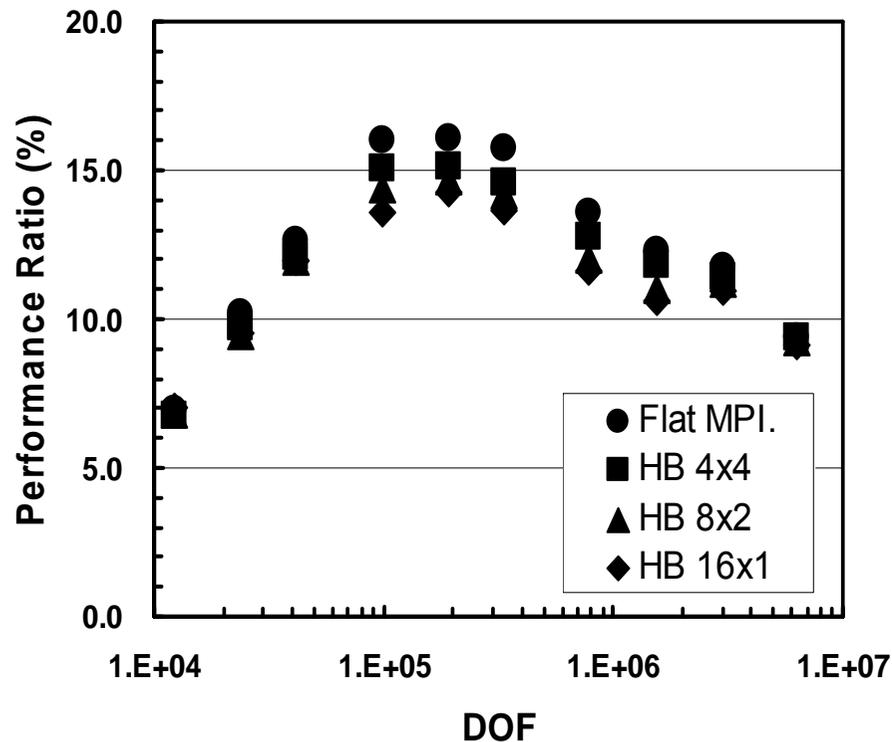
T2K/Tokyo

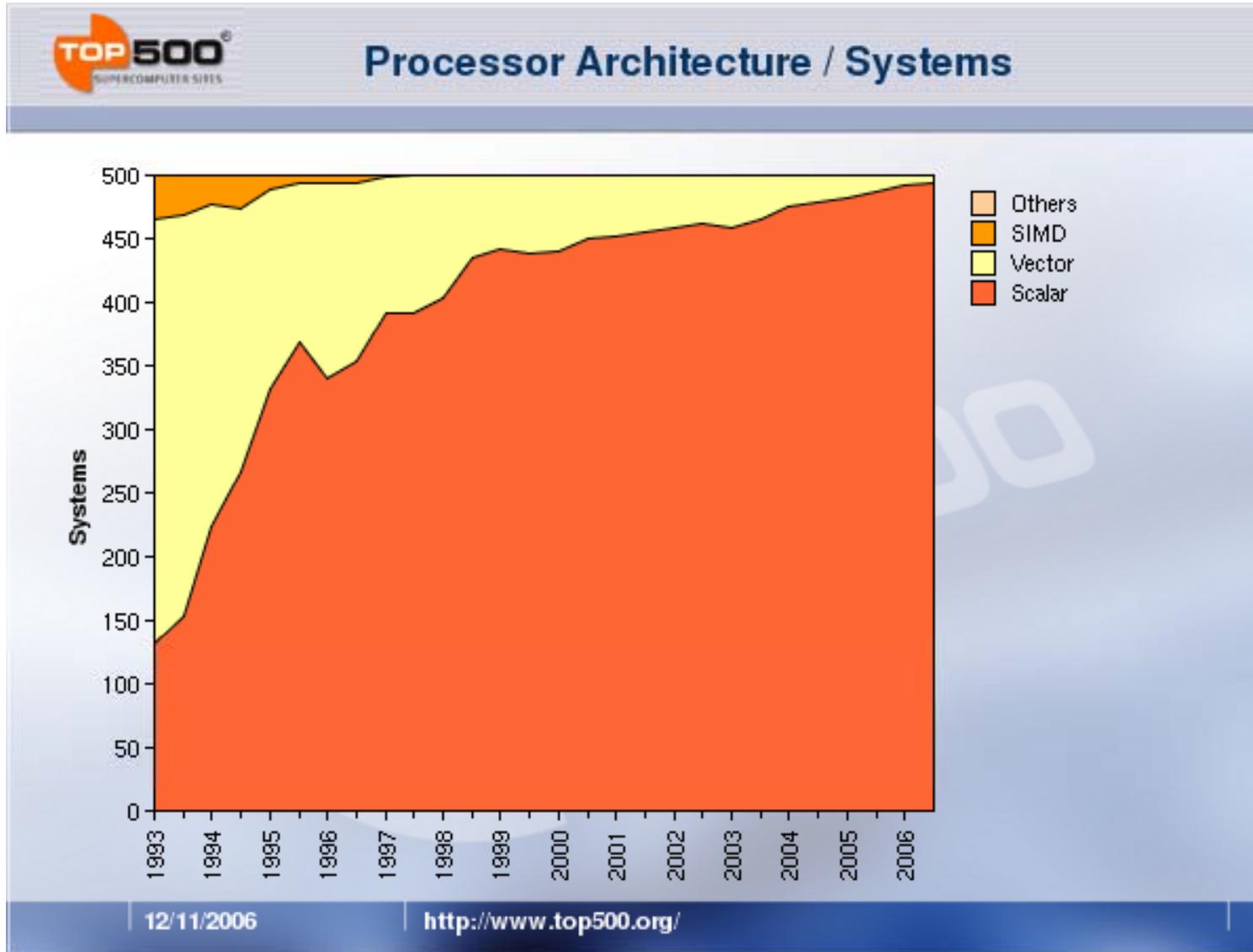
Opteron 2.3GHz x 16

147.2 GFLOPS/node

20 GB/s for STREAM/Triad

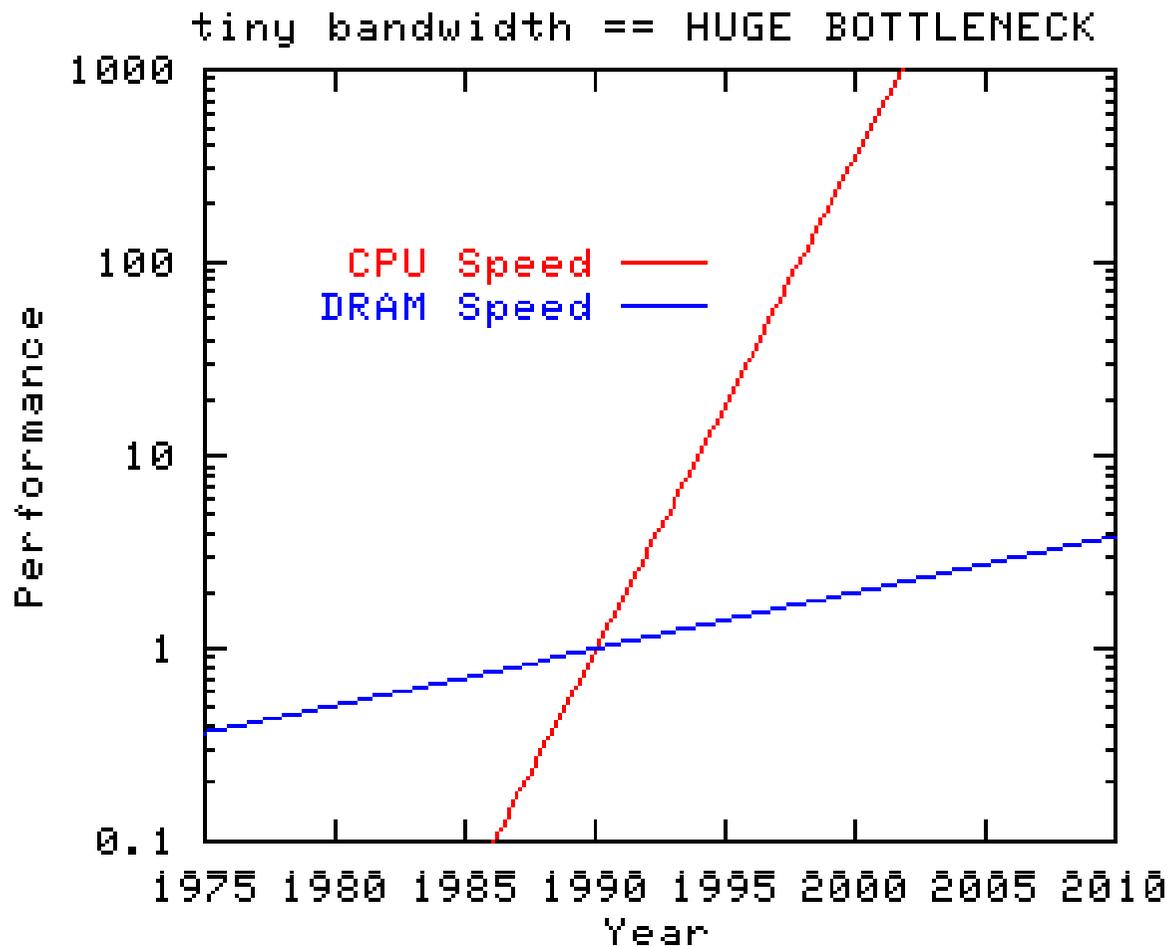
L3 cache: 0.5MB/core



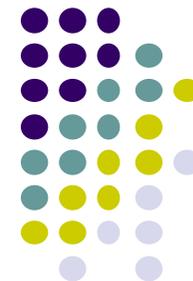


マイクロプロセッサの動向

CPU性能, メモリバンド幅のギャップ

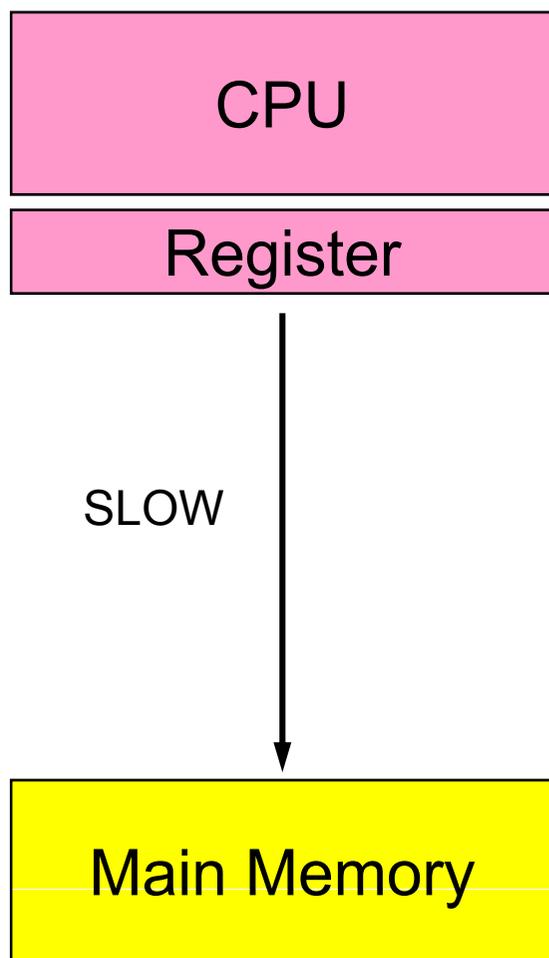


<http://www.streambench.org/>



スカラープロセッサ

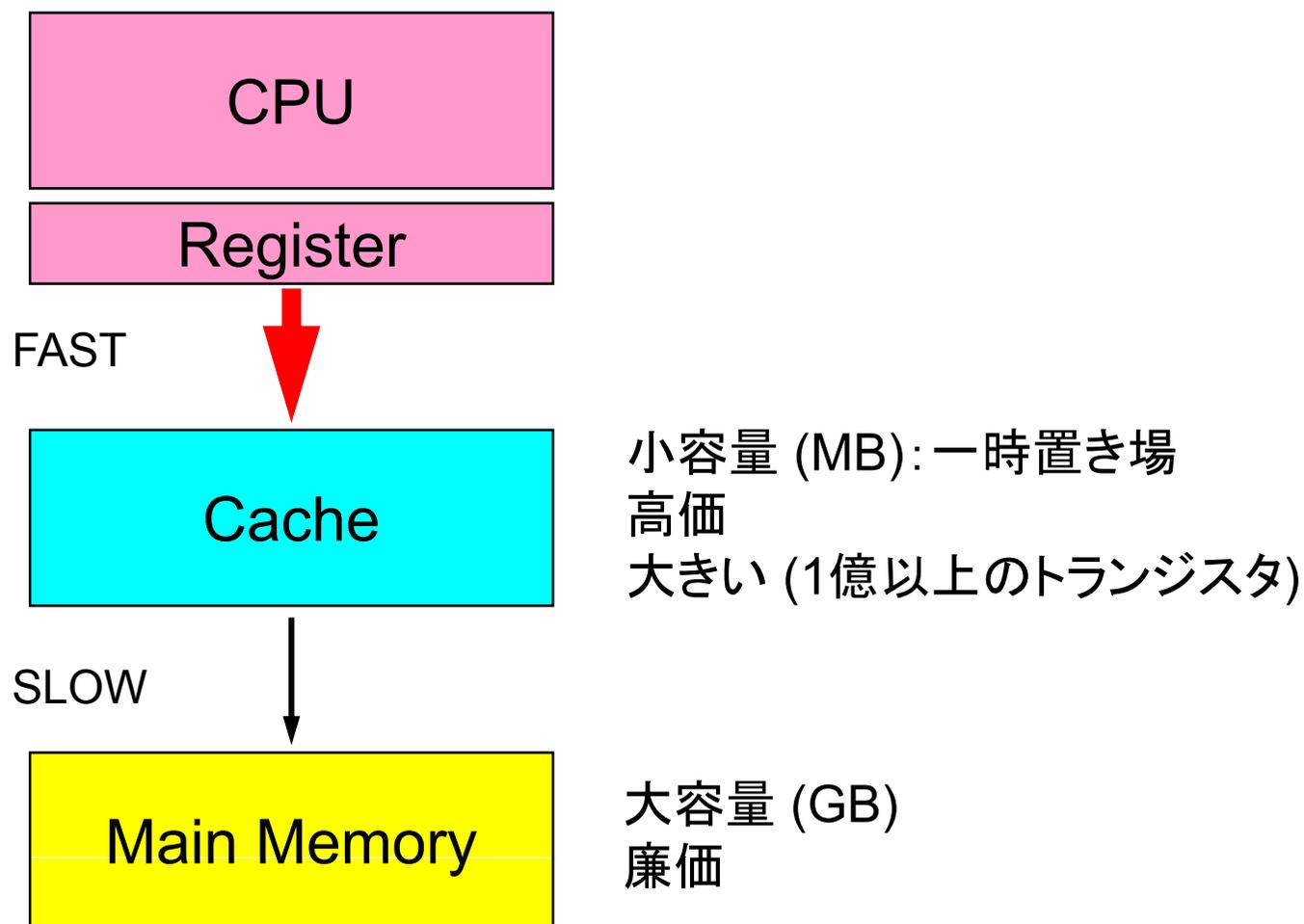
メモリへ直接アクセスするのは実際的でない





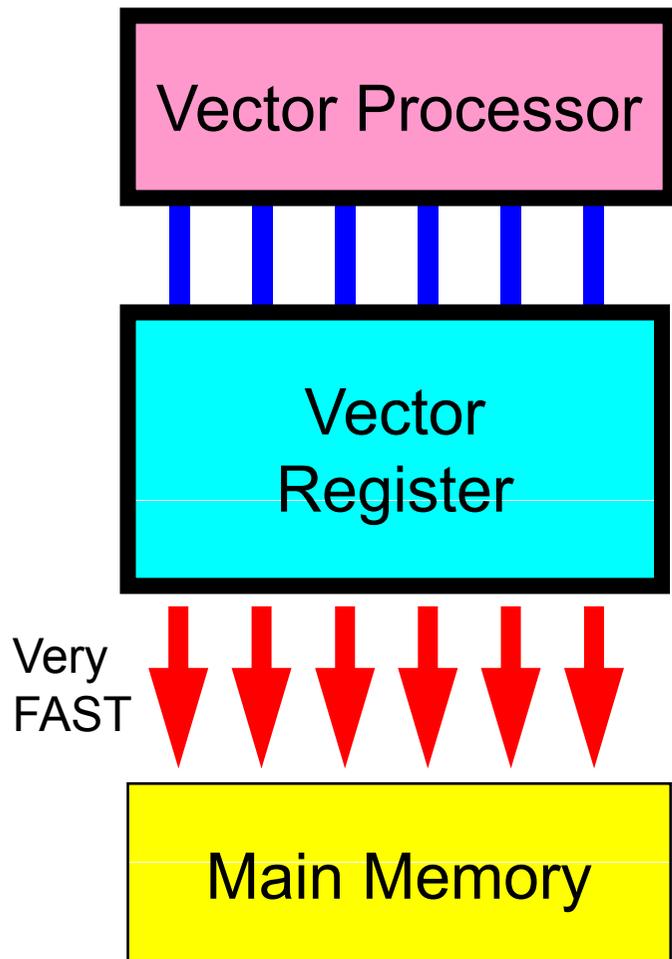
スカラープロセッサ

CPU-キャッシュ-メモリの階層構造



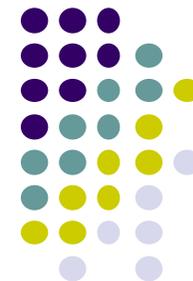
ベクトルプロセッサ

ベクトルレジスタと高速メモリ

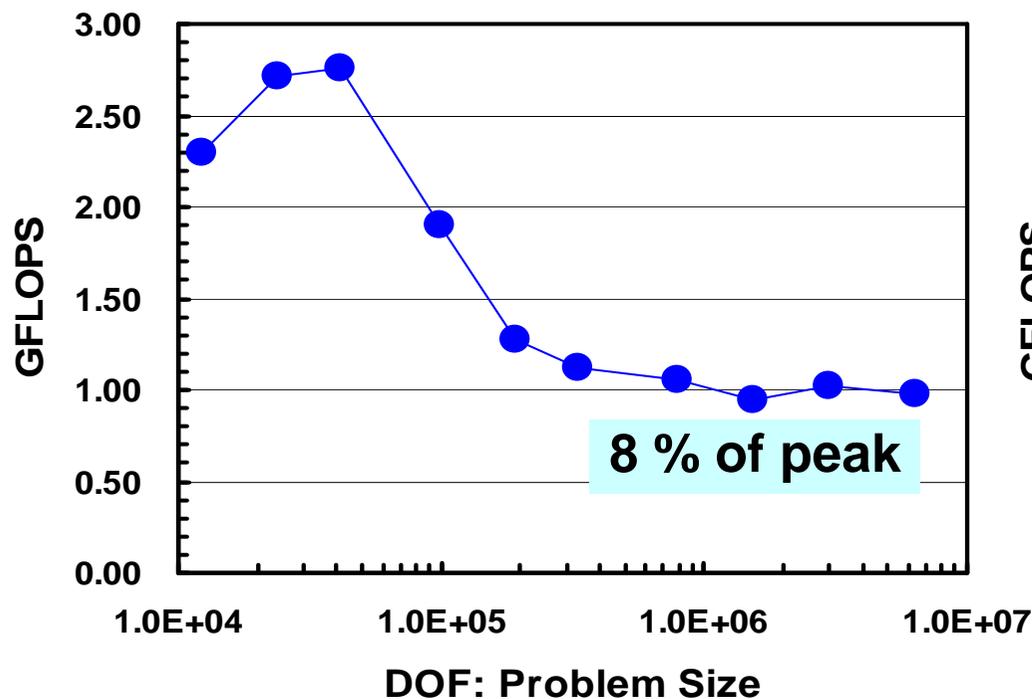


- 単純構造のDOループの並列処理
- 単純, 大規模な演算に適している

```
do i= 1, N
  A(i)= B(i) + C(i)
enddo
```

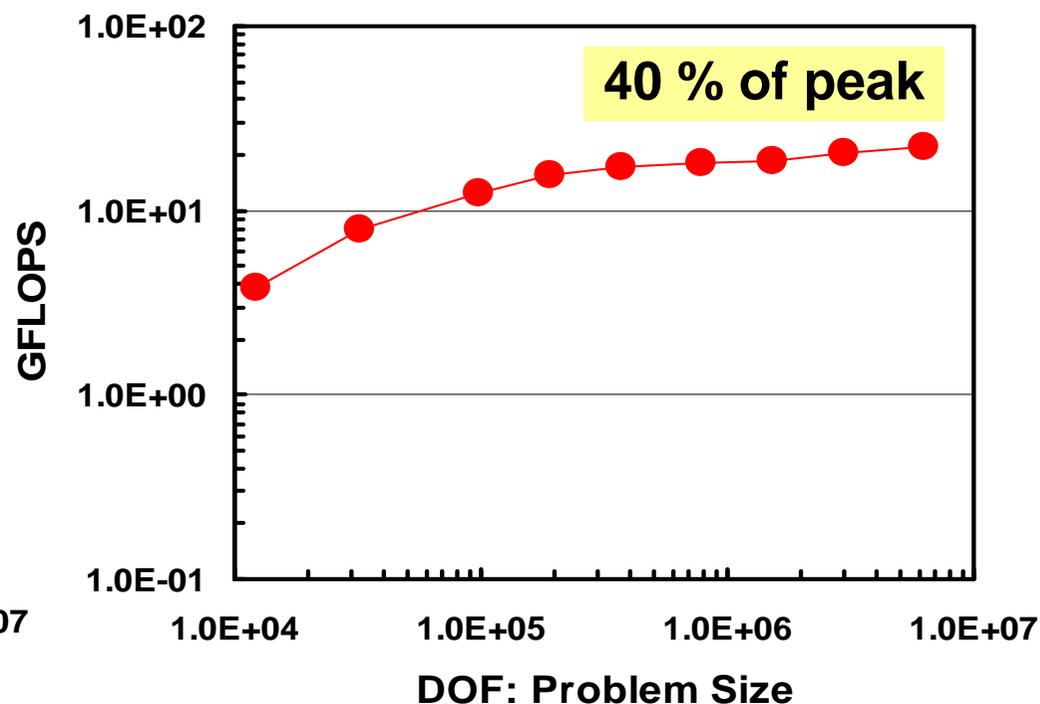


典型的な挙動:ICCG法



IBM-SP3:

問題サイズが小さい場合はキャッシュの影響のため性能が良い



Earth Simulator:

大規模な問題ほどベクトル長が長くなり、性能が高い

専用計算機

- GRAPE (GRAvity PipE)
 - 宇宙物理学におけるN対N問題用専用ハードウェア: MD, 境界要素法等
 - コストパフォーマンス
- MDGRAPE
 - MD (Molecular Dynamics) 専用のGRAPE
 - MDGRAPEを通常のクラスタのAcceleratorのように使用することもできる。

様々な技術的課題

- コアの性能→今後10年間でそれほど変化無し
- 電力消費量
 - Exascale(2018年頃)だと2GW(電気代2,000億円/年)
 - これを20MWくらいまでに抑えるための技術革新必要
 - マルチコア化, メニーコア化
- メモリー性能の低下
 - GPGPU: 高いメモリバンド幅
 - 用途は限られる
- 設置場所
- 故障
 - Exascale: 「億」規模のコア数

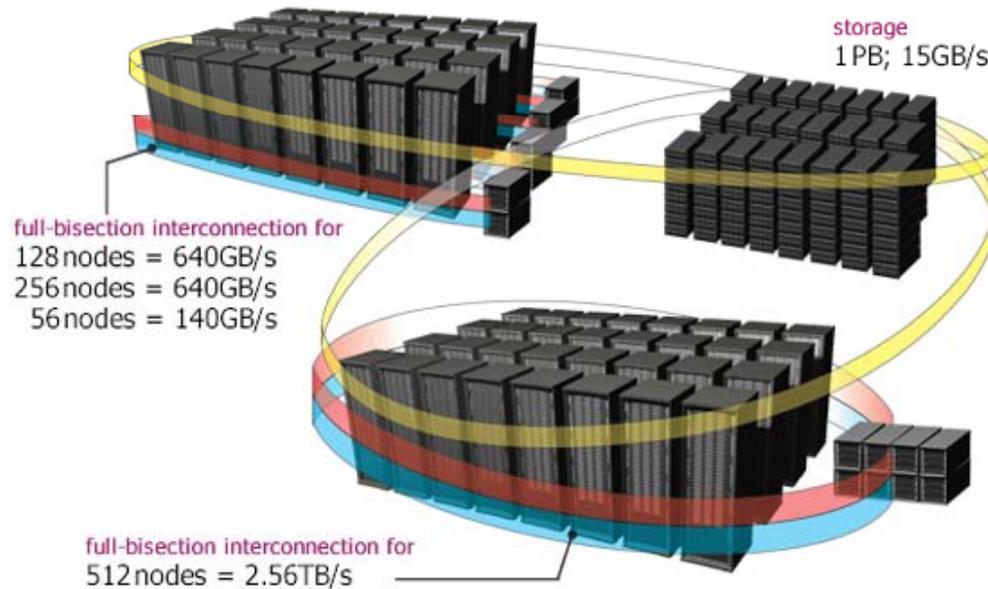


The T2K Open Supercomputer Alliance

<http://www.open-supercomputer.org/>

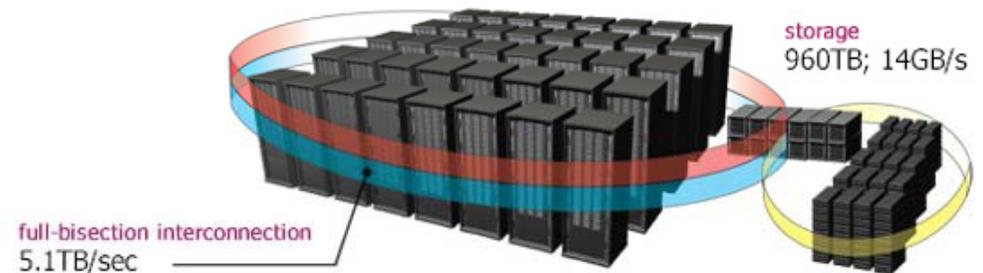
University of Tokyo

nodes = 952 Rpeak = 140.1TFlops Memory = 31TB



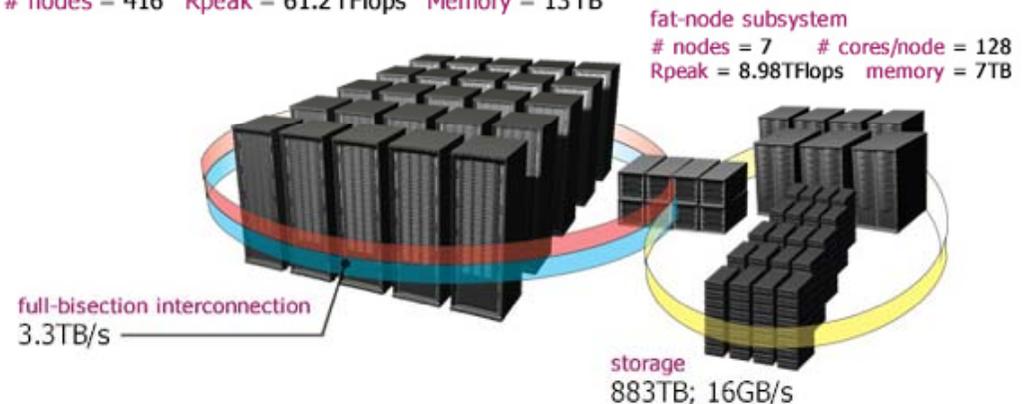
University of Tsukuba

nodes = 648 Rpeak = 95.4TFlops Memory = 20TB



Kyoto University

nodes = 416 Rpeak = 61.2TFlops Memory = 13TB

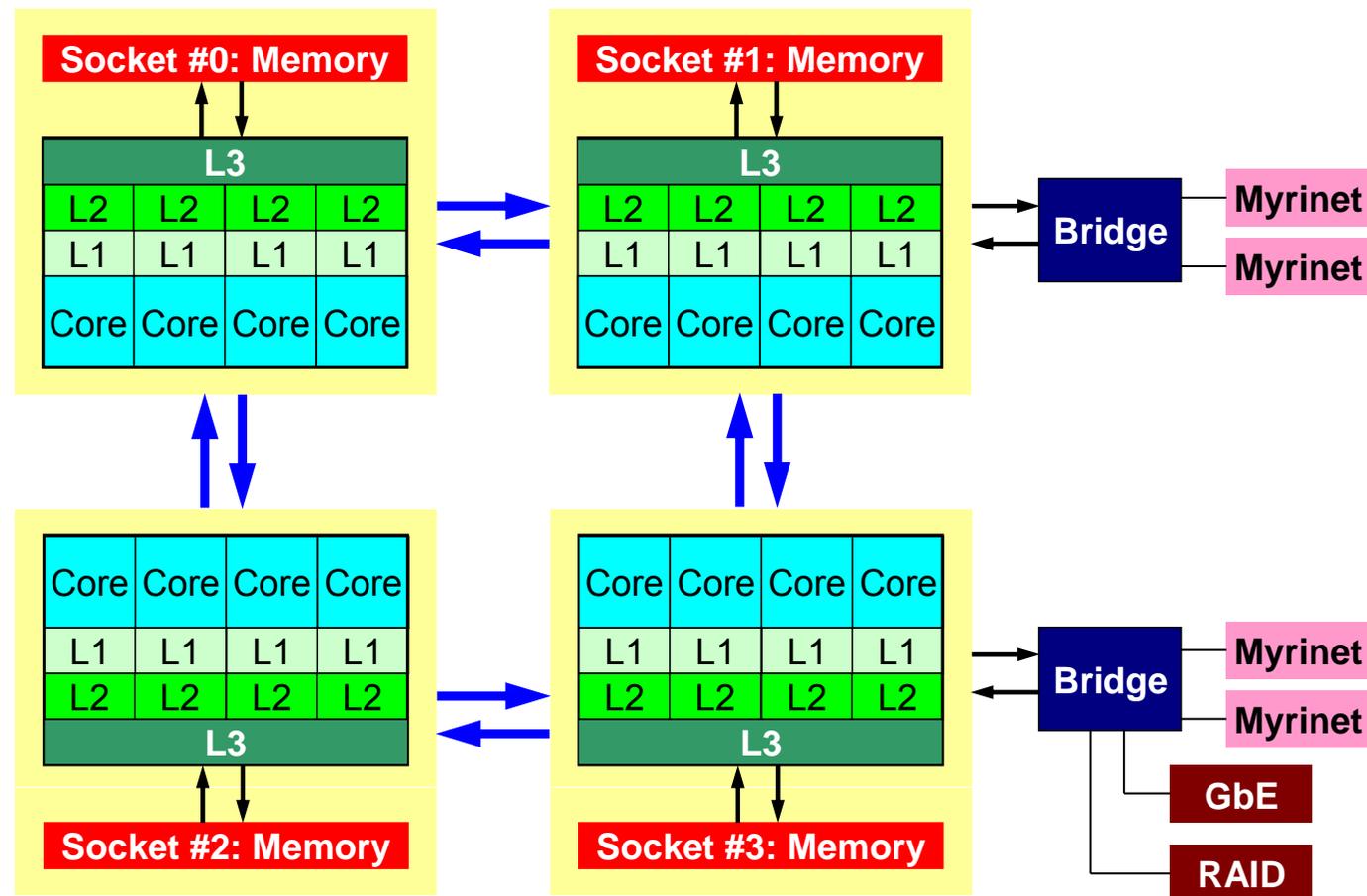


Tsukuba: Appro-Cray
Tokyo: Hitachi
Kyoyo: Fujitsu

The T2K Open Supercomputer Alliance

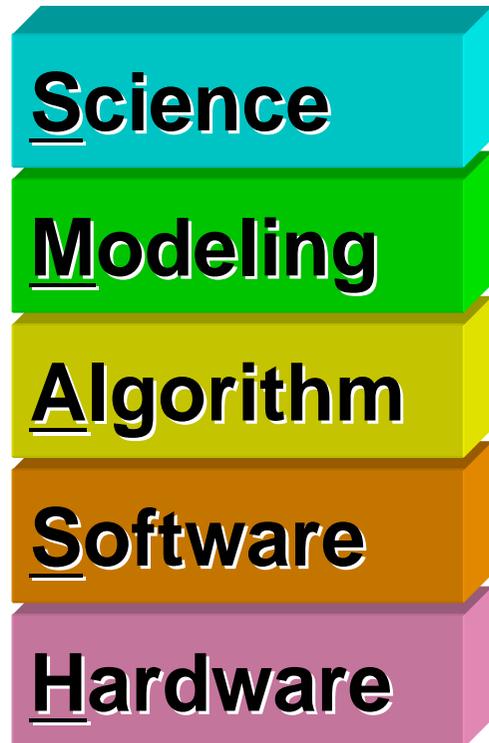
<http://www.open-supercomputer.org/>

各ノード: Quad-core Opteron (Barcelona) × 4



本講義の目的(2/3)

- 本講義・実習は「東京大学 学際計算科学・工学 人材育成プログラム」の一環として実施され,
- 科学技術計算プログラミングに必須の項目である「SMASH (Science-Modeling-Algorithm-Software-Hardware)」を, できるだけ幅広くカバーし, 広い視野を持った人材を育成することを目標とするものです。



本講義の目的(3/3)

- 並列計算機を使用した大規模シミュレーションの実施のためには、科学・工学と計算機科学、数値アルゴリズム(数理科学)の専門家の密接な協力が必要となります。
 - そのためには、数理科学、計算機科学の専門家もある程度アプリケーションに関する知識と経験が要求されます。
 - 本講義は、情報理工学系研究科の学生がアプリケーションに関する知識を効率的に得るのにも適しています。

担当教員

- 中島研吾
 - 情報基盤センタースーパーコンピューティング研究部門
 - 大学院情報理工学系研究科数理情報学専攻兼担
 - 工学部航空学科（現航空宇宙工学科）卒
 - （FY.2004～2007）理学系研究科地球惑星科学専攻
- 専門
 - 計算力学，数値流体力学
 - 数値線形代数
 - 並列プログラミングモデル，並列アルゴリズム
- 情報基盤センター別館3F，[内線：22719](tel:22719)
- nakajima@cc.u-tokyo.ac.jp
- 質問等は随時，e-mail・電話で所在確認
- <http://nkl.cc.u-tokyo.ac.jp/11w/>

準備等

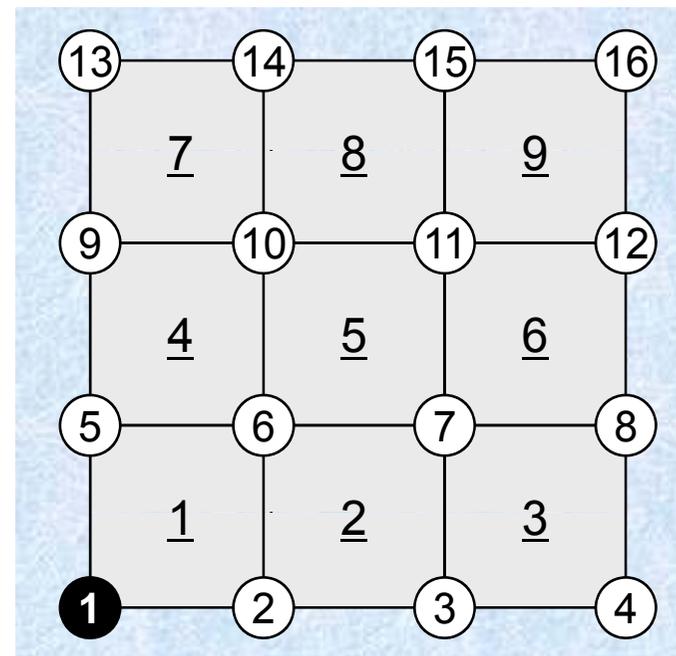
- 受講に必要な知識等
 - UNIXの操作
 - FORTRANまたはCによる基礎的なプログラミングの経験
 - 基礎的な数値解析に関する知識
- 情報基盤センター教育用計算機システムのアカウント
 - <http://www.ecc.u-tokyo.ac.jp/doc/announce/newuser.html>
- 「科学技術計算 I（有限要素法）」を履修していること、または同等の知識を有すること
 - <http://nkl.cc.u-tokyo.ac.jp/11s/>
- その他、連絡事項については随時HPに掲載する
 - <http://nkl.cc.u-tokyo.ac.jp/11w/>

Finite-Element Method (FEM)

- 偏微分方程式の解法として広く知られている
 - elements (meshes, 要素) & nodes (vertices, 節点)
- 以下の二次元熱伝導問題を考える:

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

- 16節点, 9要素 (四角形)
- 一様な熱伝導率 ($\lambda=1$)
- 一様な体積発熱 ($Q=1$)
- 節点1で温度固定: $T=0$
- 周囲断熱



Galerkin FEM procedures

- 各要素にガラーキン法を適用:

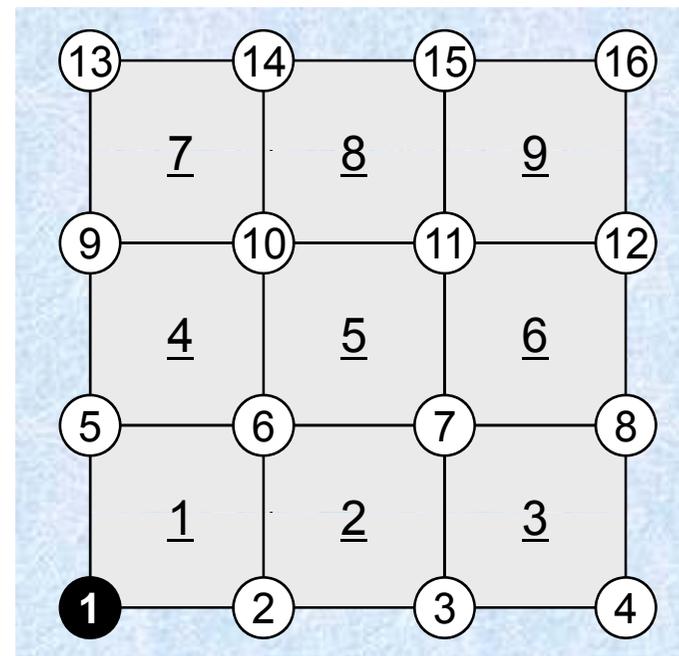
$$\int_V [N]^T \left\{ \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q \right\} dV = 0$$

各要素で: $T = [N]\{\phi\}$
 $[N]$: 形状関数(内挿関数)

- 偏微分方程式に対して, ガウス・グリーンの定理を適用し, 以下の「弱形式」を導く

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

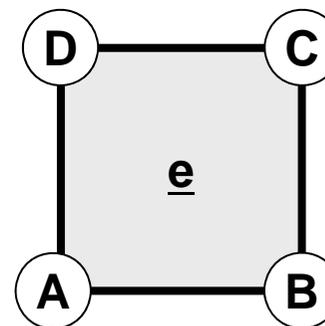
$$+ \int_V Q [N]^T dV = 0$$



Element Matrix : 要素マトリクス

- 各要素において積分を実行し，要素マトリクスを得る

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

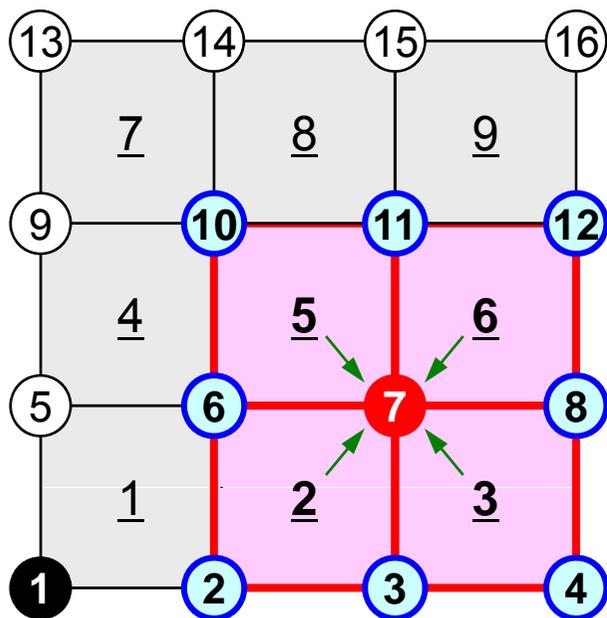


$$[k^{(e)}] \{\phi^{(e)}\} = \{f^{(e)}\}$$

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

Global/overall Matrix : 全体マトリクス

各要素マトリクスを全体マトリクスに足しこむ

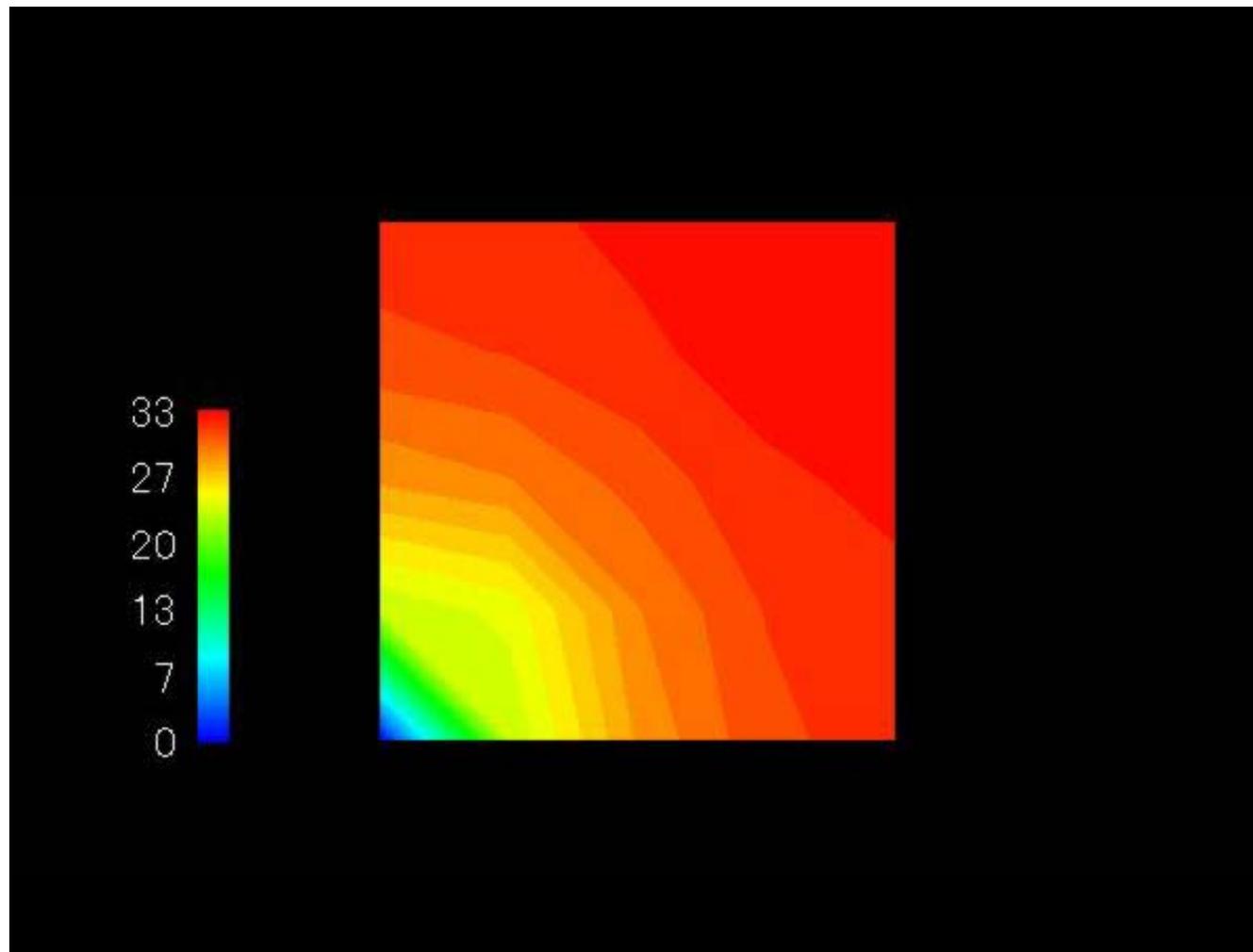
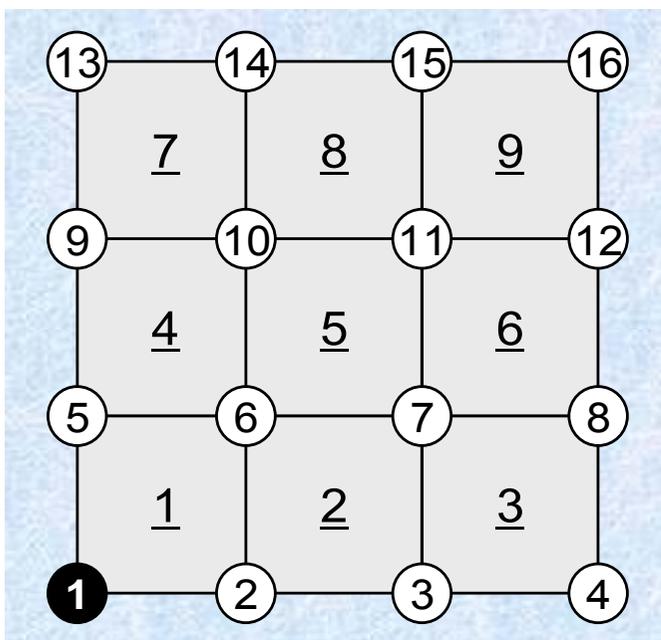


$$[K]\{\Phi\} = \{F\}$$

D	X			X	X															Φ_1	F_1
X	D	X		X	X	X														Φ_2	F_2
	X	D	X		X	X	X													Φ_3	F_3
		X	D			X	X													Φ_4	F_4
X	X			D	X			X	X											Φ_5	F_5
X	X	X		X	D	X		X	X	X										Φ_6	F_6
	X	X		X	D	X		X	X	X										Φ_7	F_7
								X	D			X	X							Φ_8	F_8
				X	X				D	X				X	X					Φ_9	F_9
				X	X	X			X	D	X			X	X	X				Φ_{10}	F_{10}
					X	X	X			X	D	X		X	X	X				Φ_{11}	F_{11}
						X	X				X	D			X	X				Φ_{12}	F_{12}
									X	X		D	X							Φ_{13}	F_{13}
									X	X	X		X	D	X					Φ_{14}	F_{14}
										X	X			X	D	X				Φ_{15}	F_{15}
															X	D				Φ_{16}	F_{16}

計算結果

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



年	月日	内 容
2010年	10月04日 (火)	はじめに
	10月11日 (火)	復習, 並列データ構造
	10月18日 (火)	並列データ構造 (続き), T2Kオーpensパソコン
	10月25日 (火)	(休講)
	11月01日 (火)	MPIによる並列プログラミング (1)
	11月08日 (火)	MPIによる並列プログラミング (2)
	11月15日 (火)	(休講)
	11月22日 (火)	チューニング入門
	11月29日 (火)	MPIによる並列プログラミング (3)
	12月06日 (火)	MPIによる並列プログラミング (4)
	12月13日 (火)	課題S1解説
	12月20日 (火)	課題S2解説
2011年	01月17日 (火)	三次元並列有限要素法 (1)
	01月24日 (火)	(休講)
	01月31日 (火)	三次元並列有限要素法 (2)
	02月07日 (火)	三次元並列有限要素法 (3)
	02月14日 (火)	最近の話題

講義の進め方

- 夏学期の復習は随時実施する
- MPIの既習者？
- 講義資料
 - 月曜正午までにWEBにアップ
 - 各自印刷のこと(配布はしない:今日は特別)
- 評価
 - レポート

- はじめに
- **MPI**とは
- 並列有限要素法とは？

MPIとは (1/2)

- Message Passing Interface
- 分散メモリ間のメッセージ通信APIの「規格」
 - プログラム, ライブラリ, そのものではない
 - <http://phase.hpcc.jp/phase/mpi-j/ml/mpi-j-html/contents.html>
- 歴史
 - 1992 MPIフォーラム
 - 1994 MPI-1規格
 - 1997 MPI-2規格(拡張版), 現在はMPI-3が検討されている
- 実装
 - mpich アルゴンヌ国立研究所
 - LAM
 - 各ベンダー
 - C/C++, FORTRAN, Java ; Unix, Linux, Windows, Mac OS

MPIとは (2/2)

- 現状では, mpich (フリー) が広く使用されている。
 - 部分的に「MPI-2」規格をサポート
 - 2005年11月から「MPICH2」に移行
 - <http://www-unix.mcs.anl.gov/mpi/>
- MPIが普及した理由
 - MPIフォーラムによる規格統一
 - どんな計算機でも動く
 - FORTRAN, Cからサブルーチン, 関数として呼び出すことが可能
 - mpichの存在
 - フリー, あらゆるアーキテクチャをサポート
- 同様の試みとしてPVM (Parallel Virtual Machine) があつたが, こちらはそれほど広がらず

参考文献

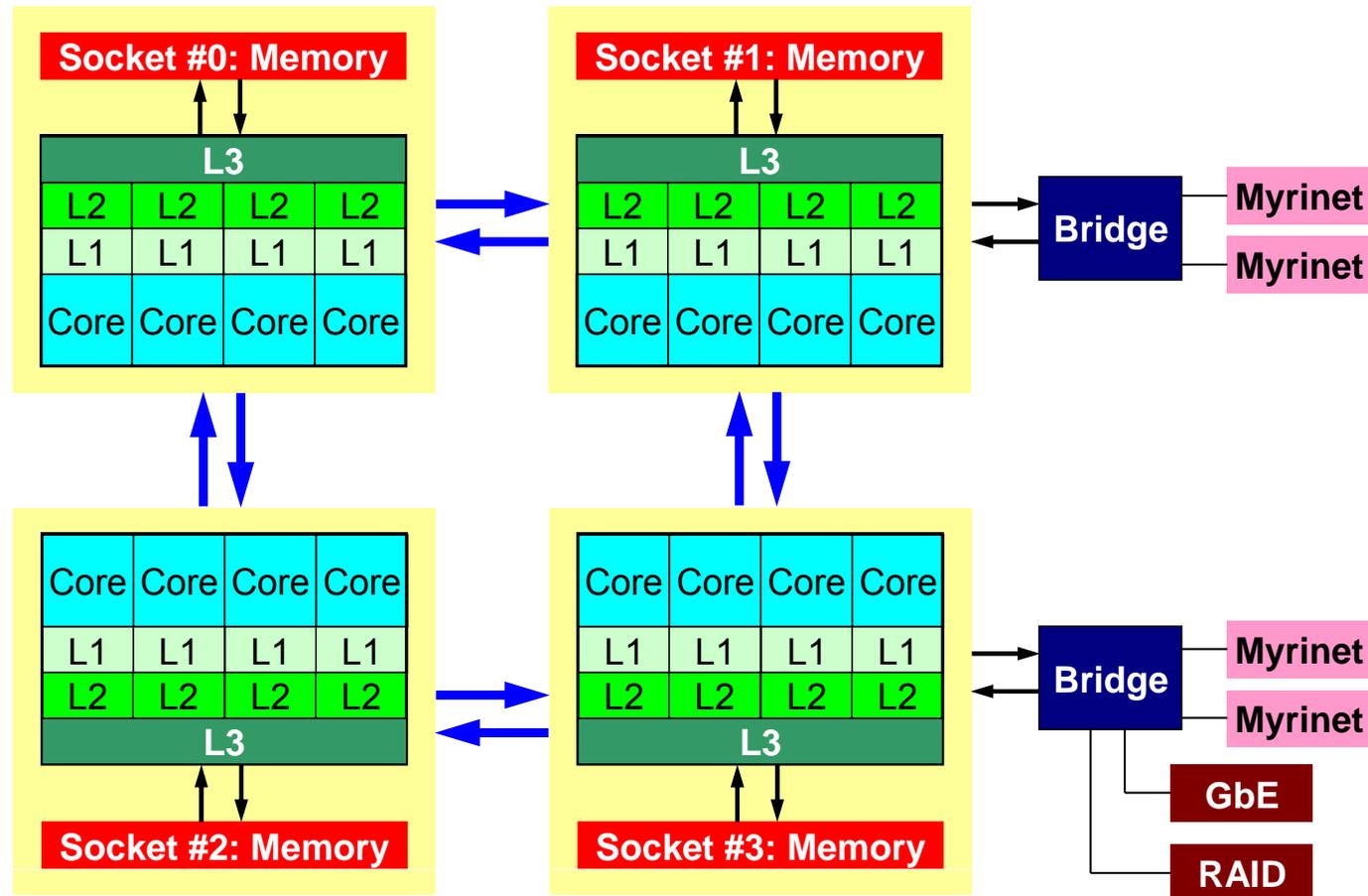
- P.Pacheco「MPI並列プログラミング」, 培風館, 2001(原著1997)
- W.Gropp他「Using MPI second edition」, MIT Press, 1999.
- M.J.Quinn「Parallel Programming in C with MPI and OpenMP」, McGrawhill, 2003.
- W.Gropp他「MPI: The Complete Reference Vol.I, II」, MIT Press, 1998.
- <http://www-unix.mcs.anl.gov/mpi/www/>
 - API(Application Interface)の説明

MPIを学ぶにあたって

- 文法
 - 「MPI-1」の基本的な機能(10程度)について習熟する
 - MPI-2では色々と便利な機能があるが...
 - あとは自分に必要な機能について調べる, あるいは知っている人, 知っていそうな人に尋ねる
- 実習の重要性
 - プログラミング
 - その前にまず実行してみること
- SPMD/SIMDのオペレーションに慣れること...「つかむ」こと
 - Single Program/Instruction Multiple Data
 - 基本的に各プロセスは「同じことをやる」が「データが違う」
 - 大規模なデータを分割し, 各部分について各プロセス(プロセッサ)が計算する
 - 全体データと局所データ, 全体番号と局所番号

T2Kオープンスパコン(東大)

1コア=1領域



- はじめに
- MPIとは
- 並列有限要素法とは？
 - 基本的な考え方
 - 局所データ構造

並列計算の目的

- 高速, 大規模

- 「大規模」の方が「新しい科学」という観点からのウェイトとしては高い。しかし, 「高速」ももちろん重要である。

- 細かいメッシュ

- +複雑

- 理想: Scalable

- N倍の規模の計算をN倍のCPUを使って, 「同じ時間で」解く
(大規模性の追求: Weak Scaling)

- 実際はそうは行かない

- 例: 共役勾配法⇒問題規模が大きくなると反復回数が増加

- 同じ問題をN倍のCPUを使って「1/Nの時間で」解く・・・という場合もある(高速性の追求: Strong Scaling)

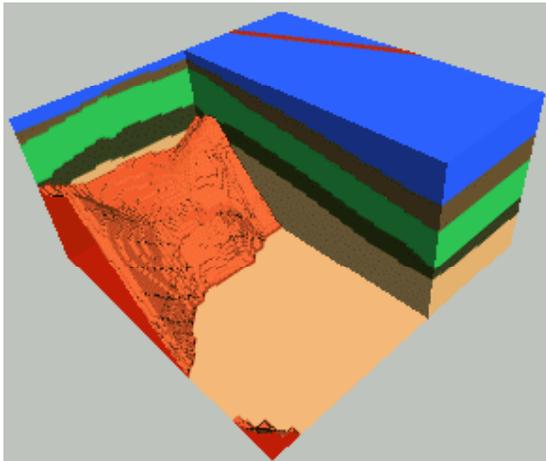
- これも余り簡単な話では無い

並列計算とは？(1/2)

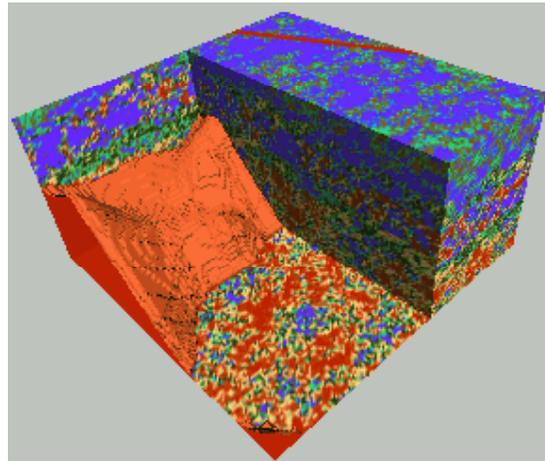
- より大規模で複雑な問題を高速に解きたい

Homogeneous/Heterogeneous Porous Media

Lawrence Livermore National Laboratory



Homogeneous

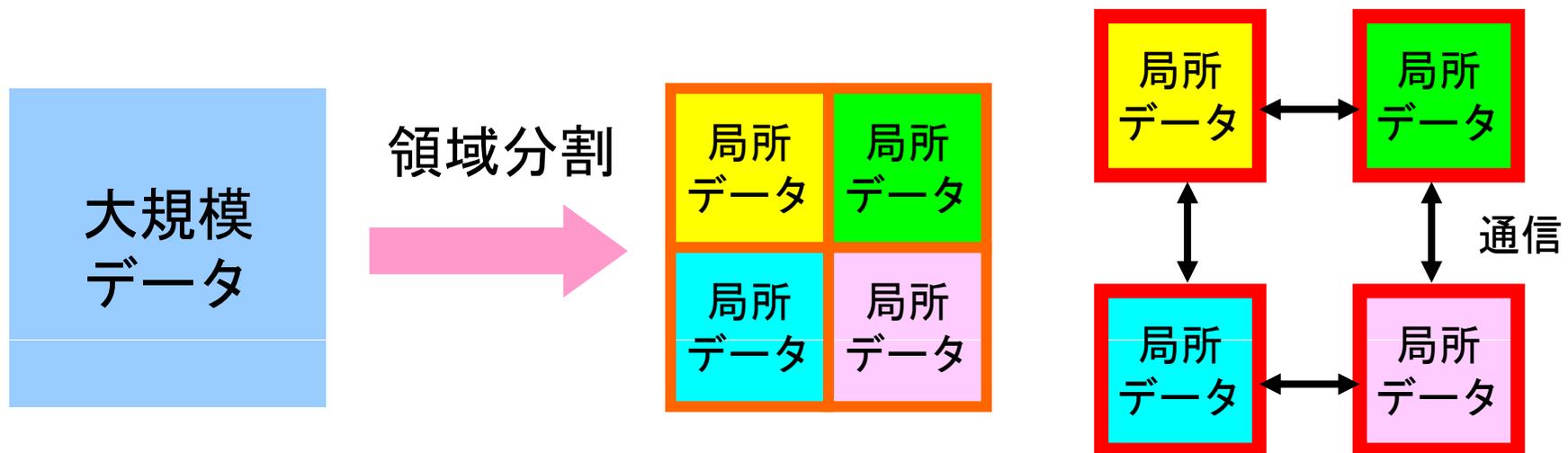


Heterogeneous

このように不均質な場を模擬するには非常に細かいメッシュが必要となる

並列計算とは？(2/2)

- 1GB程度のPC → $<10^6$ メッシュが限界:FEM
 - 1000km × 1000km × 100kmの領域(西南日本)を1kmメッシュで切ると 10^8 メッシュになる
- 大規模データ → 領域分割, 局所データ並列処理
- 全体系計算 → 領域間の通信が必要



通信とは？

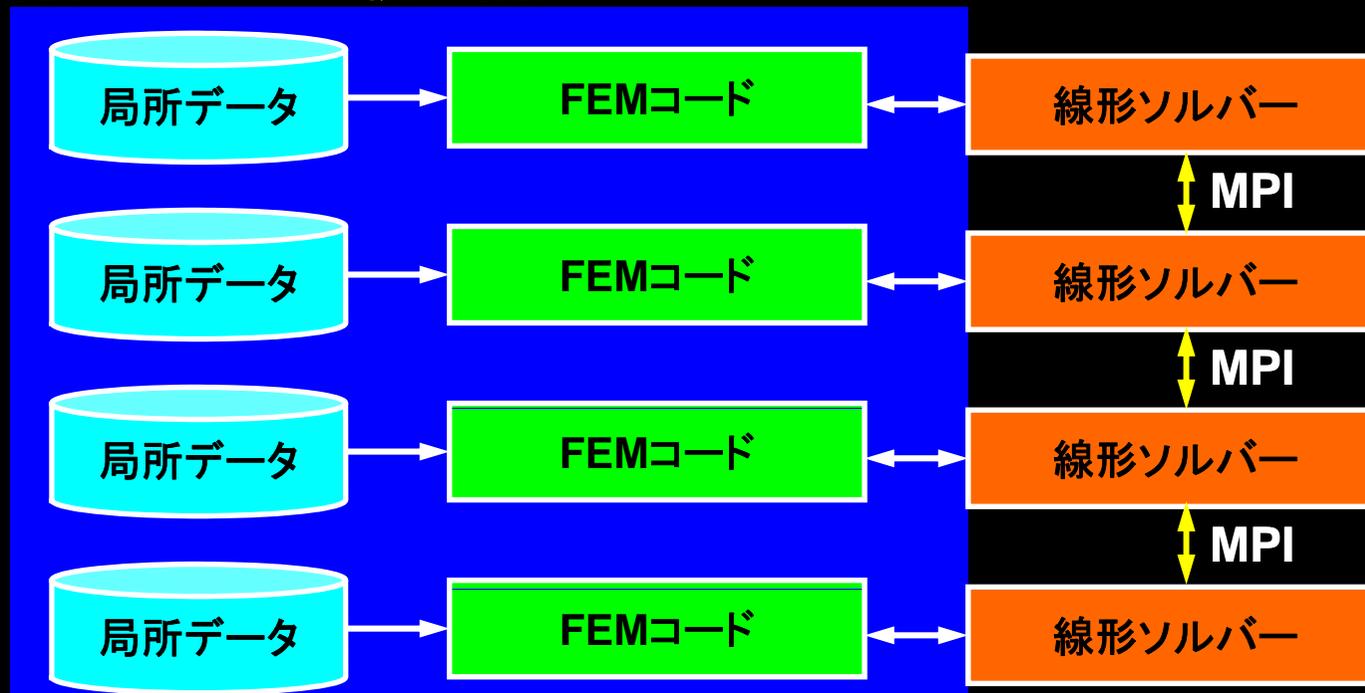
- 並列計算とはデータを処理をできるだけ「局所的 (local)」に実施すること。
 - 要素マトリクスの計算
 - 有限要素法の計算は本来並列計算向けである
- 「大域的 (global)」な情報を必要とする場合に通信が生じる (必要となる)。
 - 全体マトリクスを線形ソルバーで解く

並列有限要素法の処理: SPMD

巨大な解析対象 → 局所分散データ, 領域分割 (Partitioning)

有限要素コードは領域ごとに係数マトリクスを生成: 要素単位の処理によって可能: シリアルコードと変わらない処理

グローバル処理, 通信は線形ソルバーのみで生じる
内積, 行列ベクトル積, 前処理



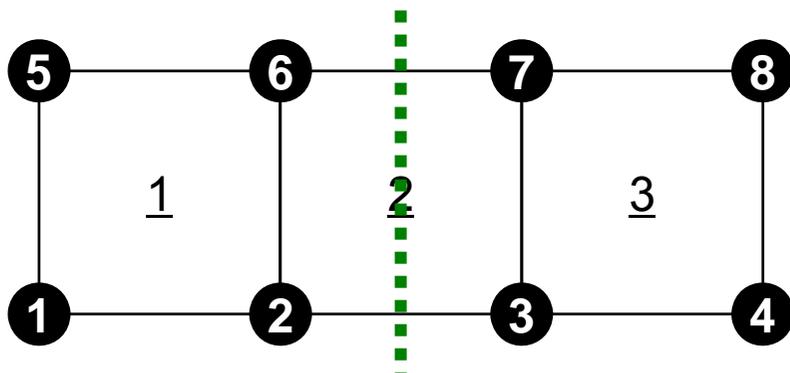
並列有限要素法プログラムの開発

- 前頁のようなオペレーションを実現するためのデータ構造が重要
 - アプリケーションの「並列化」にあたって重要なのは、適切な局所分散データ構造の設計である。
- 前処理付反復法
- マトリクス生成: ローカルに処理

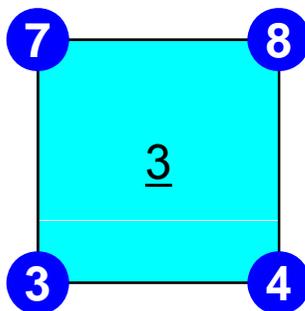
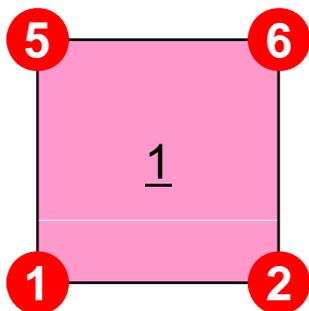
- はじめに
- MPIとは
- 並列有限要素法とは？
 - 基本的な考え方
 - 局所データ構造



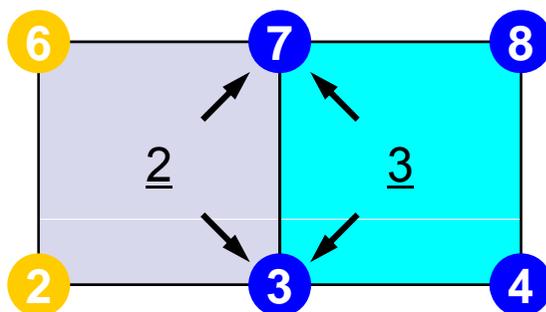
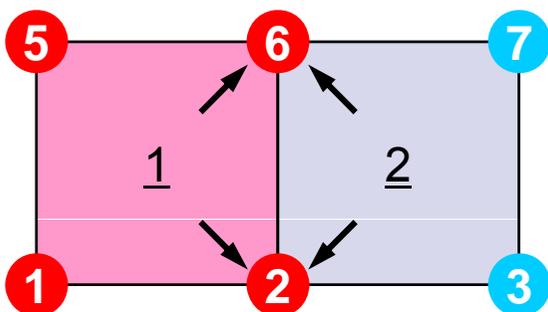
四角形要素



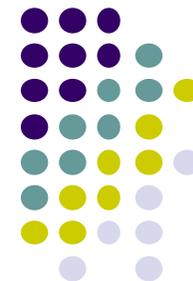
「節点ベース(領域ごとの節点数がバランスする)」の分割
自由度: 節点上で定義



これではマトリクス生成に必要な情報は不十分



マトリクス生成のためには、オーバーラップ部分の要素と節点の情報が必要



並列有限要素法の局所データ構造

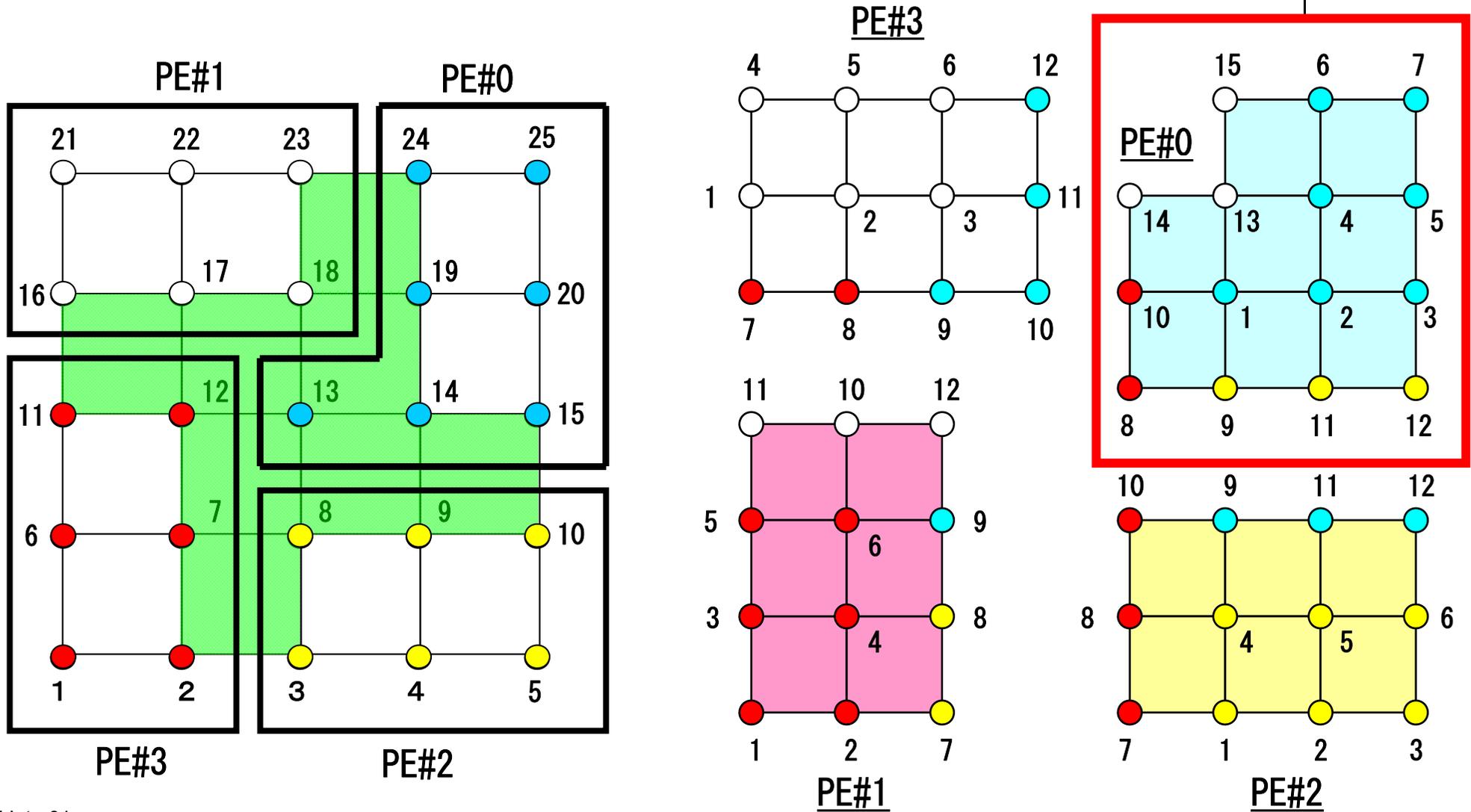
- **節点ベース** : Node-based partitioning
- 局所データに含まれるもの：
 - その領域に本来含まれる節点
 - それらの節点を含む要素
 - 本来領域外であるが、それらの要素に含まれる節点
- 節点は以下の3種類に分類
 - **内点** : Internal nodes その領域に本来含まれる節点
 - **外点** : External nodes 本来領域外であるがマトリクス生成に必要な節点
 - **境界点** : Boundary nodes 他の領域の「外点」となっている節点
- 領域間の通信テーブル

- 領域間の接続をのぞくと、大域的な情報は不要
 - 有限要素法の特徴 : 要素で閉じた計算



Node-based Partitioning

internal nodes - elements - external nodes

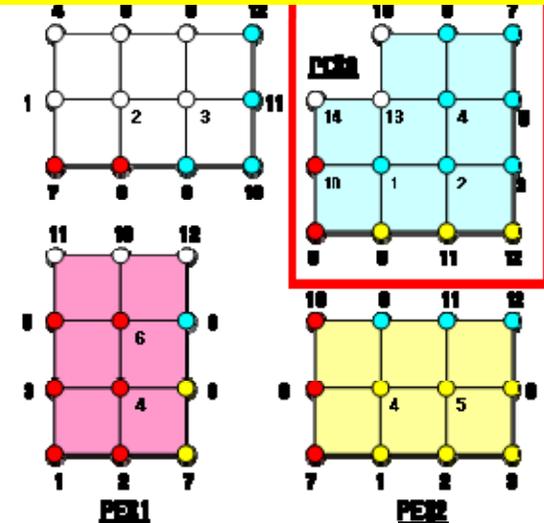
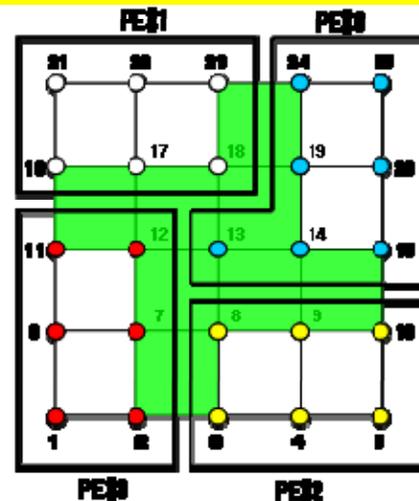
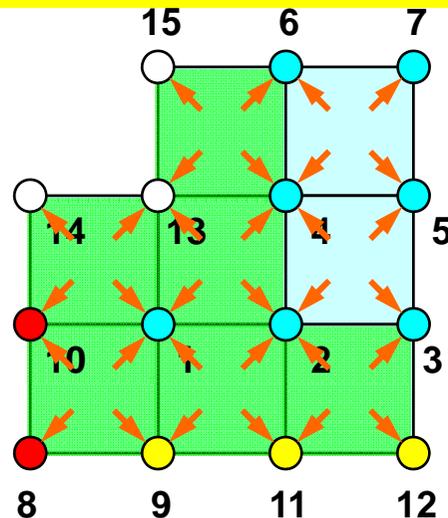




Node-based Partitioning

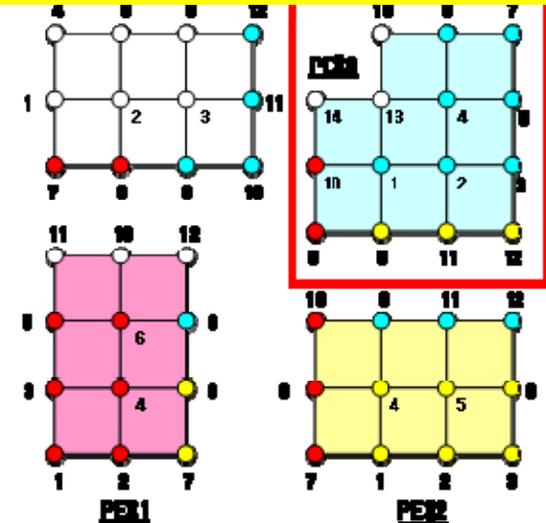
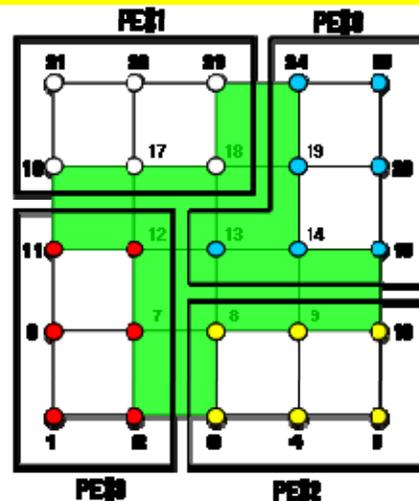
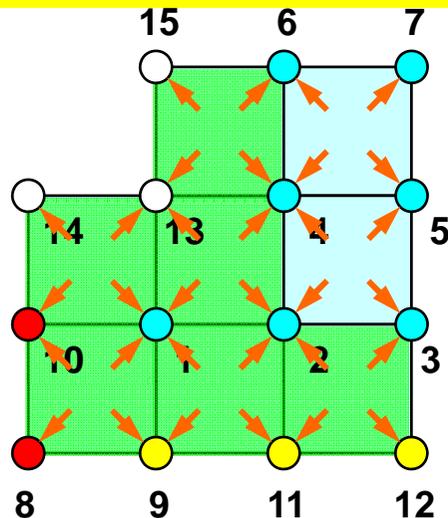
internal nodes - elements - external nodes

- Partitioned nodes themselves (Internal Nodes) 内点
- Elements which include Internal Nodes 内点を含む要素
- External Nodes included in the Elements 外点
in overlapped region among partitions.
- Info of External Nodes are required for completely local element-based operations on each processor.



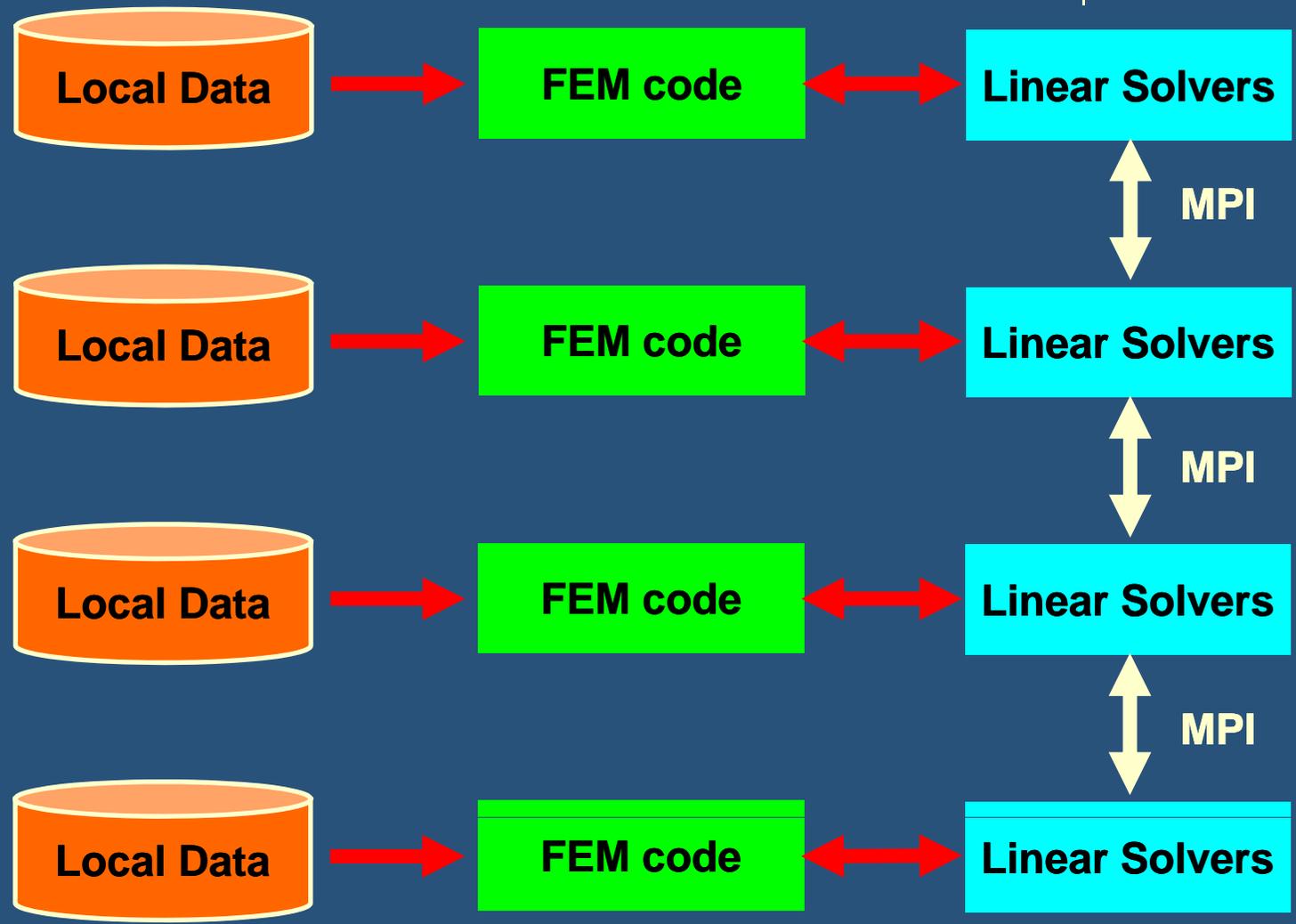
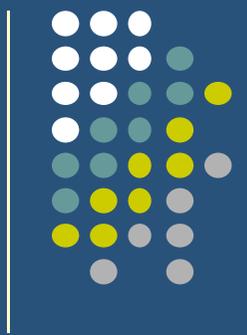
マトリクス生成時の通信は不要

- Partitioned nodes themselves (Internal Nodes) 内点
- Elements which include Internal Nodes 内点を含む要素
- External Nodes included in the Elements 外点
in overlapped region among partitions.
- Info of External Nodes are required for completely local element-based operations on each processor.



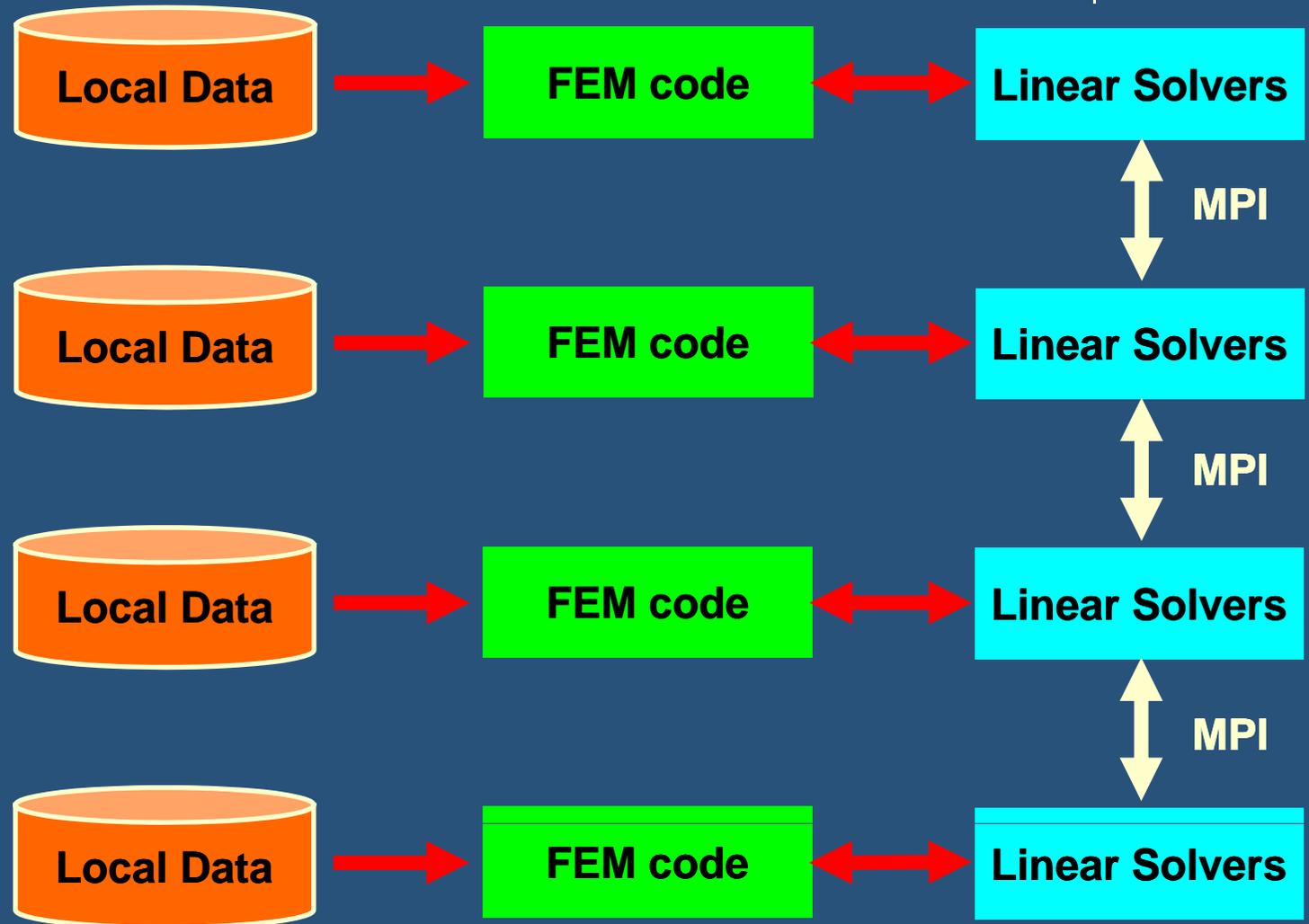
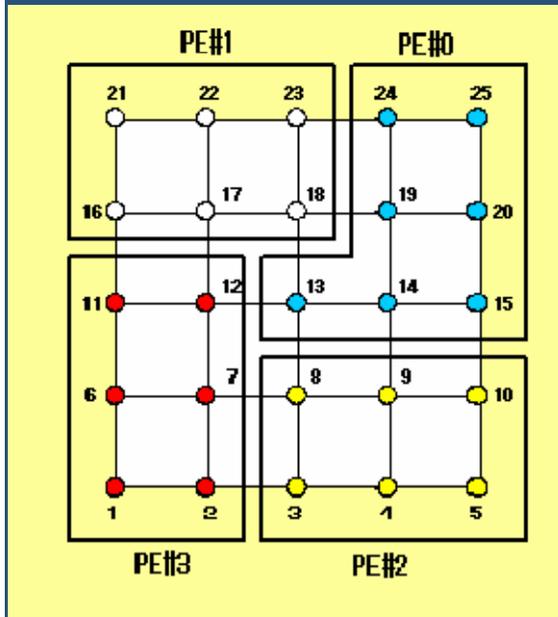
Parallel Computing in FEM

SPMD: Single-Program Multiple-Data



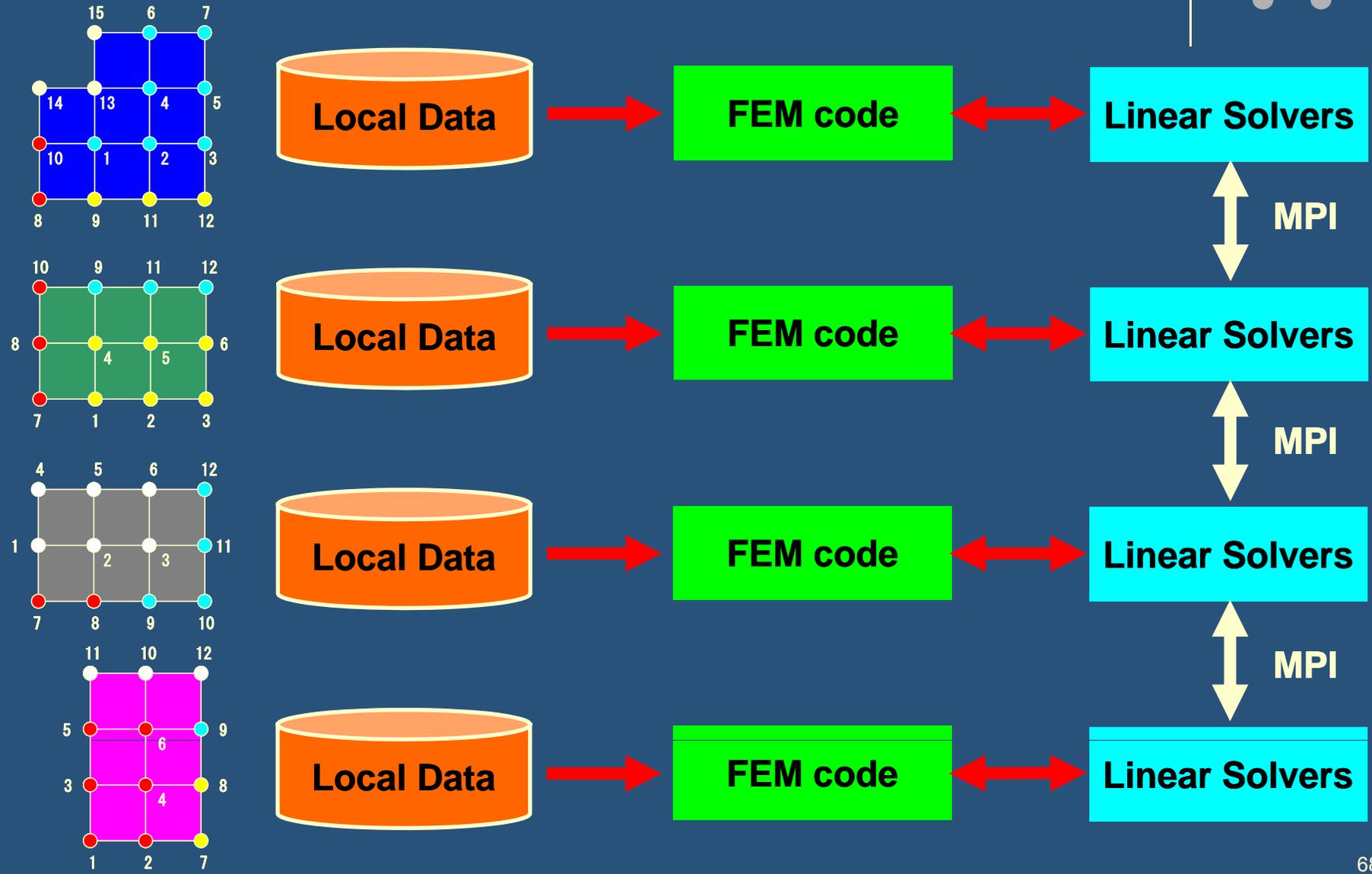
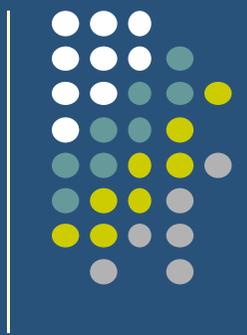
Parallel Computing in FEM

SPMD: Single-Program Multiple-Data



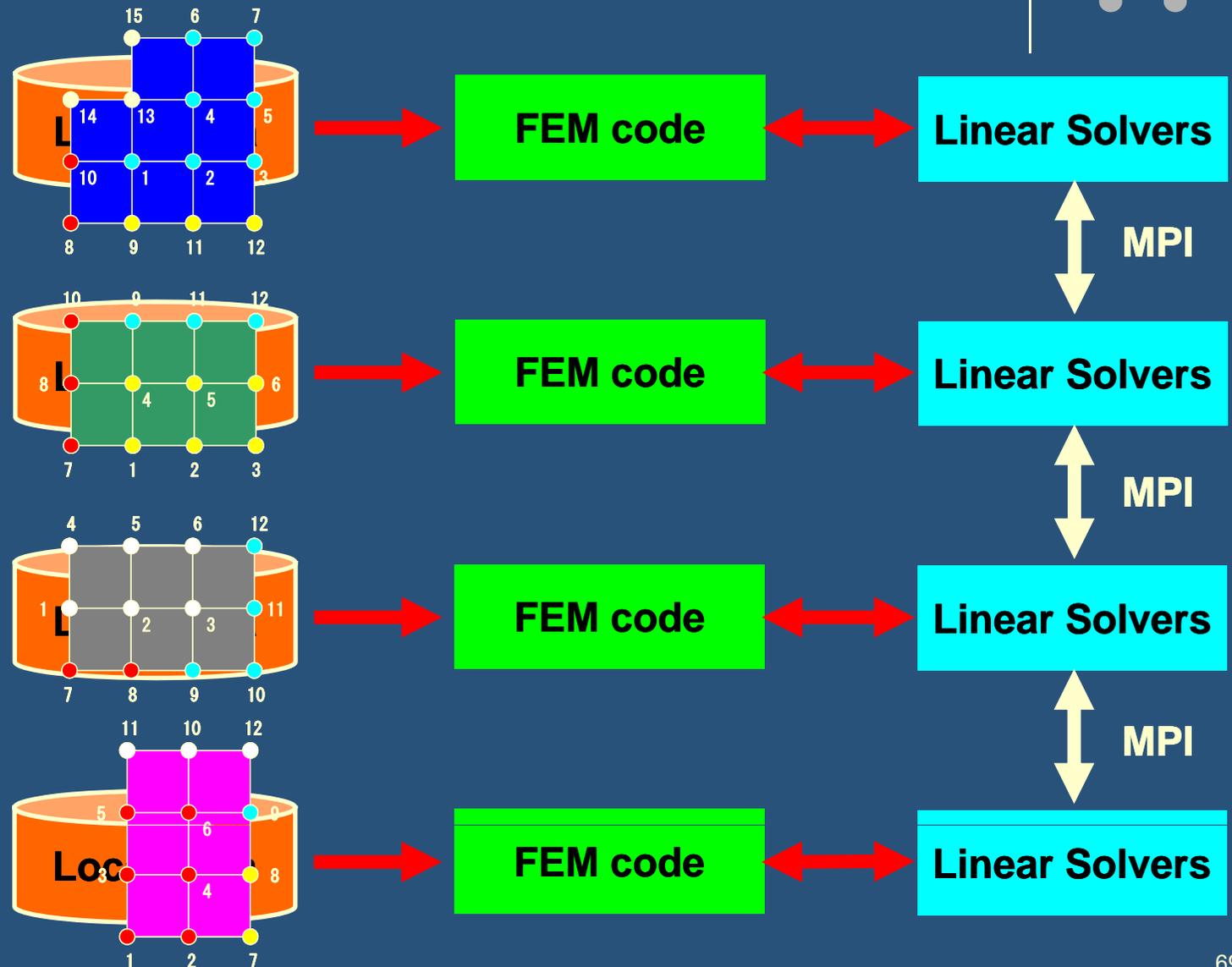
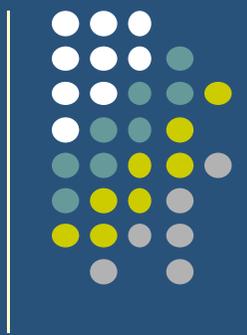
Parallel Computing in FEM

SPMD: Single-Program Multiple-Data



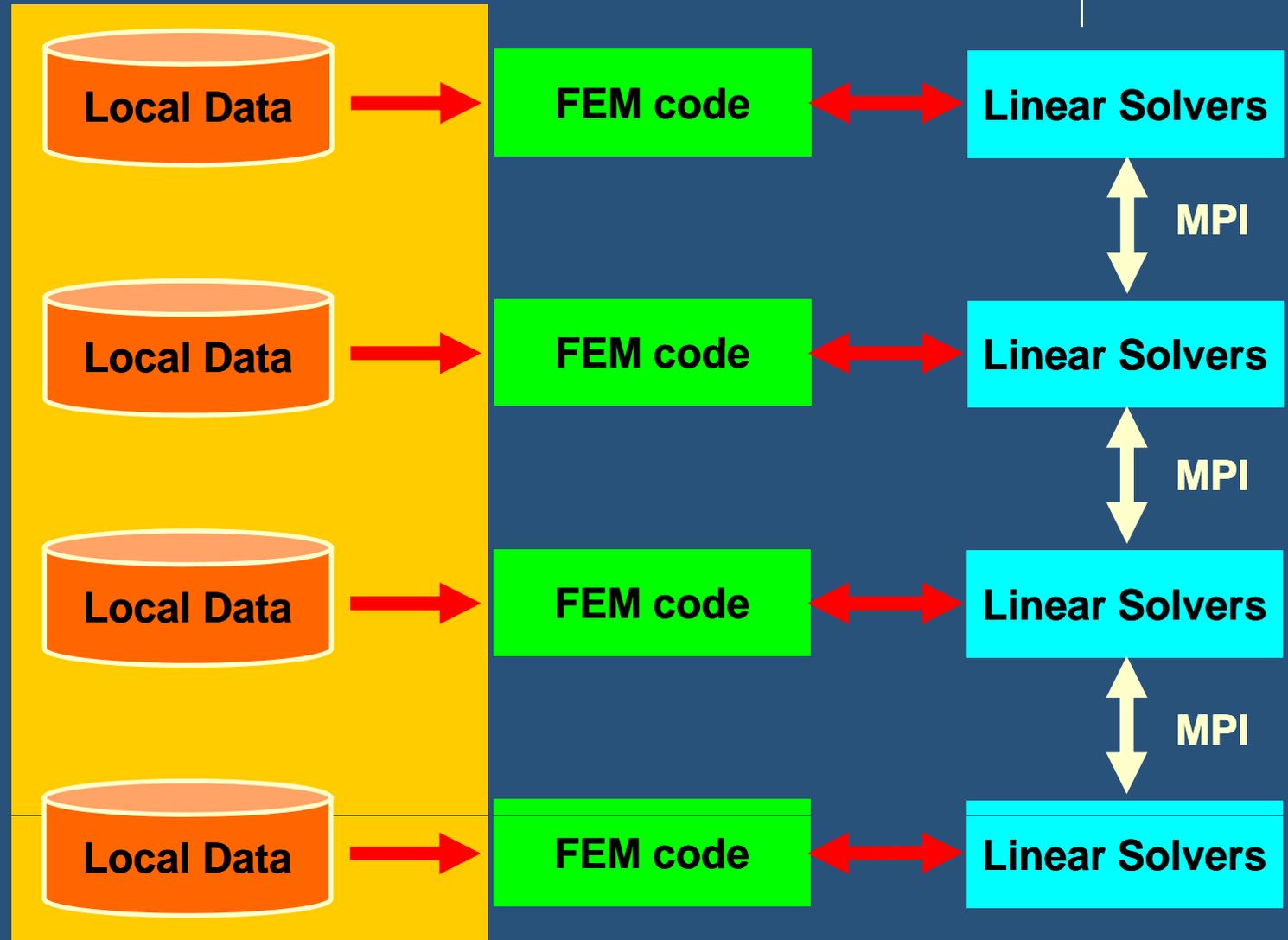
Parallel Computing in FEM

SPMD: Single-Program Multiple-Data

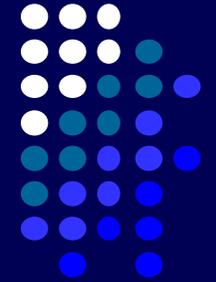


Parallel Computing in FEM

SPMD: Single-Program Multiple-Data



通信とは何か？



- 「外点」の情報を外部の領域からもらってこること
- 「通信テーブル」にその情報が含まれている