

はじめに, 並列有限要素法とは

2012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)

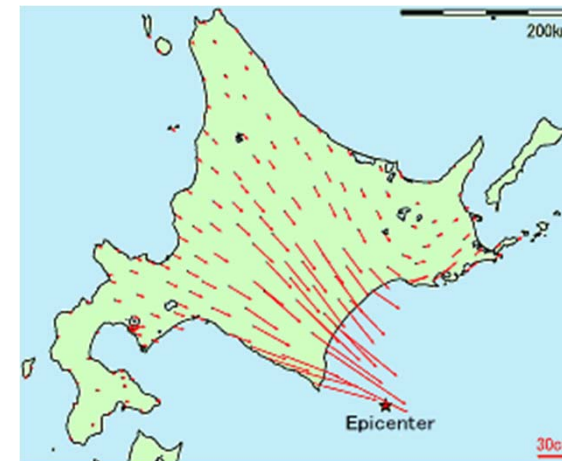
- 科学技術計算Ⅱ(4820-1028)
 - 情報理工学系数理情報学専攻
- コンピュータ科学特別講義Ⅱ(4810-1205)
 - 情報理工学系コンピュータ科学専攻
- 科学技術計算Ⅰ・コンピュータ科学特別講義Ⅰ(有限要素法)(夏学期)に引き続き以下の講義, 実習を実施:
 - MPIによる並列計算プログラミング入門
 - 並列有限要素法のためのデータ構造
 - 並列プログラムの作成法
 - Oakleaf-FX(Fujitsu FX10)によるプログラミング実習
- 夏学期に扱った三次元弾性静解析プログラム「fem3d」の並列化をMPIによって実施する

並列計算の意義

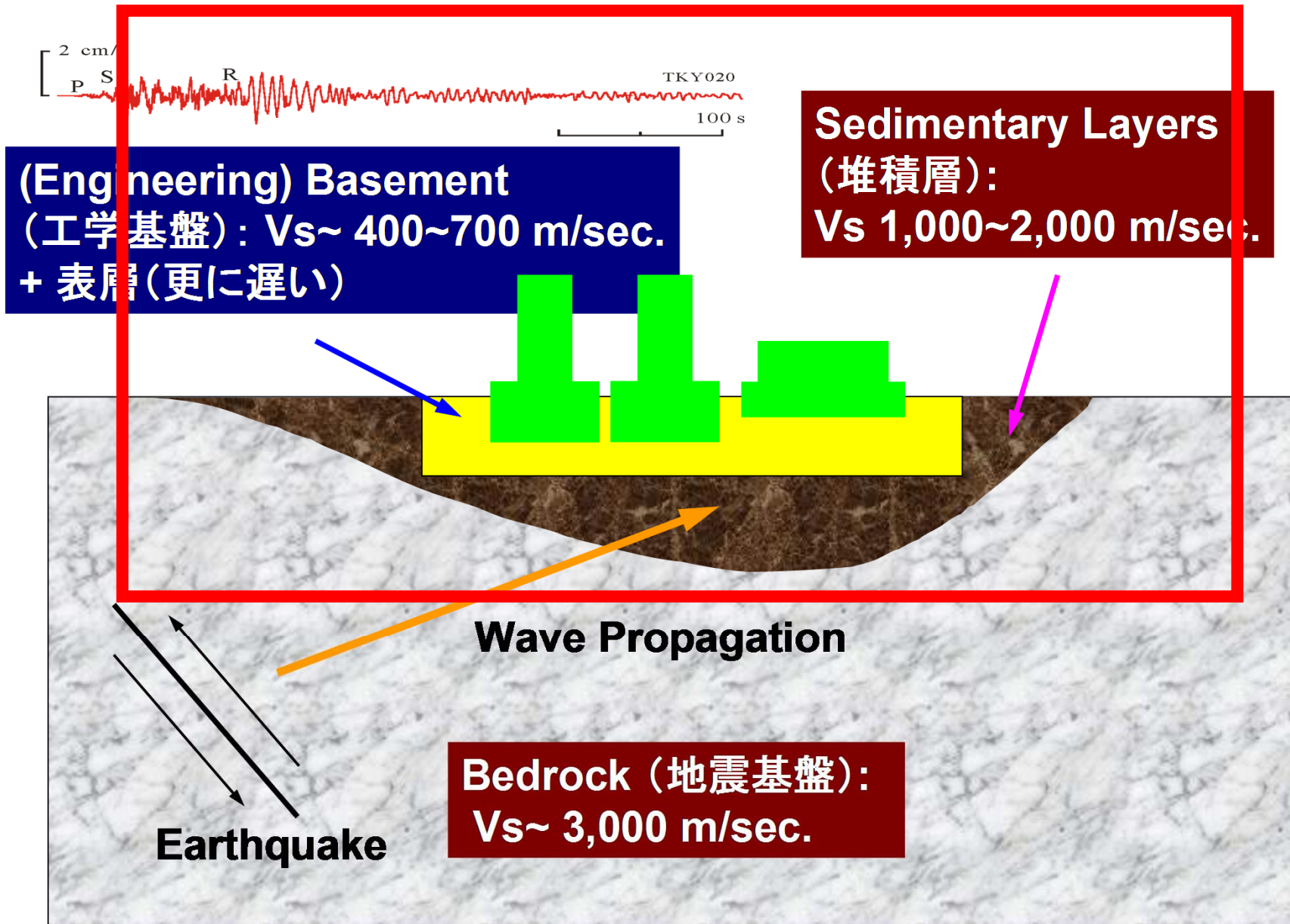
- たくさんの計算機を使って
- より速く
- より大規模に
- より複雑に
 - 連成, 連結シミュレーション

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

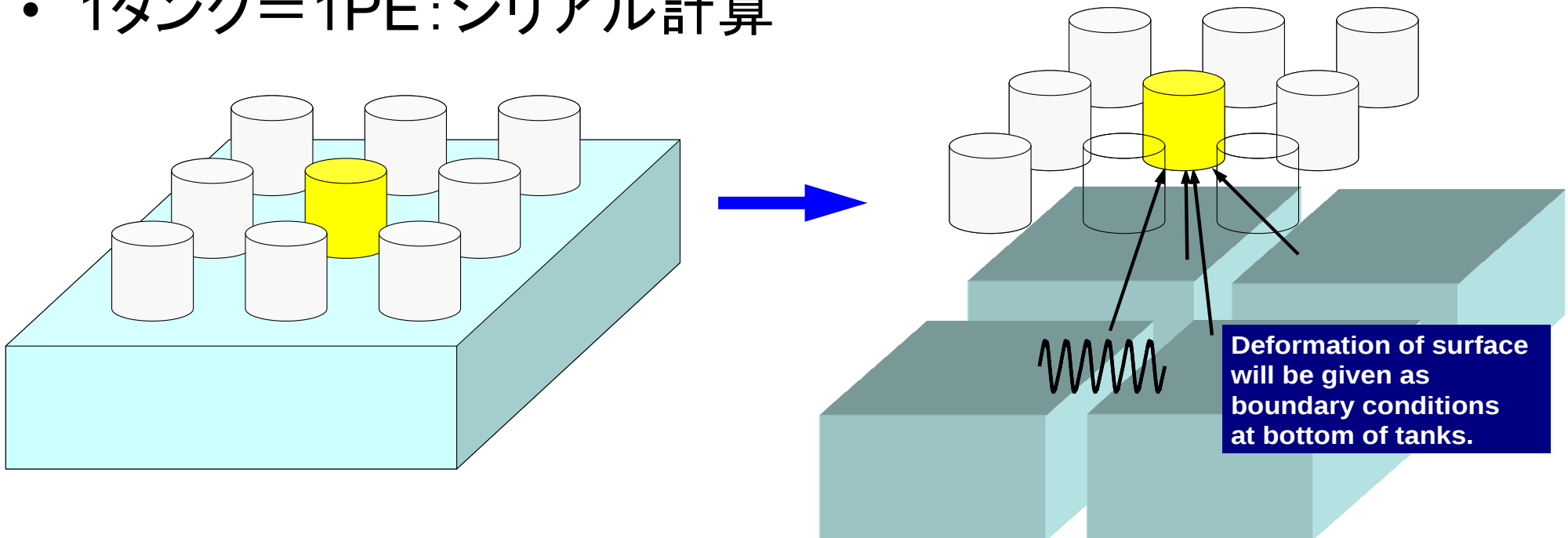


地盤・石油タンク振動連成シミュレーション



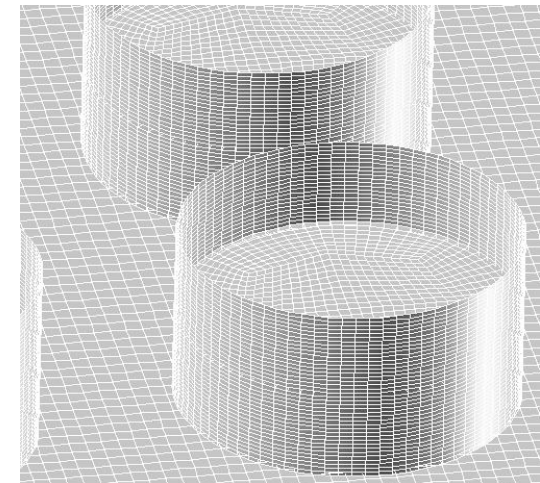
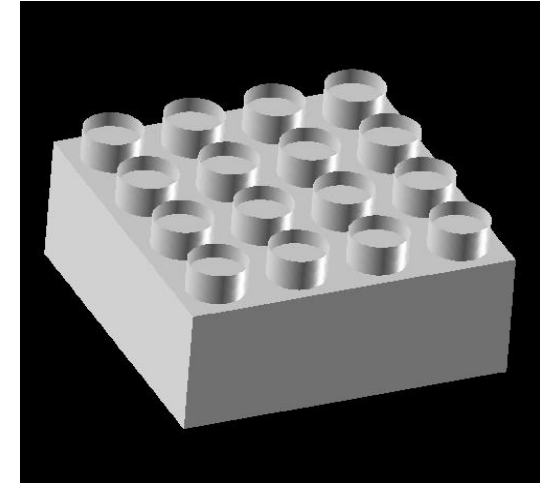
対象とするアプリケーション

- 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

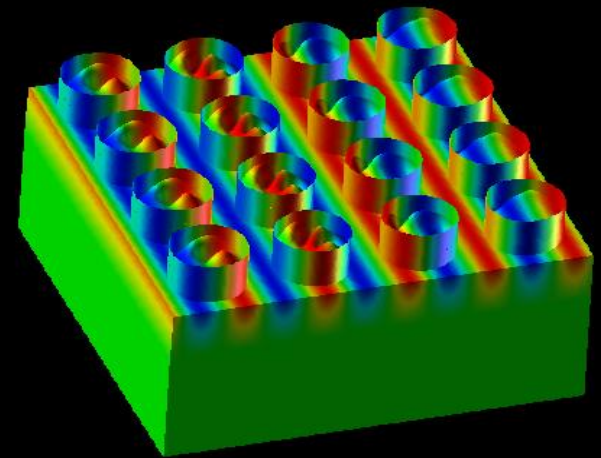
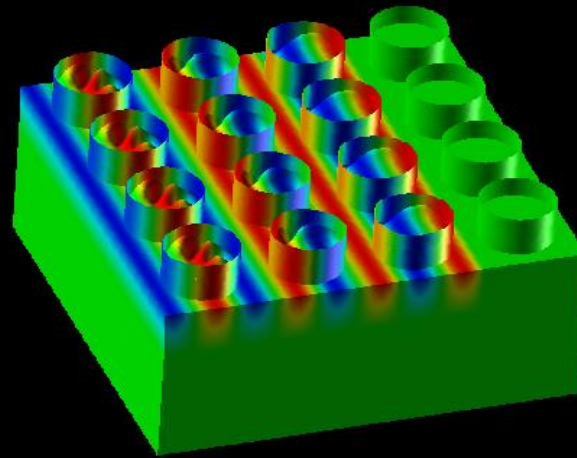
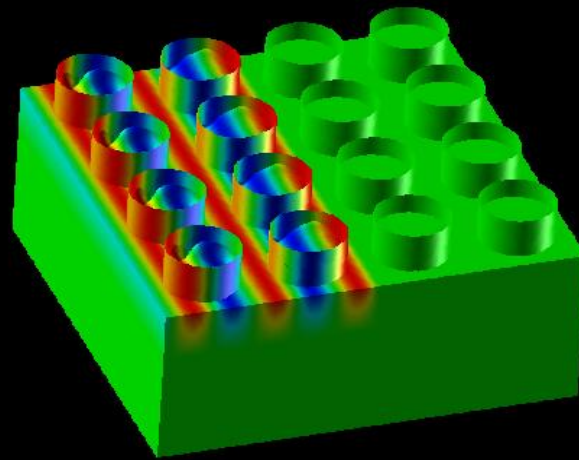
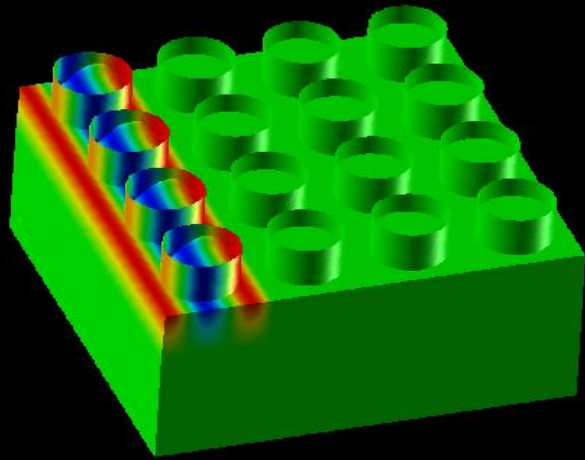


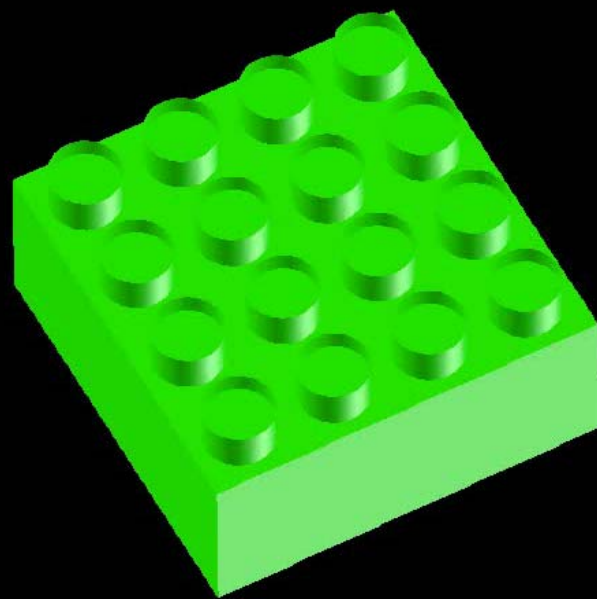
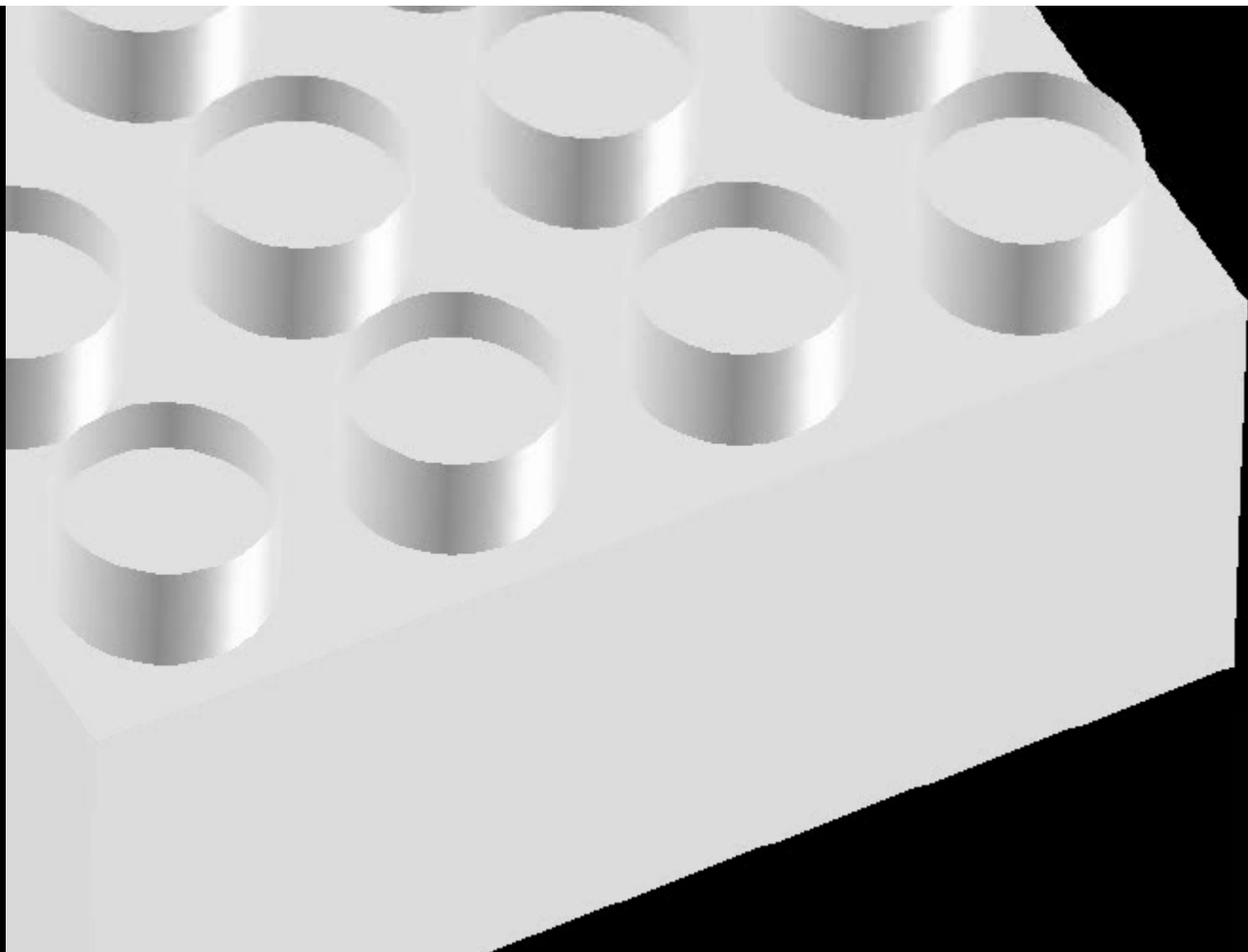
地盤，タンクモデル

- 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
 - $240\text{m} \times 240\text{m} \times 100\text{m}$
- 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





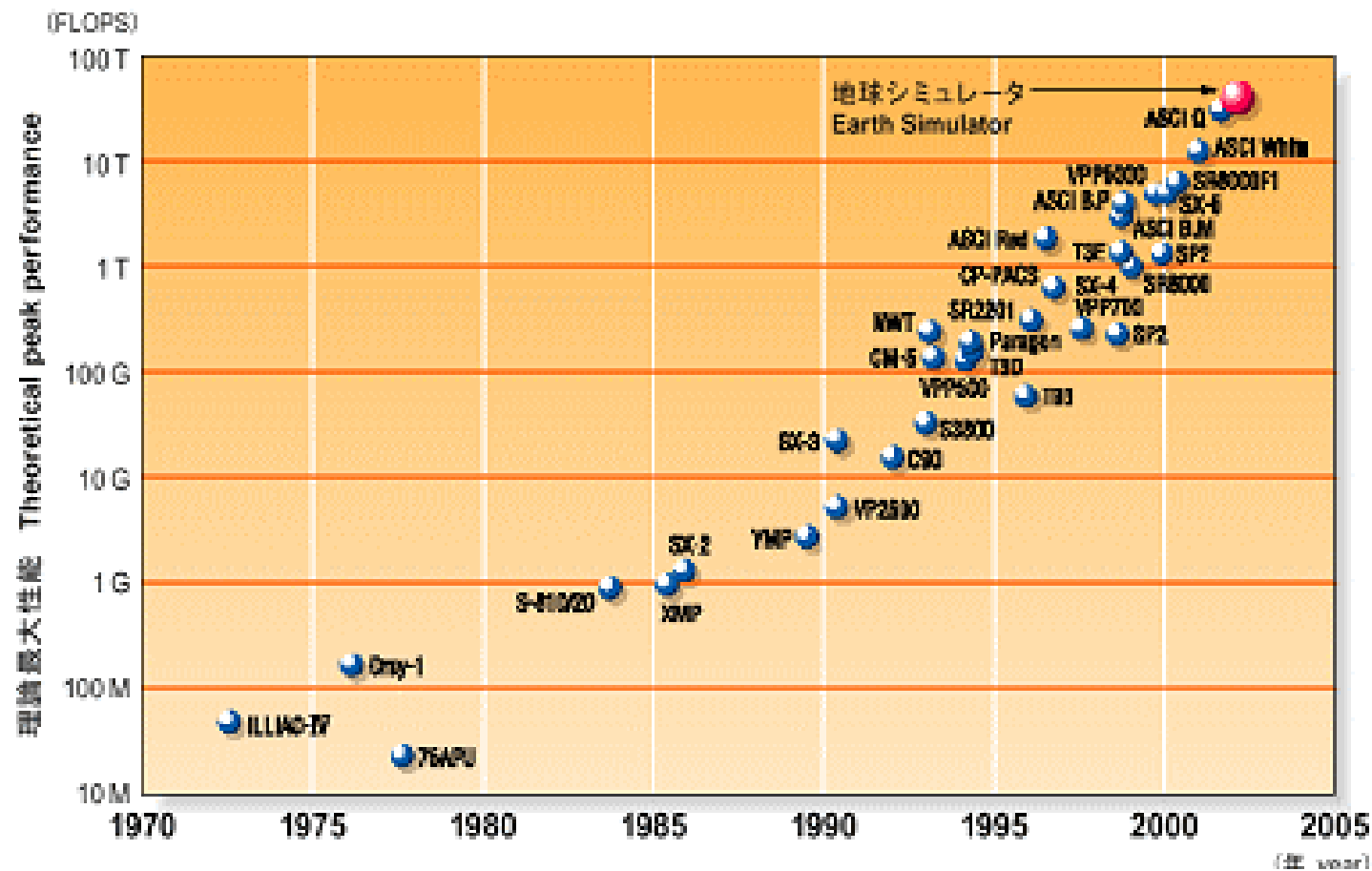
本講義の歴史

- 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

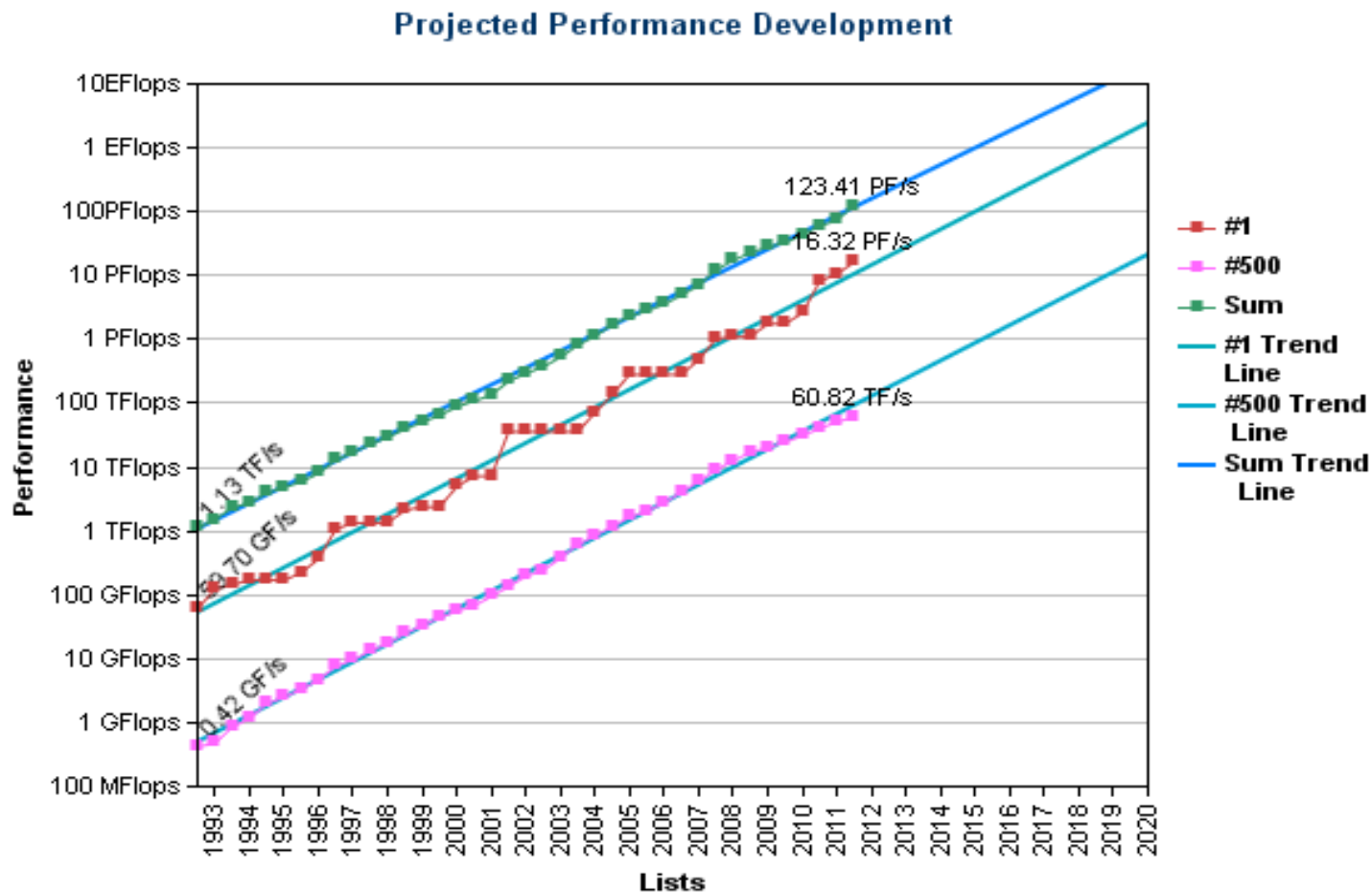


<http://www.es.jamstec.go.jp/>

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%
- 最近は消費電力も考慮したGreen500というランキングもある



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

39th TOP 500 List (June 2012) (1/2)

	Site	Computer/Year Vendor	Cores	R _{max}	R _{peak}	Power
1	LLNL United States	Sequoia - BlueGene/Q 2011 IBM	1572864	16324.75	20132.66	7890.0
2	RIKEN AICS Japan	K computer - SPARC64 VIIIfx 2011 Fujitsu	705024	10510.00	11280.38	12659.9
3	Argonne United States	Mira - BlueGene/Q 2012 IBM	786432	8162.38	10066.33	3945.0
4	Leibniz Rechenzentrum Germany	SuperMUC - iDataPlex/Xeon E5-2680, 2012 IBM	147456	2897.00	3185.05	3422.7
5	NSCS in Tianjin China	Tianhe-1A 2010 NUDT	186368	2566.00	4701.00	4040.0
6	ORNL United States	Jaguar - Cray XK6 2009 Cray Inc.	298592	1941.00	2627.61	5142.0
7	CINECA Italy	Fermi - BlueGene/Q 2012 IBM	163840	1725.49	2097.15	821.9
8	Forschungszentrum Juelich (FZJ) Germany	JuQUEEN – BlueGene/Q 2012 IBM	131072	1380.39	1677.72	657.5
9	CEA/TGCC-GENCI France	Curie thin nodes- Xeon E5-2680 8C 2012 Bull	77184	1359.00	1667.17	2251.0
10	NSCS in Shenzhen China	Nebulae 2010 Dawning	120640	1271.00	2984.30	2580.0

39th TOP 500 List (June 2012) (2/2)

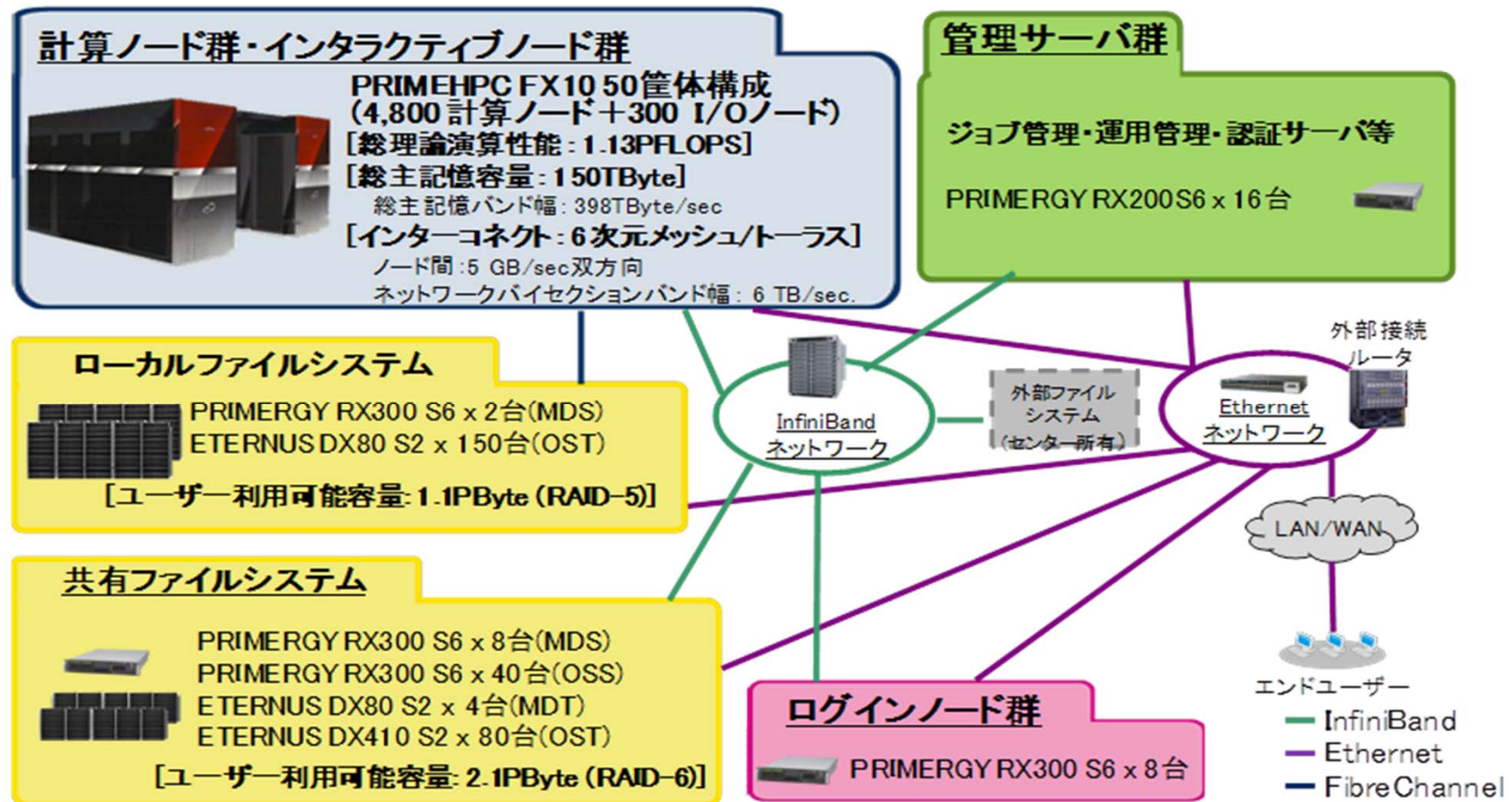
	Site	Computer/Year Vendor	Cores	R _{max}	R _{peak}	Power
11	NASA Ames United States	Pleiades - SGI Altix ICE 2011 SGI	125980	1243.00	1731.84	3987.0
12	IFRC, EU-Japan Japan	Helios - Xeon E5-2680 2011 Bull	70560	1237.00	1524.10	2200.0
13	Daresbury Lab. United Kingdom	Blue Joule - BlueGene/Q 2012 IBM	114688	1207.84	1468.01	575.3
14	GSIC – Tokyo Tech Japan	TSUBAME 2.0 2010 NEC/HP	73278	1192.00	2287.63	1398.6
15	LANL/SNL United States	Cielo - Cray XE6 2011 Cray Inc.	142272	1110.00	1365.81	3980.0
16	LBNL United States	Hopper - Cray XE6 2010 Cray Inc.	153408	1054.00	1288.63	2910.0
17	CEA France	Tera-100 - Bull bullx super-node 2010 Bull	138368	1050.00	1254.55	4590.0
18	ITC/U. Tokyo Japan	Oakleaf-FX - SPARC64 IXfx 2012 Fujitsu	76800	1043.00	1135.41	1176.8
19	LANL United States	Roadrunner 2009 IBM	122400	1042.00	1375.78	2345.0
20	U. Edinburgh United Kingdom	DiRAC – BlueGene/Q 2012 IBM	98304	1035.30	1258.29	493.1

「京」コンピュータ

- 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
- 2012年6月現在世界第2位
- 2012年9月28日正式運用開始



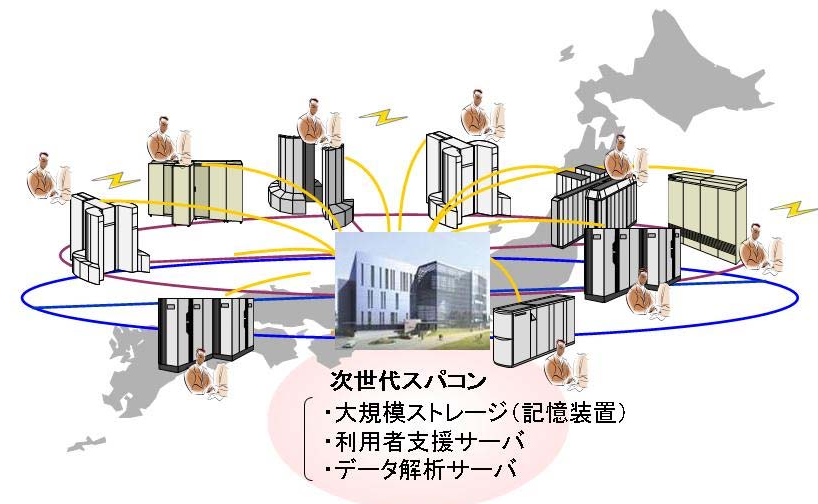
FX10 System (Oakleaf-FX)



- Aggregate memory bandwidth: 398 TB/sec.
- Local file system for staging with 1.1 PB of capacity and 131 GB/sec of aggregate I/O performance (for staging)
- Shared file system for storing data with 2.1 PB and 136 GB/sec.
- External file system: 3.6 PB

革新的ハイパフォーマンス・ コンピューティング・インフラ(HPCI)

- 次世代スーパーコンピュータ(京)
- HPCI
 - 「京」と全国の大学や研究所などに設置されている主要なスパコンをネットワークで結び、利用者の多様なニーズに応える計算環境を実現
 - データの共有, 共同分析
 - HPCIコンソーシアム
 - スパコン資源提供機関, 38機関
 - 平成24年9月28日運用開始
 - <https://www.hpci-office.jp/>



HPCI共用ストレージの運用(東大:東西拠点)

HPCI (High Performance Computing Infrastructure)

- 京コンピュータを頂点とし国内のスパコン群をSINET4を介して利用
- 京コンピュータの運用開始(平成24年秋)とともにHPCI資源も提供開始予定
- 文科省の支援を受けて、HPCIとして拠出される計算資源をユーザは無償で利用可能
- データ共有インフラを提供

理研計算科学研究機構に設置されているHPCI共用ストレージ等

- W1 storage
 - Gfarm
 - DDN SFA10000(Total10PB)
 - メタデータサーバ2台
 - データサーバ16台
 - 10GbE ネットワーク
- データ解析システム
 - 理論ピーク性能 12.37TFlops
 - 総主記憶容量8.4TB
 - 計算ノード88台 ログインノード2台
- 60 PB tape archive



本ストレージおよびポスト・プレ処理用クラスター群は、最先端研究施設補助金「e-サイエンス実現のためのシステム統合・連携ソフトウェアの高度利用促進」で整備された機器がベース

東京大学情報基盤センターが提供する機器

- HITACHI HA8000
 - 512ノード提供可能
 - 75.366TFlops
- E1 storage
 - Gfarm
 - DDN SFA10000 9 セット (Total 8PB)
 - メタデータサーバ8台
 - データサーバ36台
 - ログイン4台
 - 10GbE ネットワーク
- E2 storage
 - Gfarm
 - DDN SFA10000(Total 5.5PB)
 - メタデータサーバ2台
 - データサーバ8台
 - 10GbE ネットワーク
- E3 storage
 - DDN S2A6620 288TB
 - メタデータサーバ2台、データサーバ5台、ログイン1台
- 20 PB tape archive (プリ・ポスト処理、可視化のための装置)
- データ解析システム
 - 理論ピーク性能 13.78TFlops
 - 総主記憶容量 8.3TB
 - 計算ノード63台
- GPUクラスシステム
 - 理論ピーク性能 CPU 4.5TFlops+GPU16.48TFlops
 - 総主記憶容量1.5TB
 - 計算ノード32台、ログインノード3台



HPCI戦略プログラム

- HPCIを活用した成果の創出と、研究推進・研究支援や人材育成等を進めるための体制整備：戦略分野・機関（平成21年制定）
 - － 戦略分野1 予測する生命科学・医療および創薬基盤（戦略機関：理化学研究所）
 - － 戦略分野2 新物質・エネルギー創成（東大物性研，分子研，東北大金材研）
 - － 戦略分野3 防災・減災に資する地球変動予測（JAMSTEC）
 - － 戦略分野4 次世代ものづくり（戦略機関：東大生研，JAXA，JAEA）
 - － 戦略分野5 物質と宇宙の起源と構造（筑波大，高エネ研，国立天文台）

HPCI戦略プログラム

予測する生命科学・医療 および創薬基盤

予測医療と革新的創薬

臓器レベルでの疾患を再現する階層統合シミュレーションを実現し、予測医療に貢献。また、標的タンパク質に強く結合する薬の候補化合物の設計を行い、創薬プロセスを加速。



血栓成長による血管閉塞シミュレーション

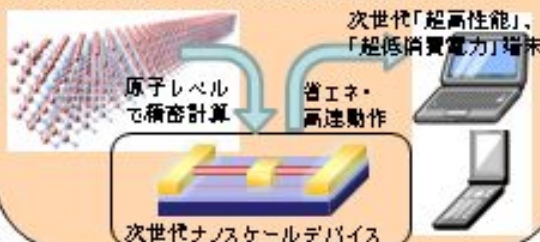


薬候補のタンパク質への結合シミュレーション

新物質・エネルギー創成

世界に先駆けた次世代デバイスを提唱

ナノスケールデバイスをまるごとシミュレーションし、機能・材料特性予測を実現することで、次世代デバイスの設計手法を提唱、超高性能・超低消費電力端末等の実現に貢献する。

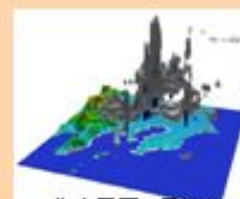


次世代ナノスケールデバイス

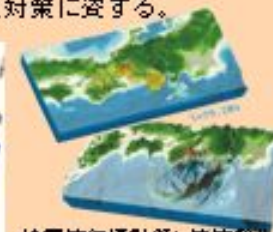
防災・減災に資する 地球変動予測

集中豪雨や地震の予測

雲解像モデル、強震動モデル等を駆使して、集中豪雨の位置や地震の被害規模を高精度に予測し、防災・減災対策に資する。



集中豪雨や局地的大雨の予測

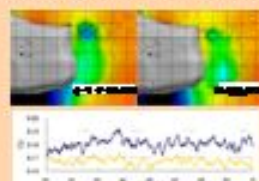


地震波伝播計算と津波発生伝播の連成シミュレーション

次世代ものづくり

設計プロセスの革新

熱流動の物理メカニズム理解に基づいた高度な設計制御技術確立することで、環境 (CO₂・NO_x) と製品性能のバランスを目指した将来の製品競争力強化に資する革新的ものづくりを実現



車体後部周りの起流場解析による最適形状の究明



非定常空力・振動連成解析による、低空気抵抗、低騒音車の開発

物質と宇宙の起源と構造

大質量星の超新星爆発の解明

超新星爆発の3次元シミュレーション



爆発時の密度分布

磁場増幅、ニュートリノ輻射輸送などを考慮した3次元シミュレーションを、次世代スパコンを用いることで世界に先駆けて実行し、大質量星が重力崩壊から超新星爆発に至る過程を解明する。

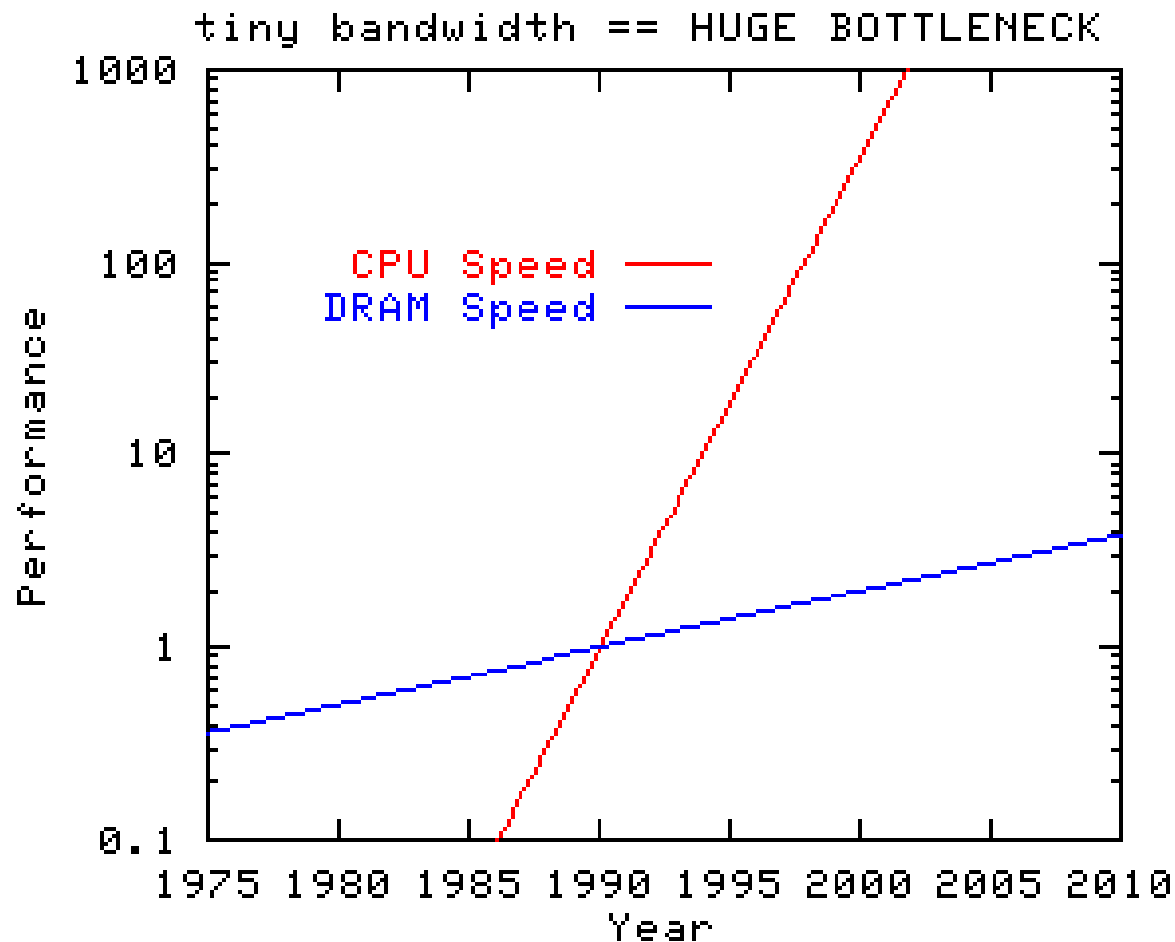
様々な技術的課題

- 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

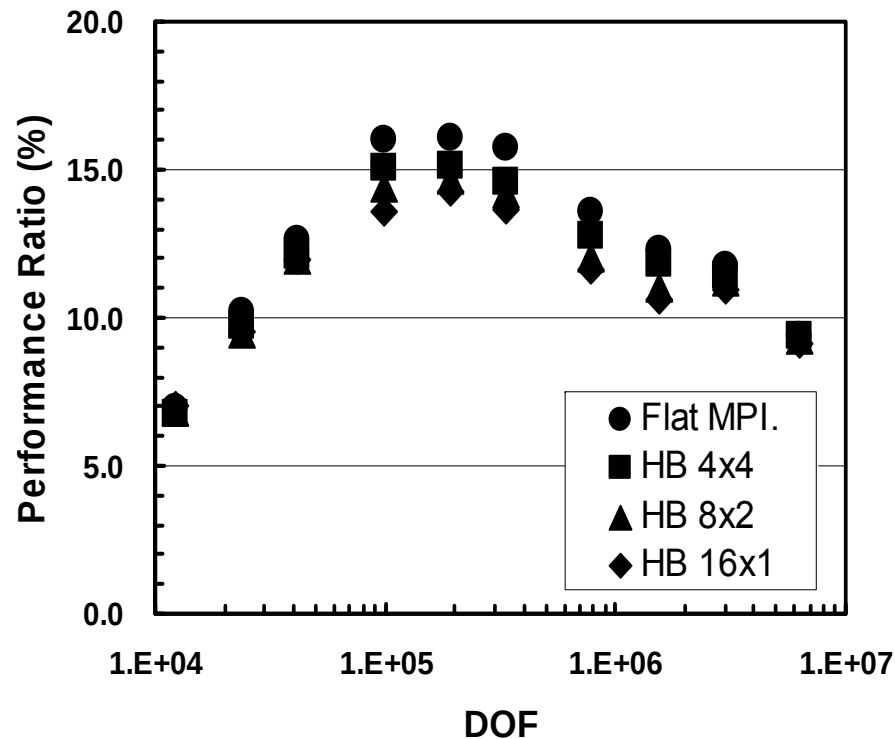
Hitachi SR11000/J2

Power 5+ 2.3GHz x 16

147.2 GFLOPS/node

100 GB/s for STREAM/Triad

L3 cache: 18MB/core



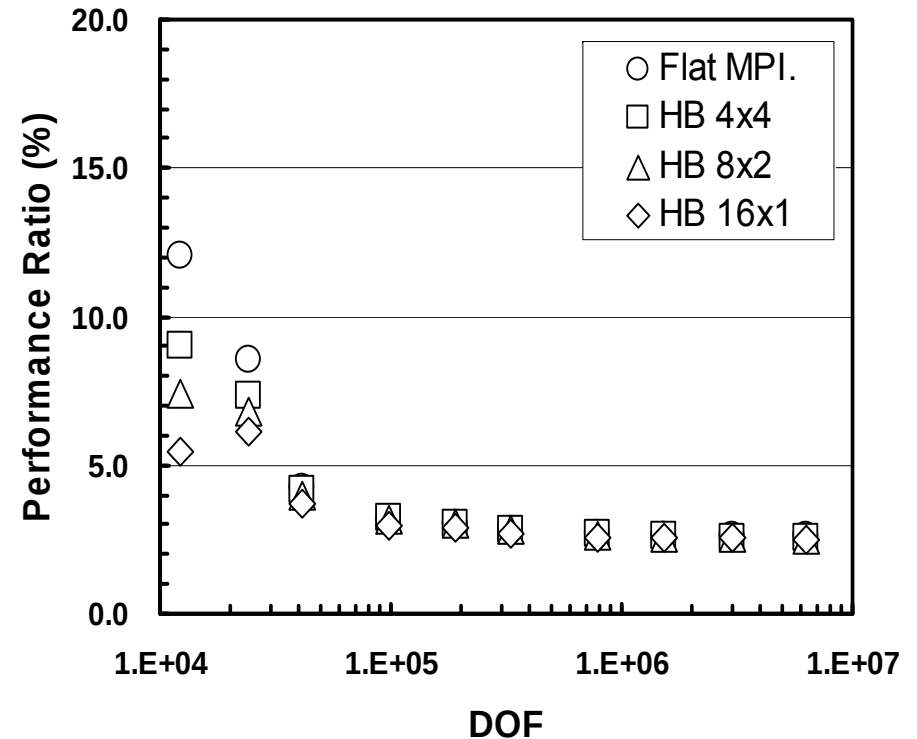
T2K/Tokyo

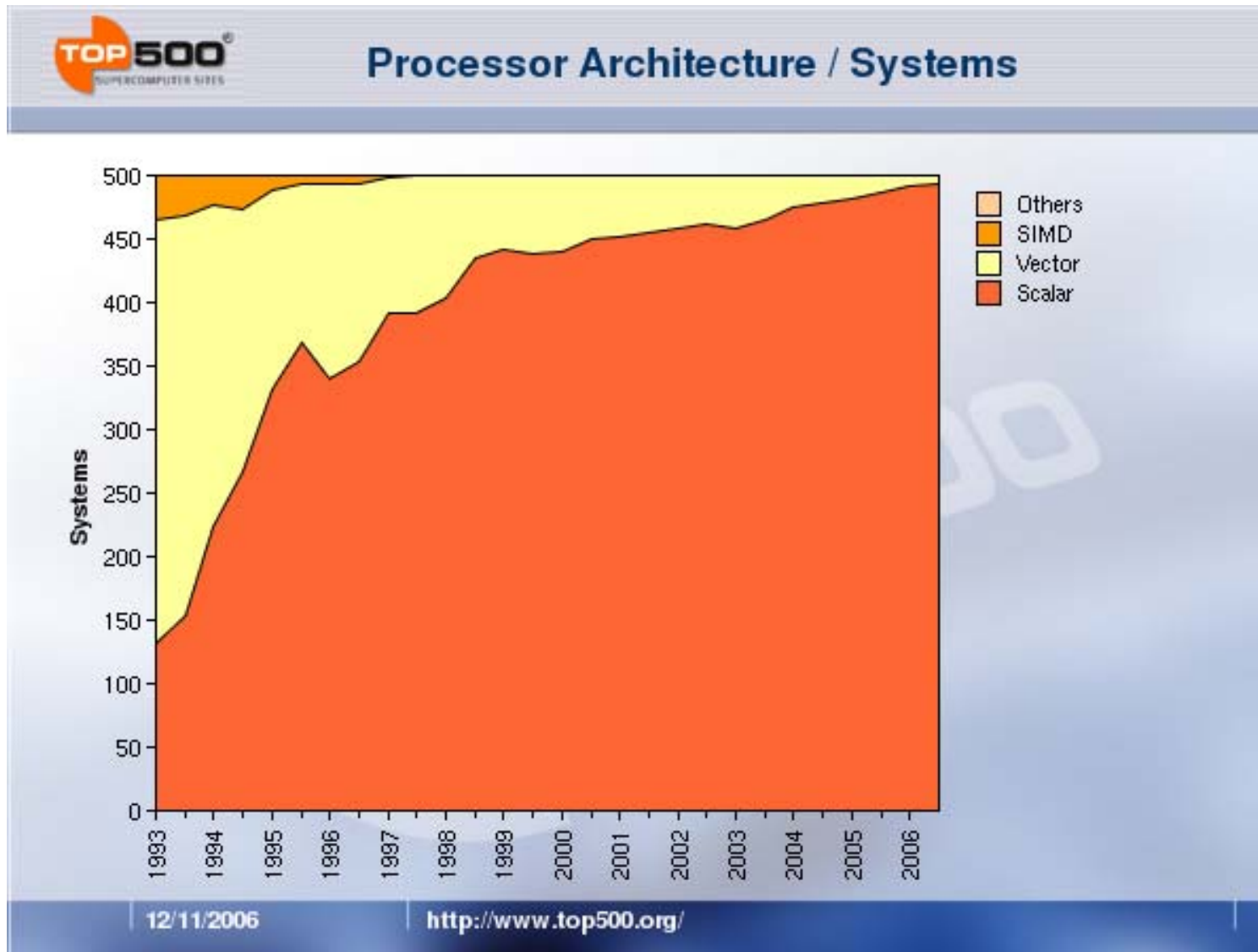
Opteron 2.3GHz x 16

147.2 GFLOPS/node

20 GB/s for STREAM/Triad

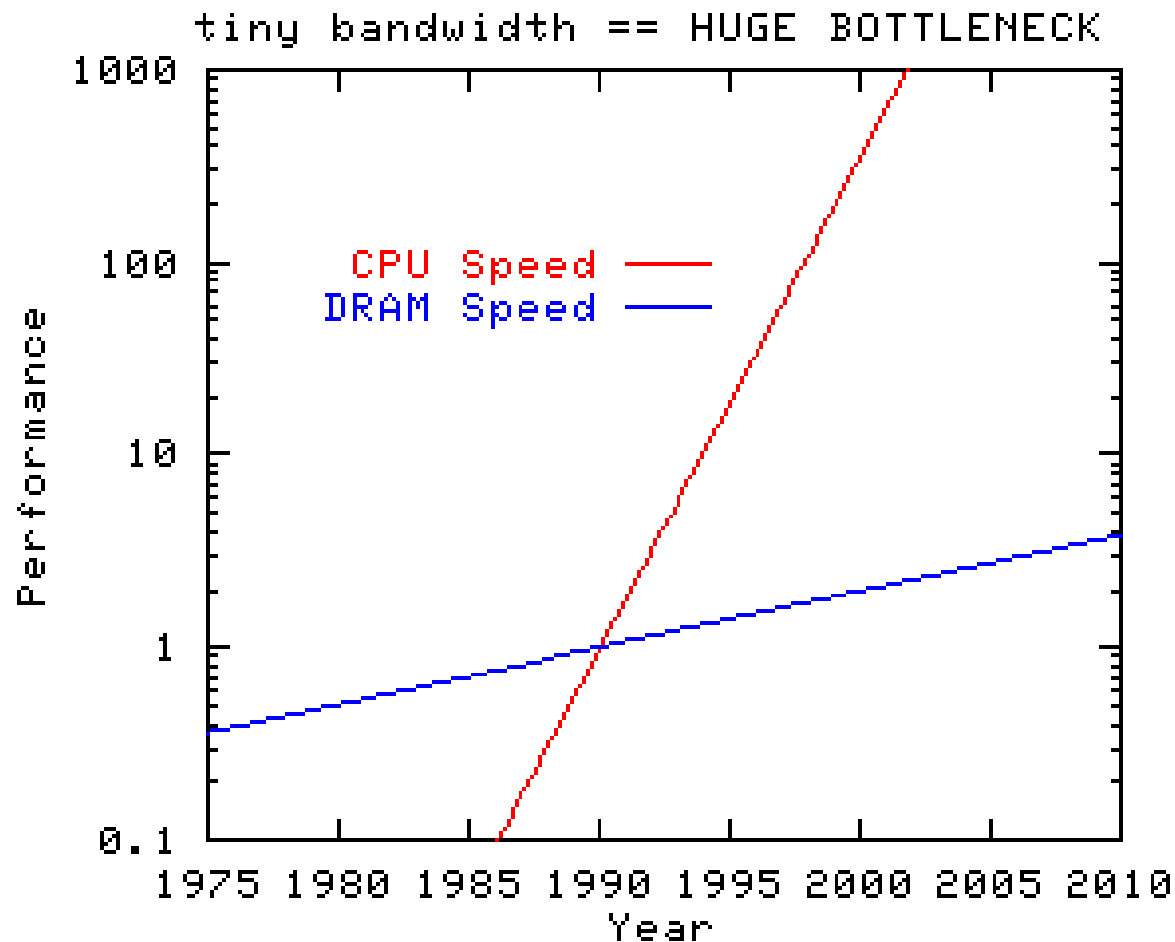
L3 cache: 0.5MB/core





マイクロプロセッサの動向

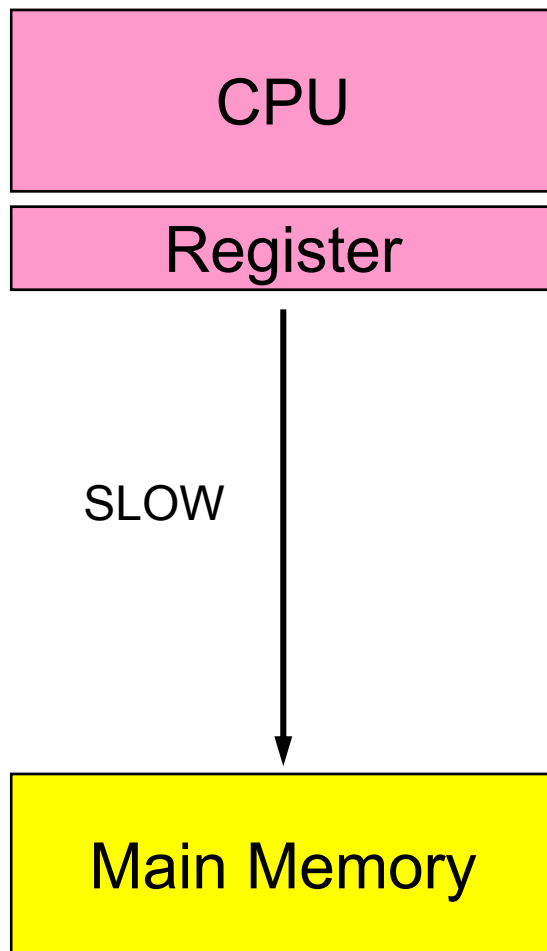
CPU性能, メモリバンド幅のギャップ

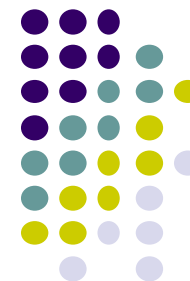


<http://www.streambench.org/>

スカラープロセッサ

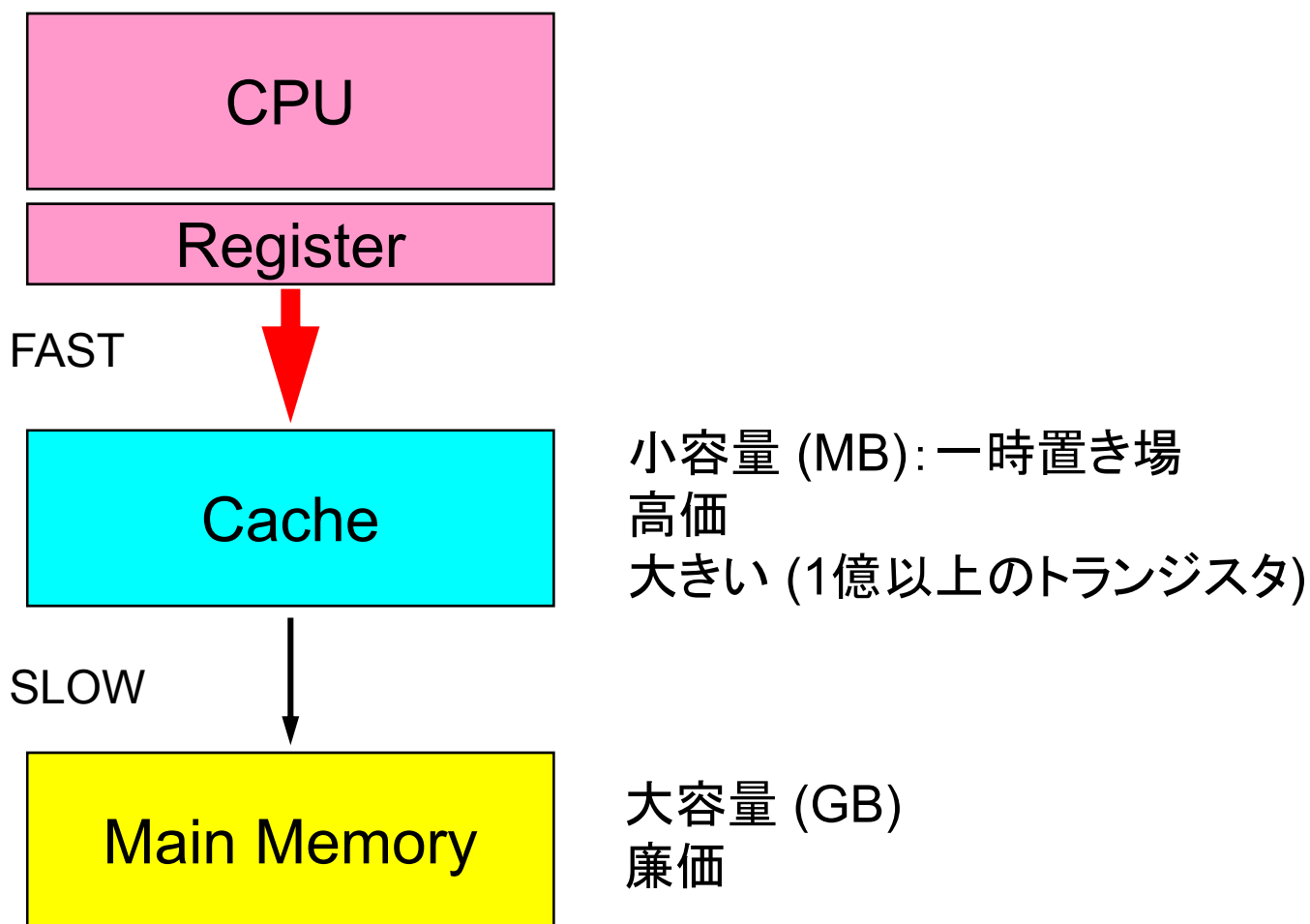
メモリへ直接アクセスするのは实际的でない





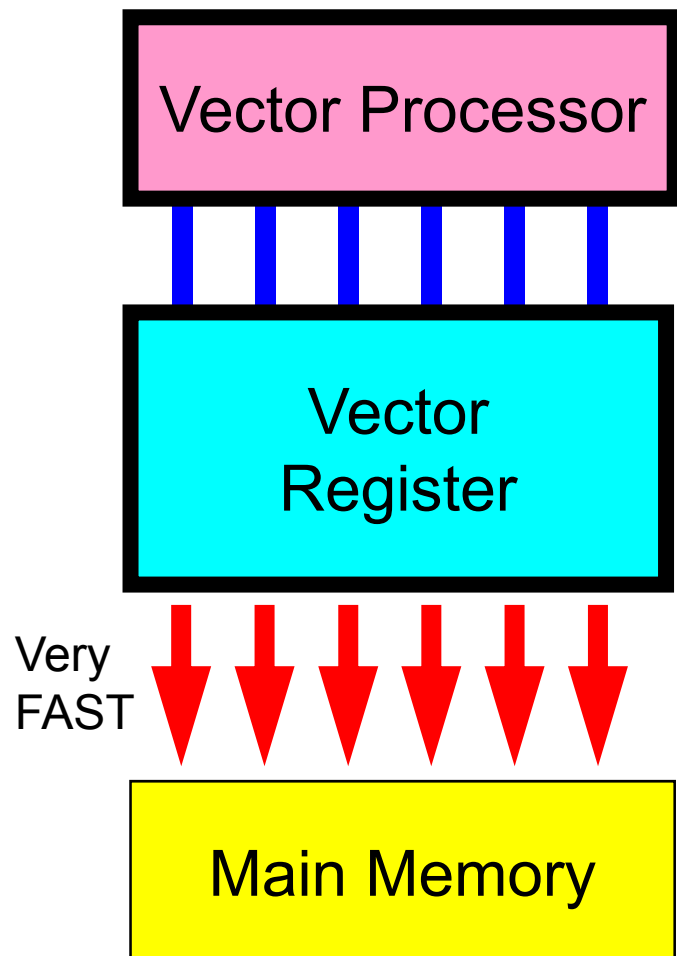
スカラープロセッサ

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

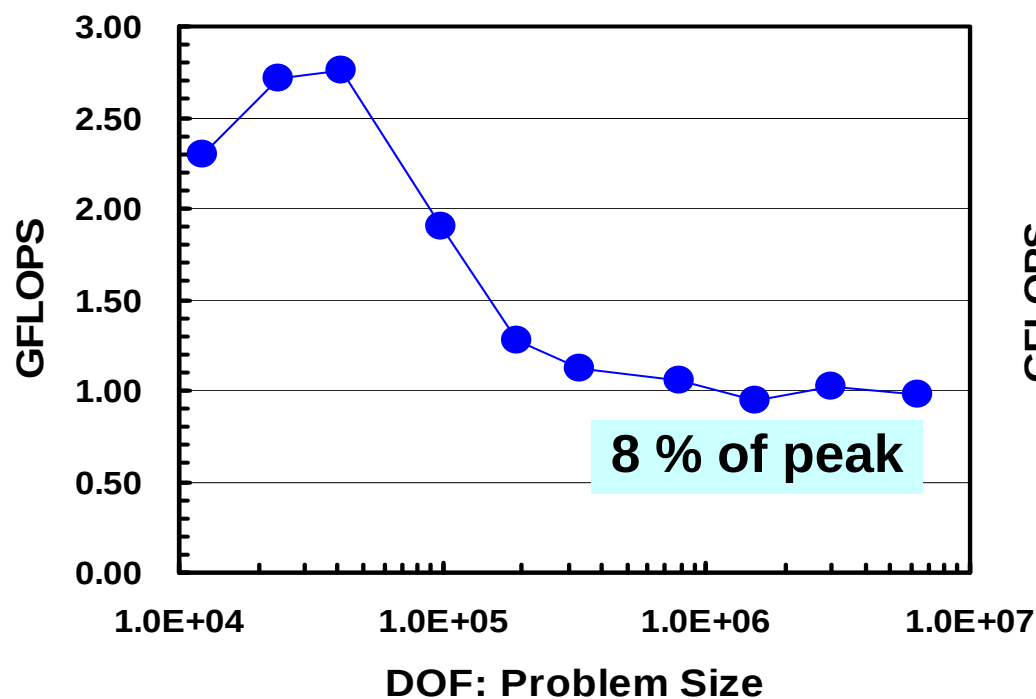


- 単純構造のDOループの並列処理
- 単純, 大規模な演算に適している

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

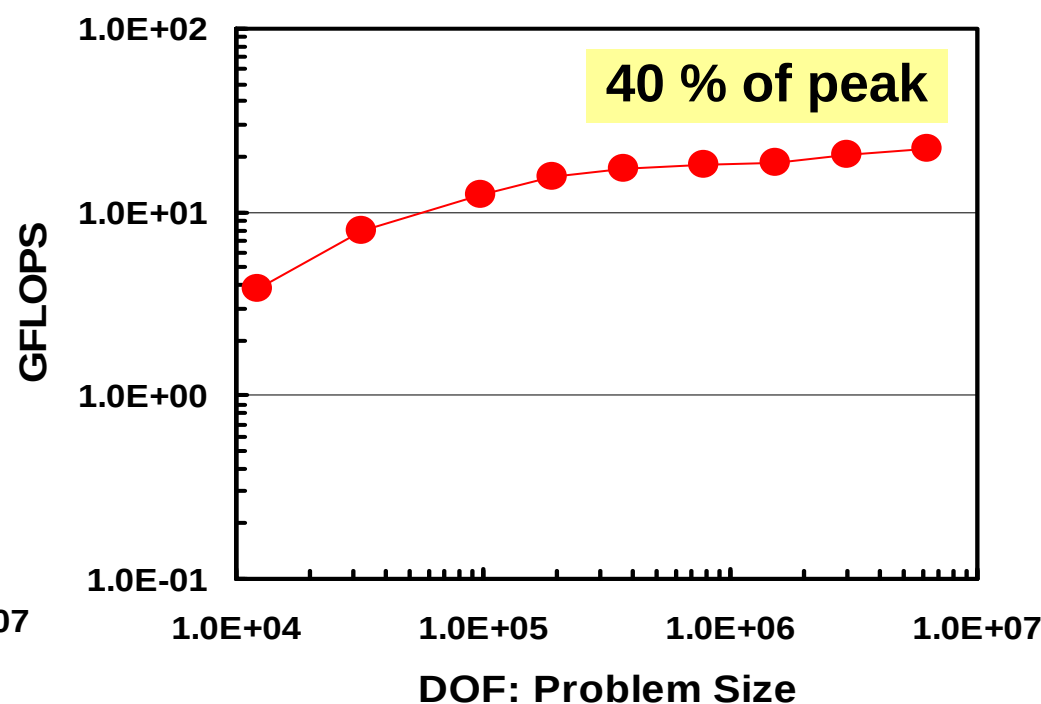


典型的な挙動:ICCG法



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

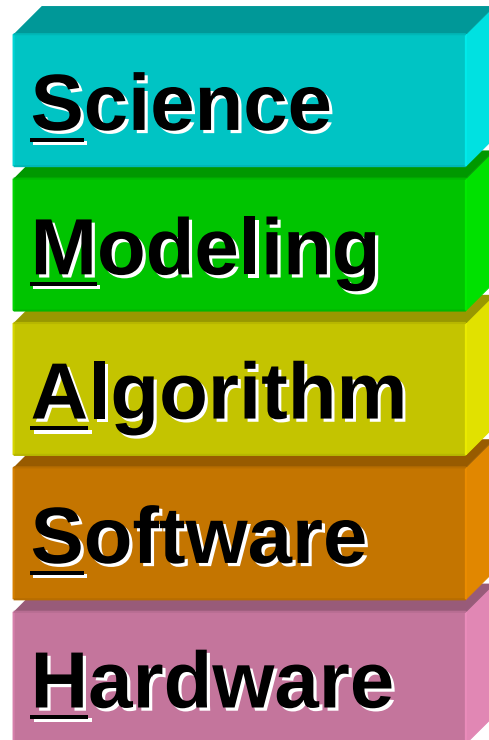


Earth Simulator:

大規模な問題ほどベクトル長が長くなり、性能が高い

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任
 - 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
- 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/12w/>

準備 等

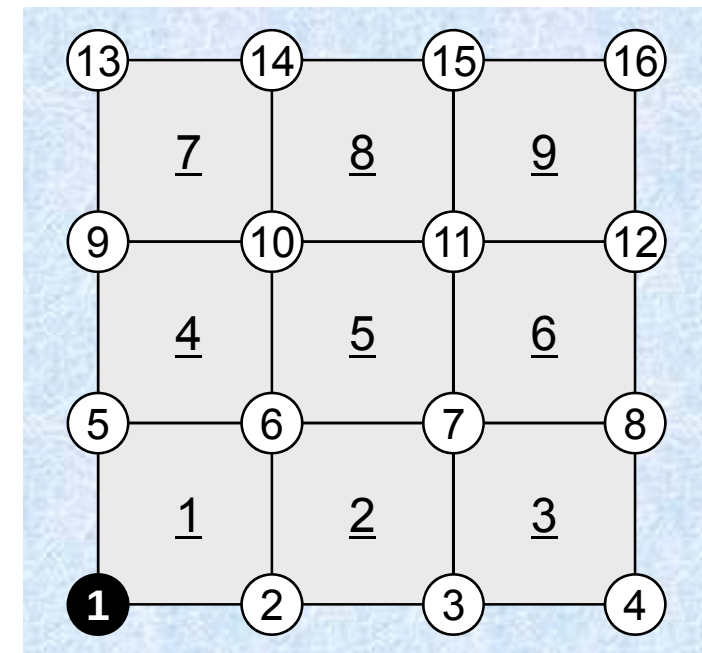
- 受講に必要な知識等
 - UNIXの操作
 - FORTRANまたはCによる基礎的なプログラミングの経験
 - 基礎的な数値解析に関する知識
- 情報基盤センター教育用計算機システムのアカウント
 - <http://www.ecc.u-tokyo.ac.jp/doc/announce/newuser.html>
- 「科学技術計算Ⅰ（有限要素法）」を履修していること、または同等の知識を有すること
 - <http://nkl.cc.u-tokyo.ac.jp/12s/>
- その他、連絡事項については随時HPに掲載する
 - <http://nkl.cc.u-tokyo.ac.jp/12w/>

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

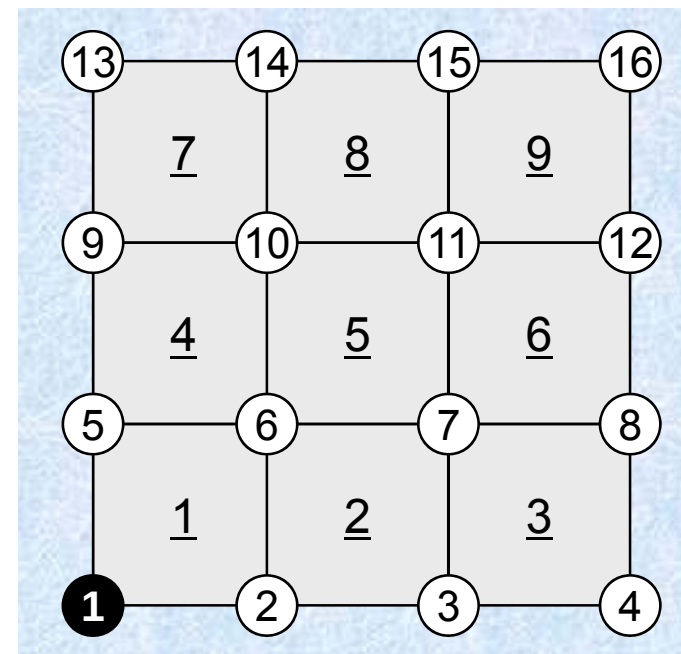
- 各要素にガラーキン法を適用:

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

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

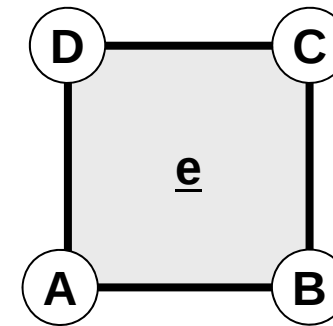
$$-\int_V \lambda \left(\frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial [N]^T}{\partial y} \frac{\partial [N]}{\partial y} \right) dV \cdot \{\phi\} + \int_V Q [N]^T dV = 0$$



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

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

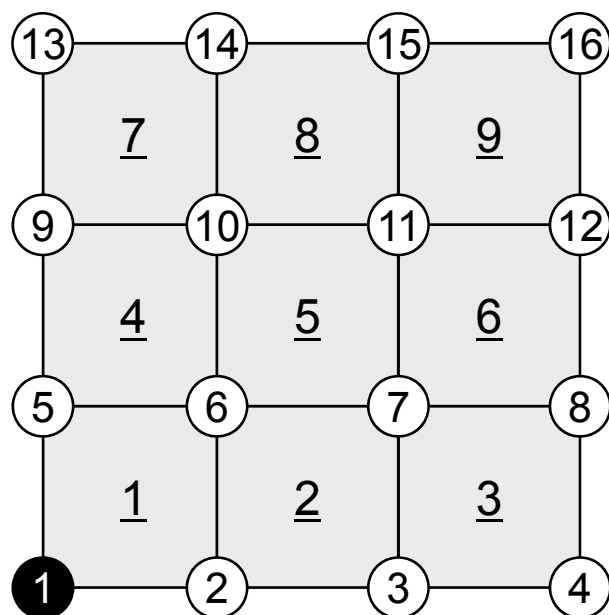


$$[k^{(e)}] \{\phi^{(e)}\} = \{f^{(e)}\}$$

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

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

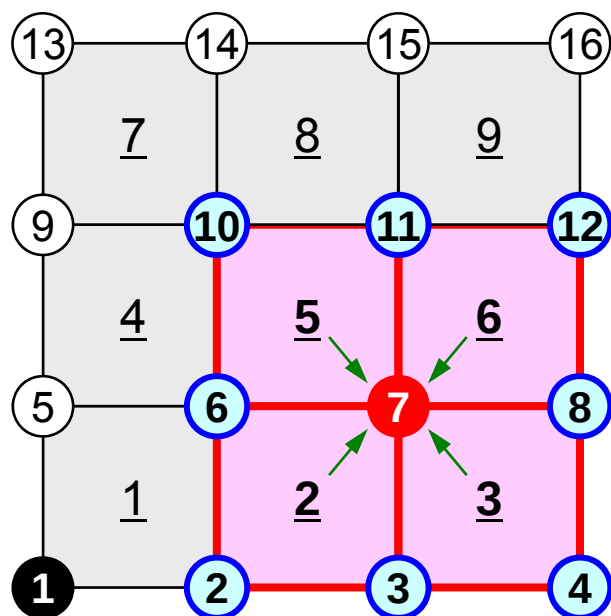


$$[K]\{\Phi\} = \{F\}$$

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

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



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

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

得られた大規模連立一次方程式を解く

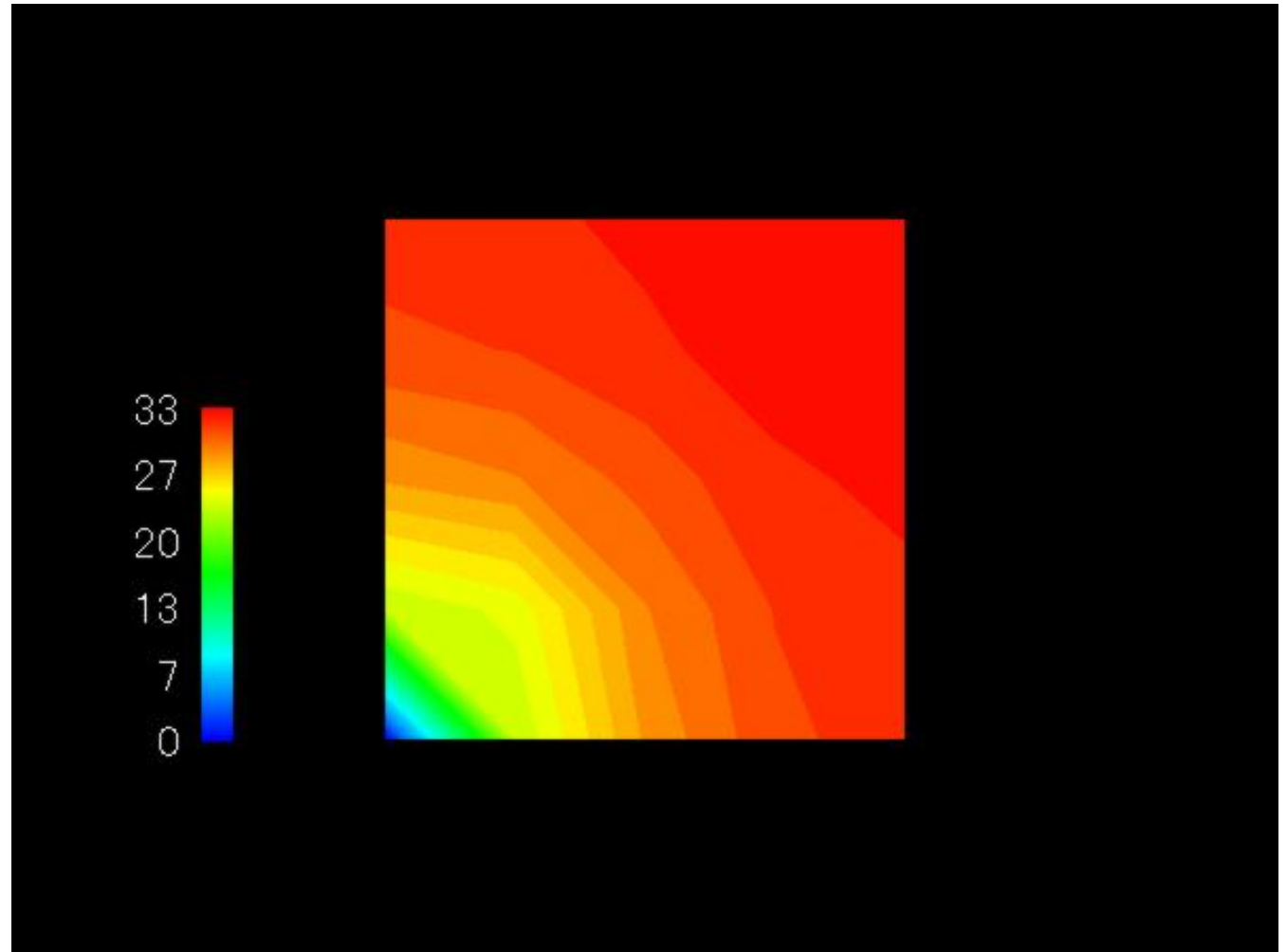
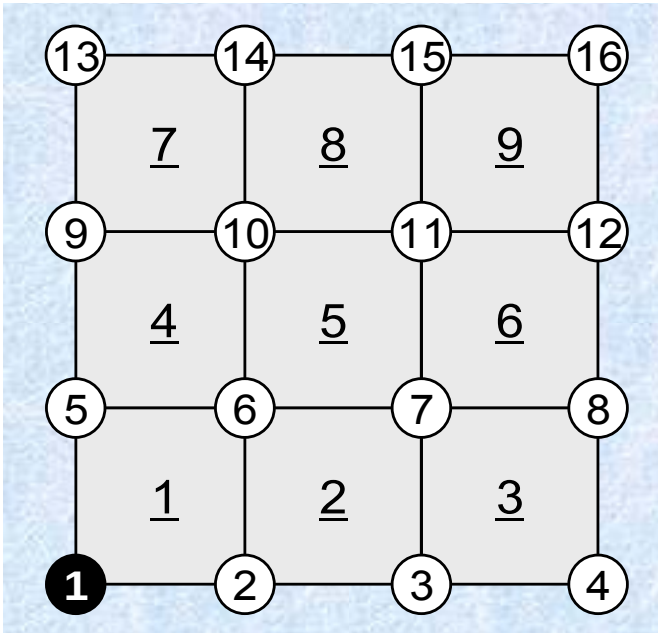
ある適切な境界条件 (ここでは $\Phi_1=0$) を適用

「疎 (ゼロが多い)」な行列

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

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
2012年	10月02日（火）	はじめに
	10月09日（火）	復習，並列データ構造
	10月16日（火）	並列データ構造（続き），Oakleaf-FX
	10月23日（火）	（休講）
	10月30日（火）	MPIによる並列プログラミング（1）
	11月06日（火）	MPIによる並列プログラミング（2）
	11月13日（火）	（休講）
	11月20日（火）	チューニング入門
	11月27日（火）	MPIによる並列プログラミング（3）
	12月04日（火）	MPIによる並列プログラミング（4）
	12月11日（火）	課題S1解説
	12月28日（火）	課題S2解説
2013年	01月08日（火）	三次元並列有限要素法（1）
	01月15日（火）	三次元並列有限要素法（2）
	01月22日（火）	（休講）
	01月29日（火）	三次元並列有限要素法（3）
	02月12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)
- 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++, FOTRAN, 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)があったが, 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に

- SPMD/SIMDのオペレーションに慣れること...「つかむ」こと

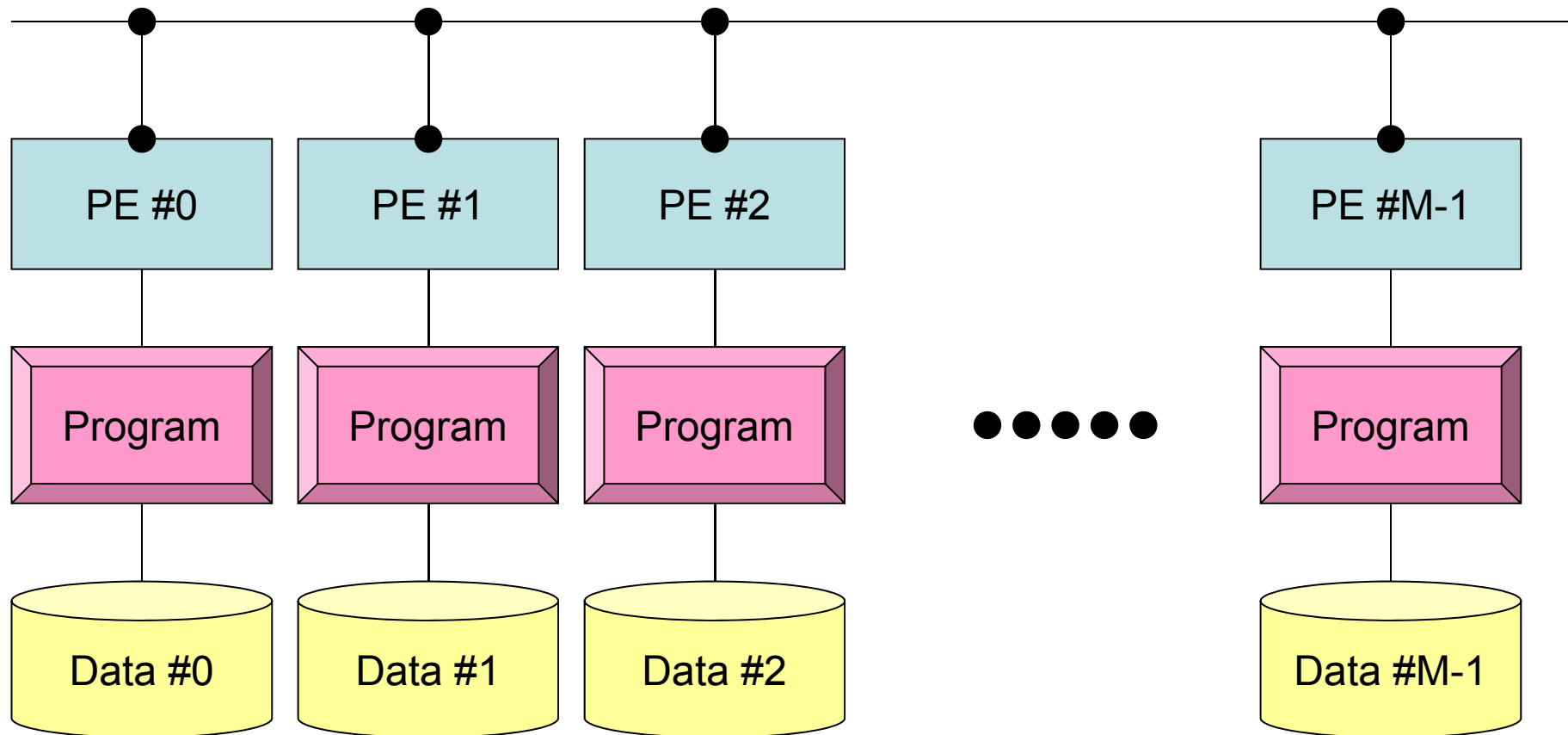
- Single Program/Instruction Multiple Data
- 基本的に各プロセスは「同じことをやる」が「データが違う」
 - 大規模なデータを分割し, 各部分について各プロセス(プロセッサ)が計算する
- 全体データと局所データ, 全体番号と局所番号

PE: Processing Element
プロセッサ, 領域, プロセス

SPMD/SIMD

```
mpirun -np M <Program>
```

この絵が理解できればMPIは
9割方理解できたことになる。
コンピュータサイエンスの学
科でもこれを上手に教えるの
は難しいらしい。

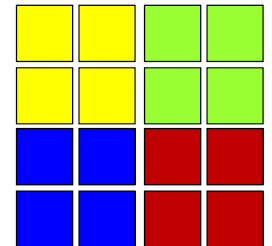
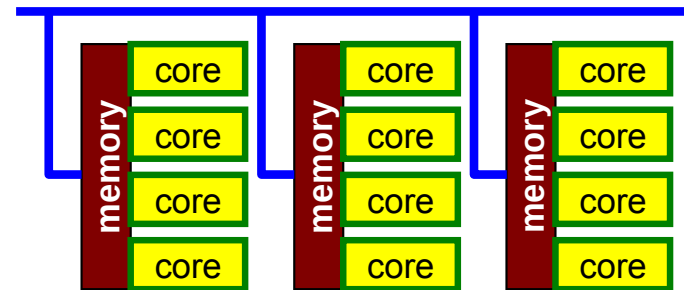


各プロセスは「同じことをやる」が「データが違う」
大規模なデータを分割し、各部分について各プロセス(プロセッサ)が計算する

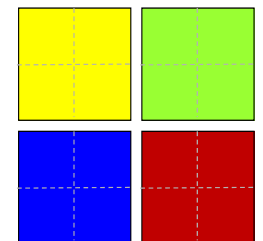
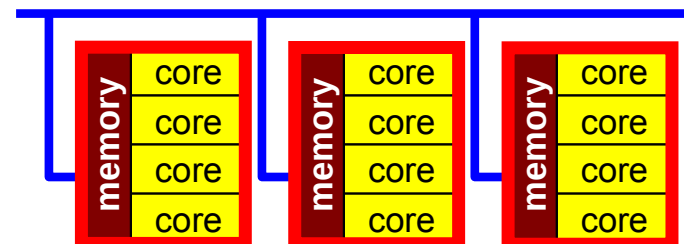
Hybrid並列プログラミング

- 本講義ではいわゆるFlat MPIを教える
- 時代の趨勢はHybrid
 - MPI+X: OpenMP, OpenACC, CUDA, OpenCL
- 12月4日・5日にOpenMPの講習会実施
 - <http://nkl.cc.u-tokyo.ac.jp/12e/05-Multicore/index.html>
- 「最近の話題」で少しとりあげる予定

Flat-MPI: Each PE -> Independent



Hybrid: Hierarchical Structure



- 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)

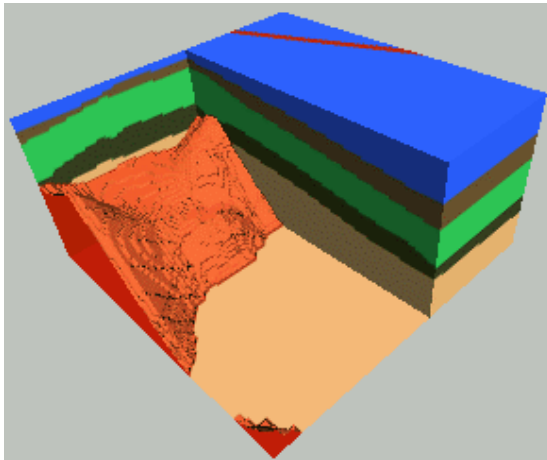
- これも余り簡単な話では無い

並列計算とは？(1/2)

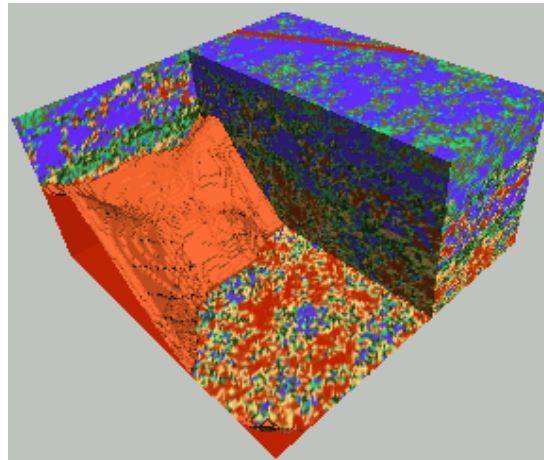
- より大規模で複雑な問題を高速に解きたい

Homogeneous/Heterogeneous Porous Media

Lawrence Livermore National Laboratory



Homogeneous

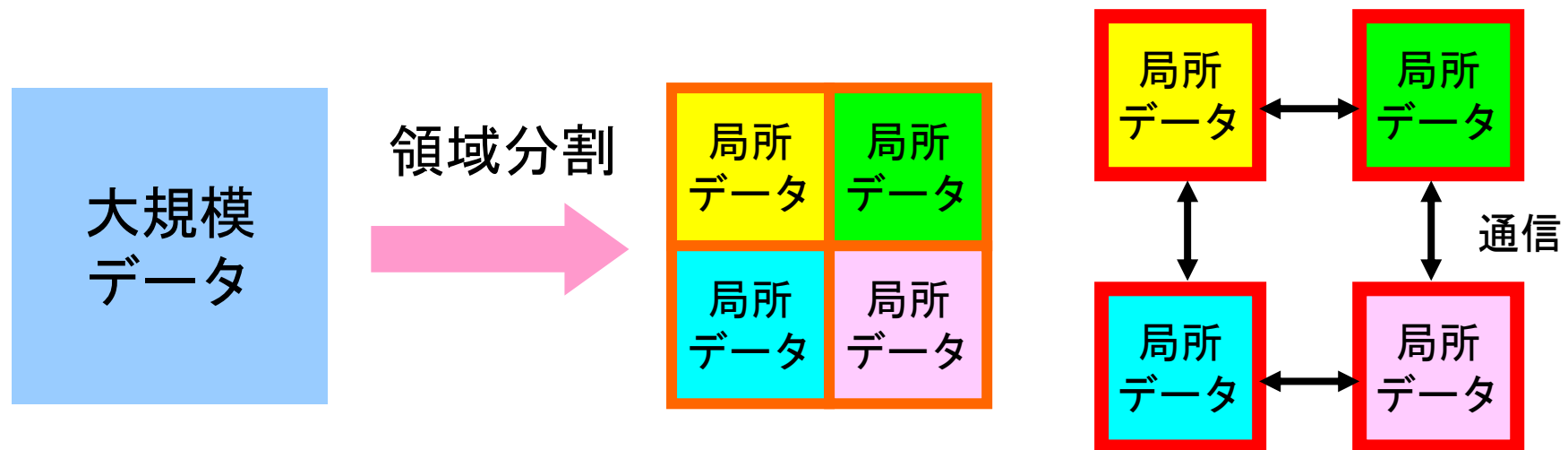


Heterogeneous

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？

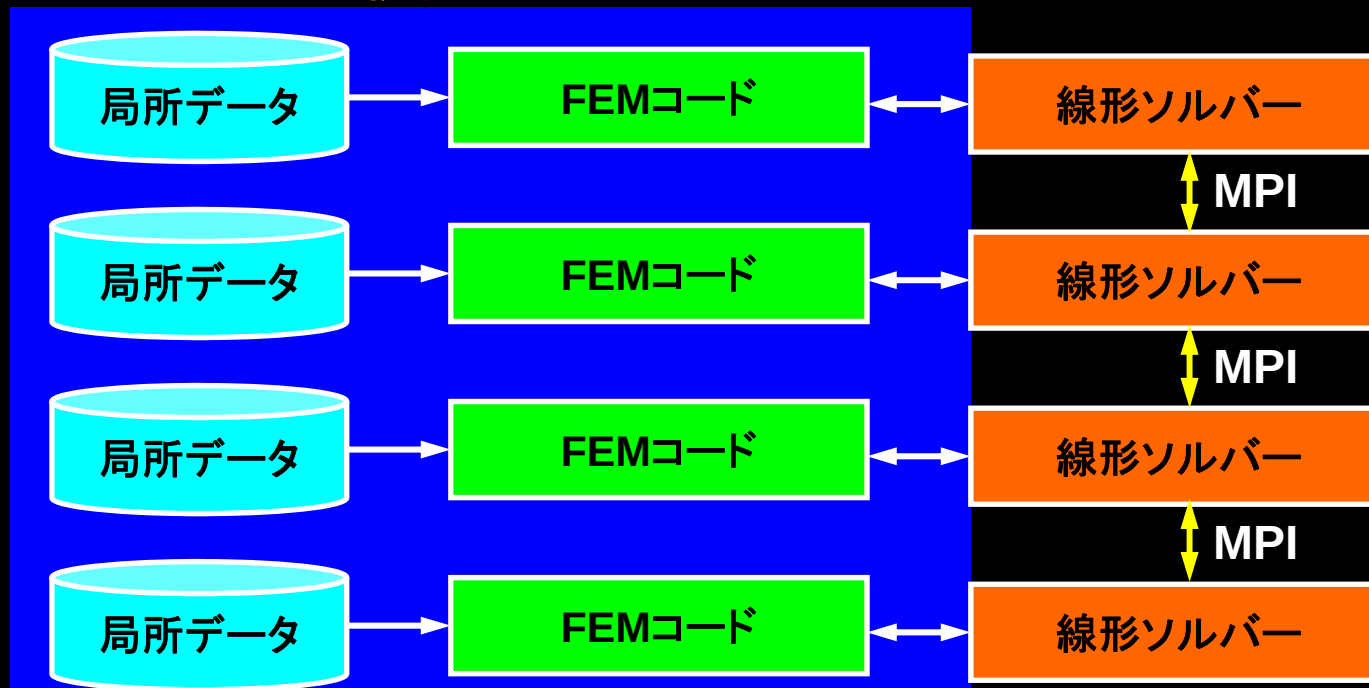
- 並列計算とはデータと処理をできるだけ「局所的 (local)」に実施すること。
 - 要素マトリクスの計算
 - 有限要素法の計算は本来並列計算向けである
- 「大域的 (global)」な情報を必要とする場合に通信が生じる (必要となる)。
 - 全体マトリクスを線形ソルバーで解く

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

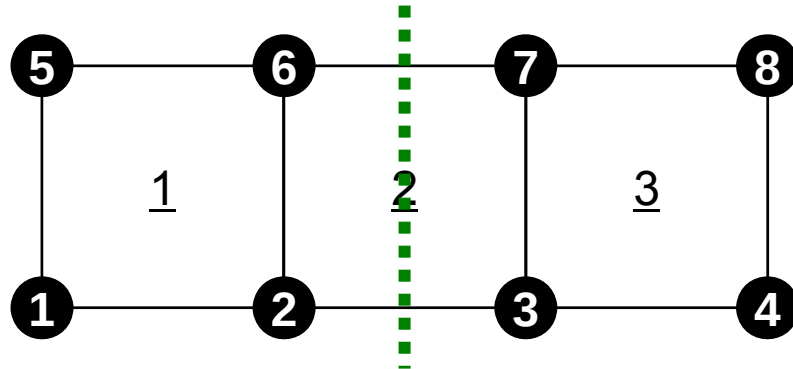


並列有限要素法プログラムの開発

- 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

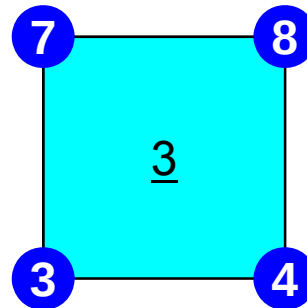
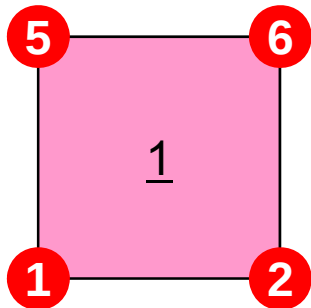
- はじめに
- MPIとは
- 並列有限要素法とは？
 - 基本的な考え方
 - 局所データ構造

四角形要素

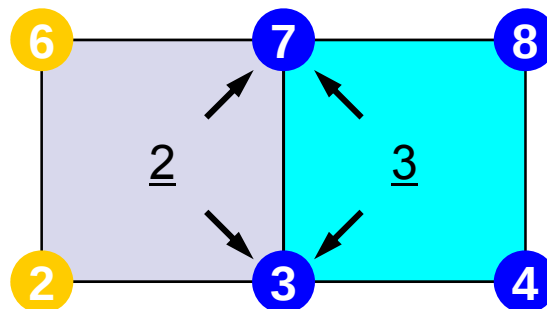
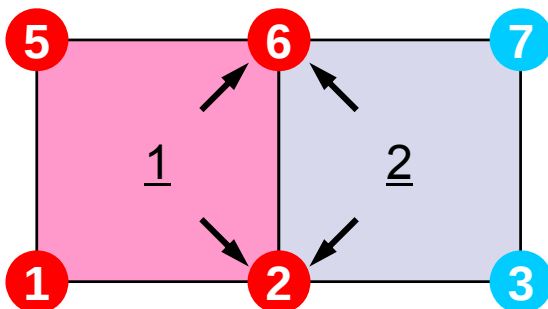


「節点ベース(領域ごとの節点数がバランスする)」の分割

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

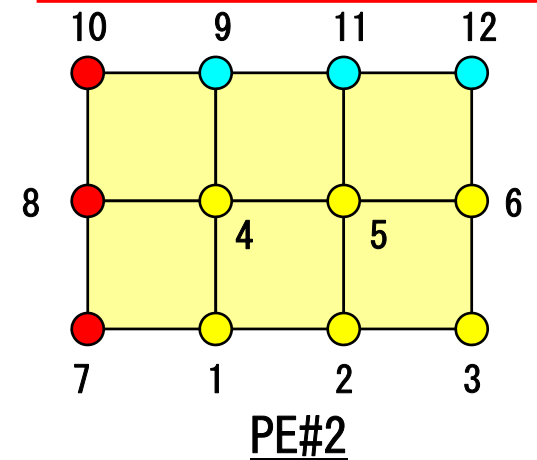
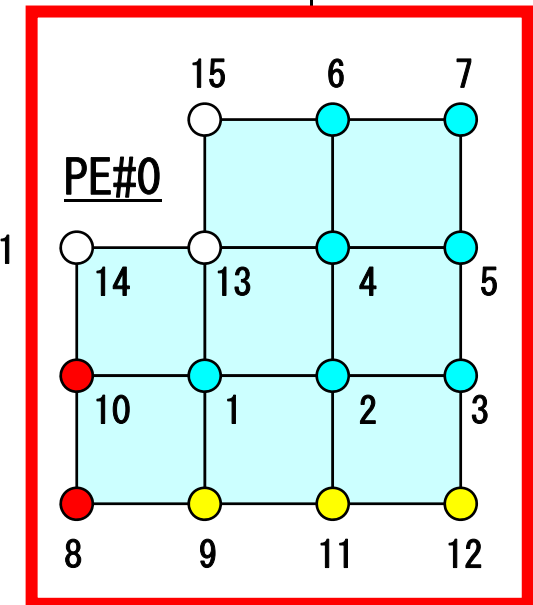
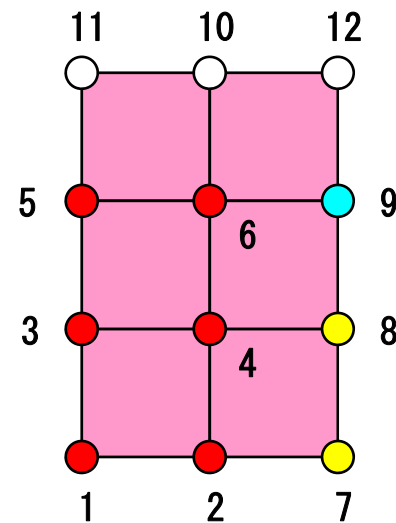
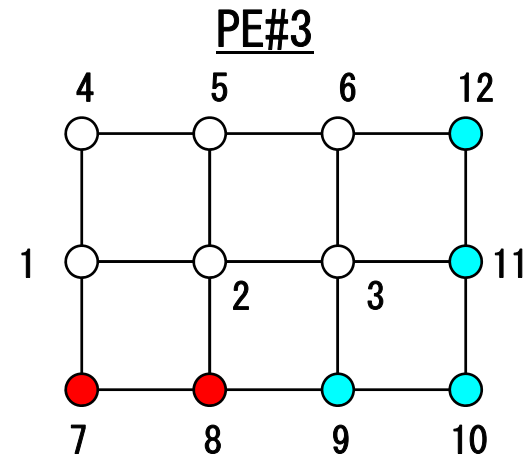
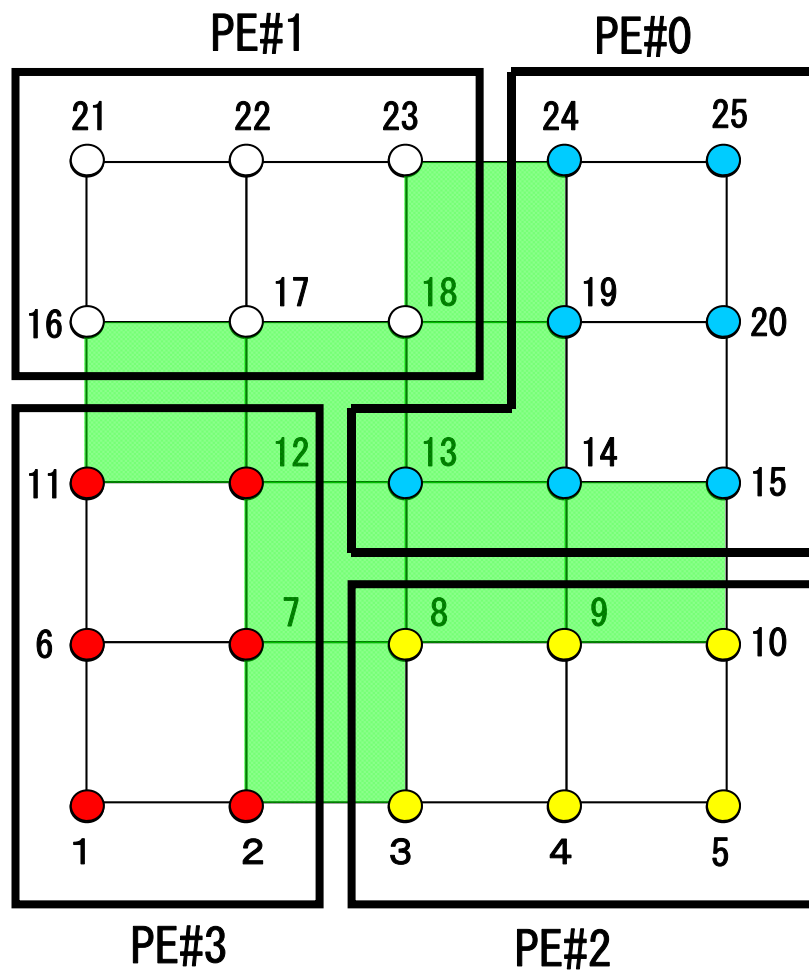


並列有限要素法の局所データ構造

- 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

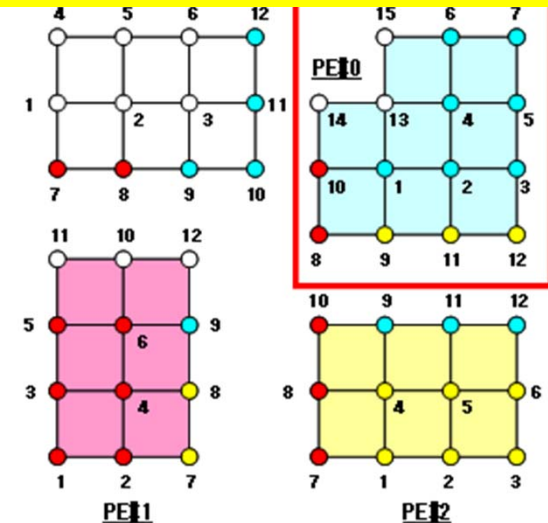
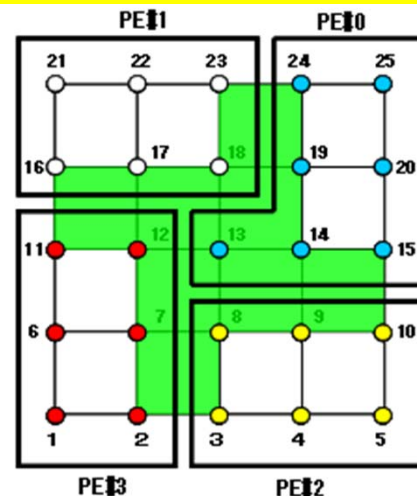
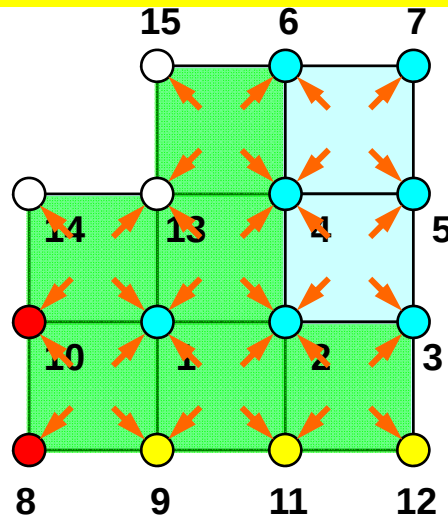


Node-based Partitioning

internal nodes - elements - external nodes

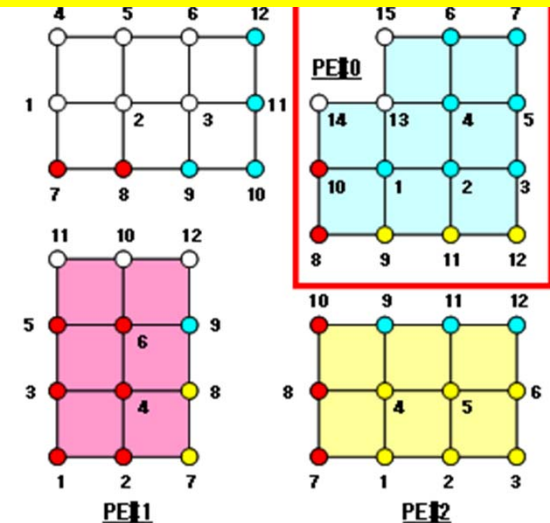
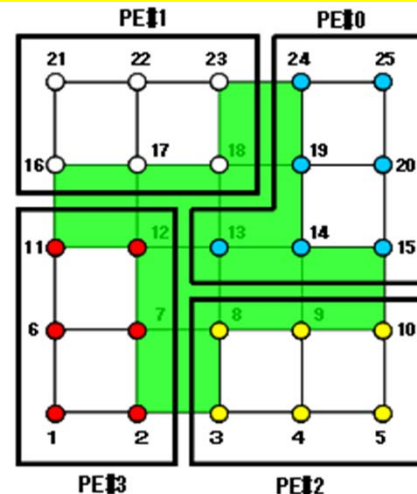
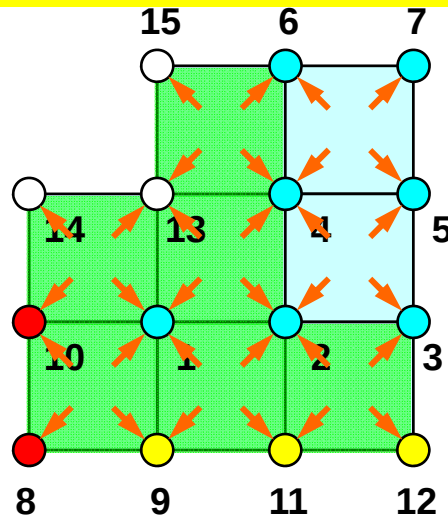


- Partitioned nodes themselves (Internal Nodes) 内点
- Elements which include Internal Nodes 内点を含む要素
- External Nodes included in the Elements 外点
in overlapped region among partitions.
- Info of External Nodes are required for completely local element-based operations on each processor.



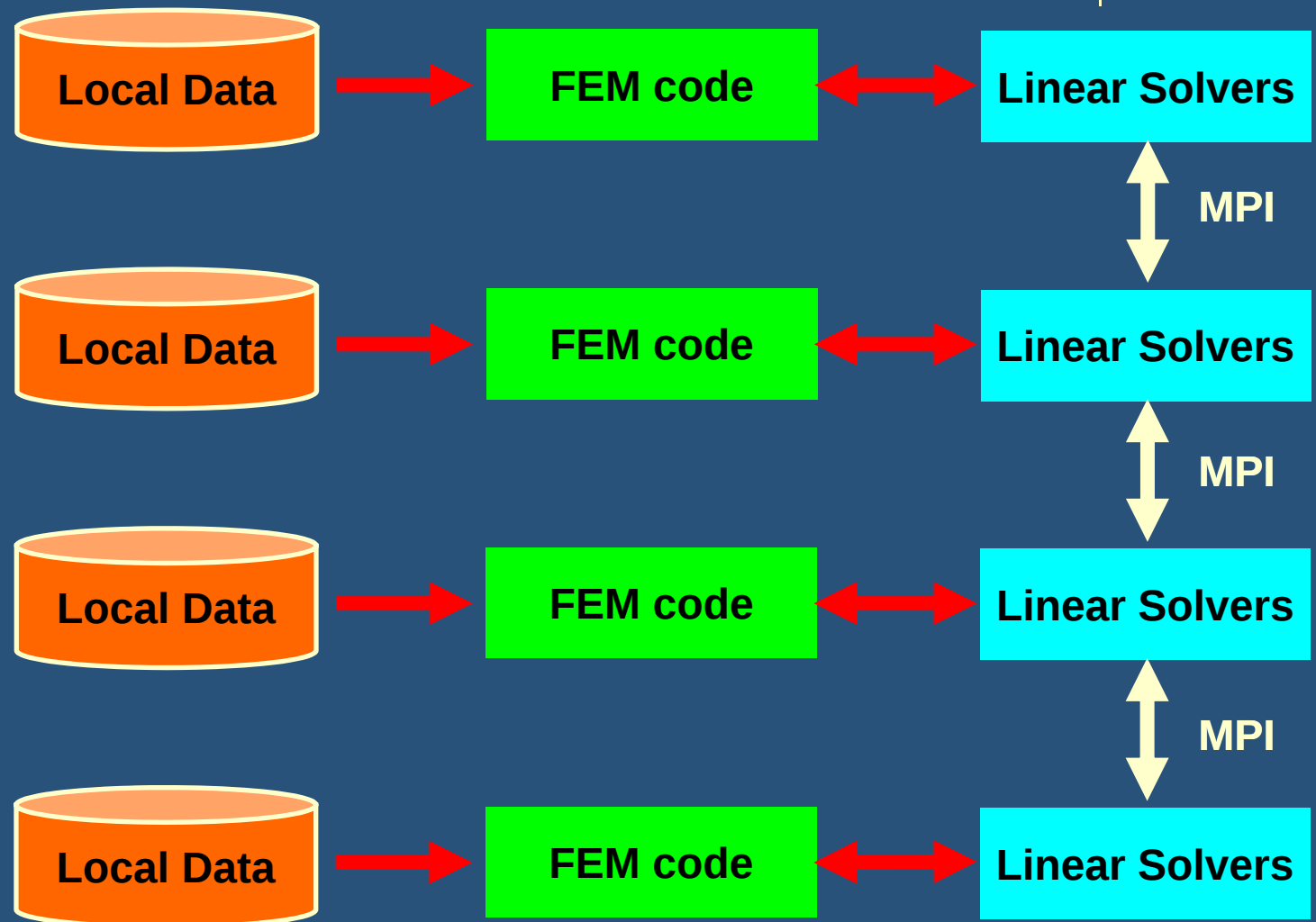
マトリクス生成時の通信は不要

- 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.



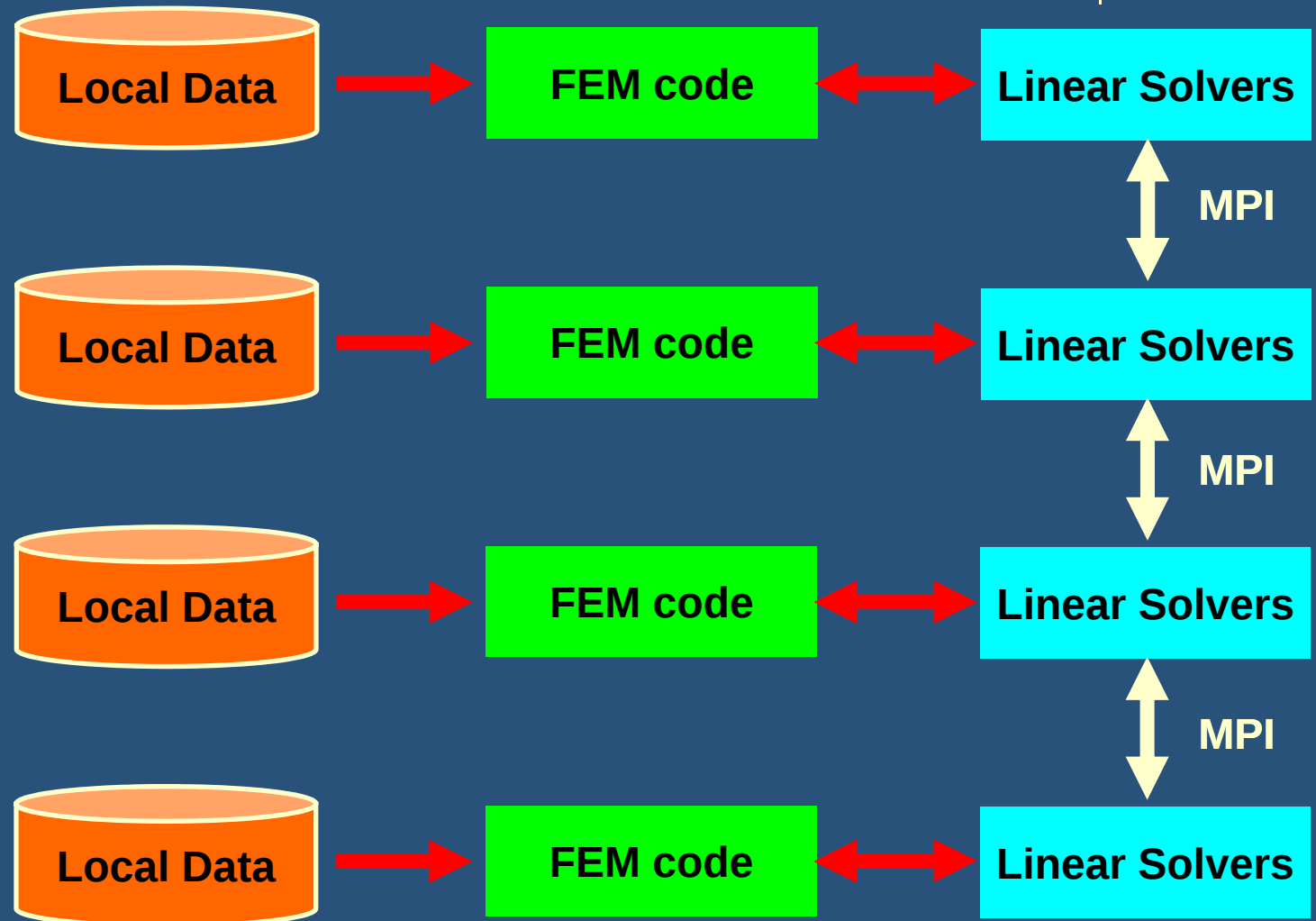
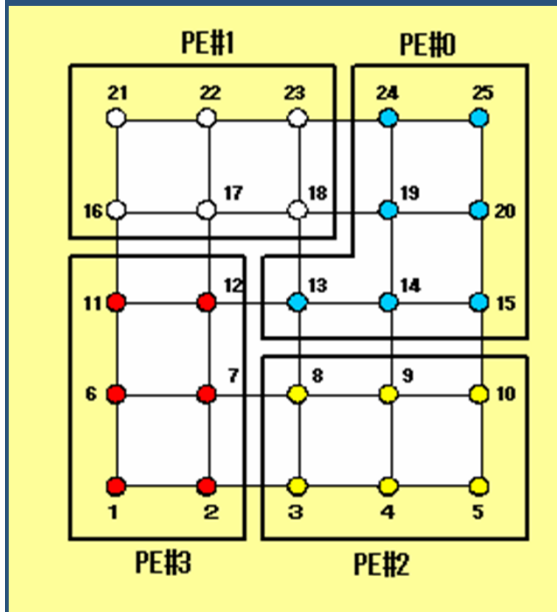
Parallel Computing in FEM

SPMD: Single-Program Multiple-Data



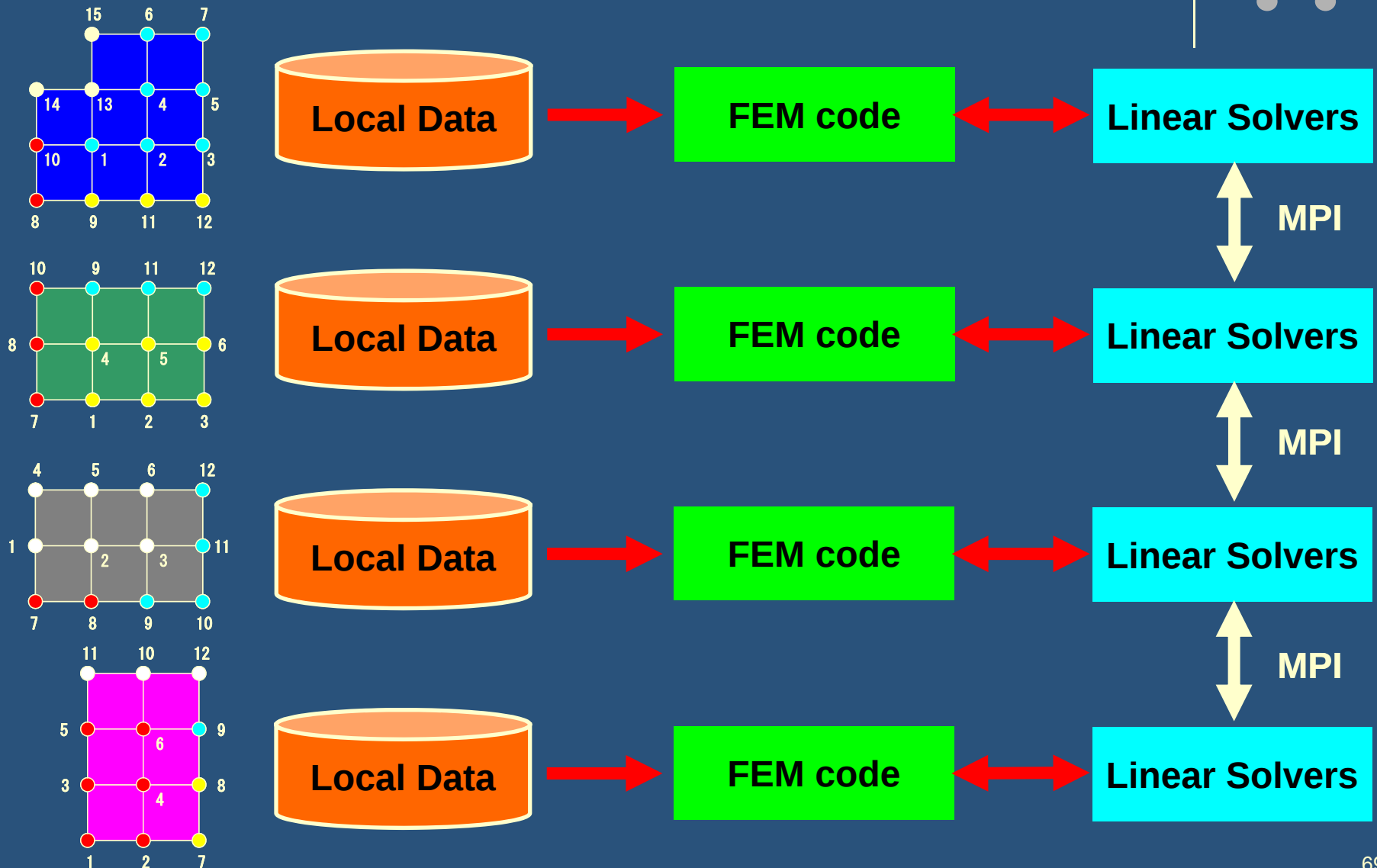
Parallel Computing in FEM

SPMD: Single-Program Multiple-Data



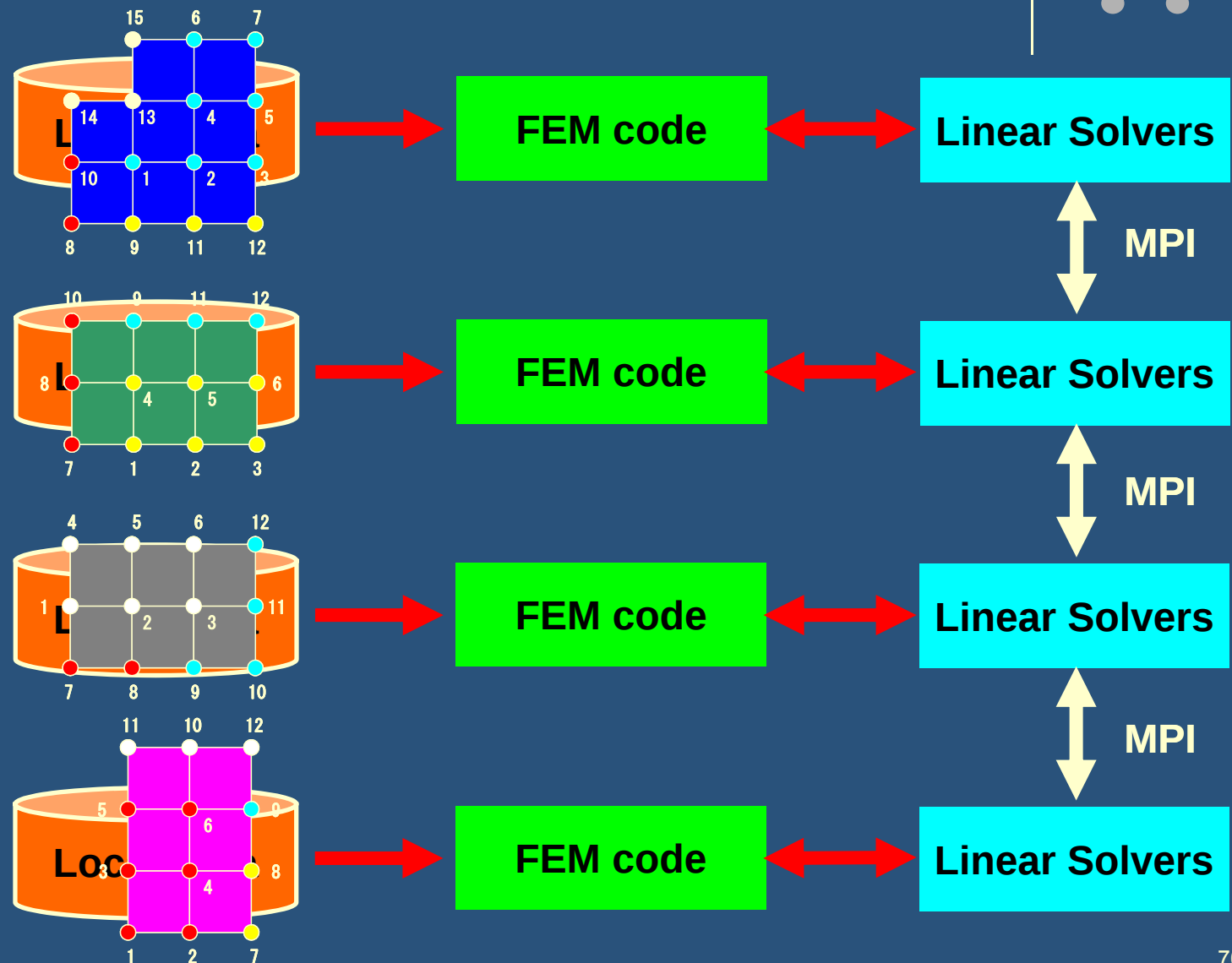
Parallel Computing in FEM

SPMD: Single-Program Multiple-Data



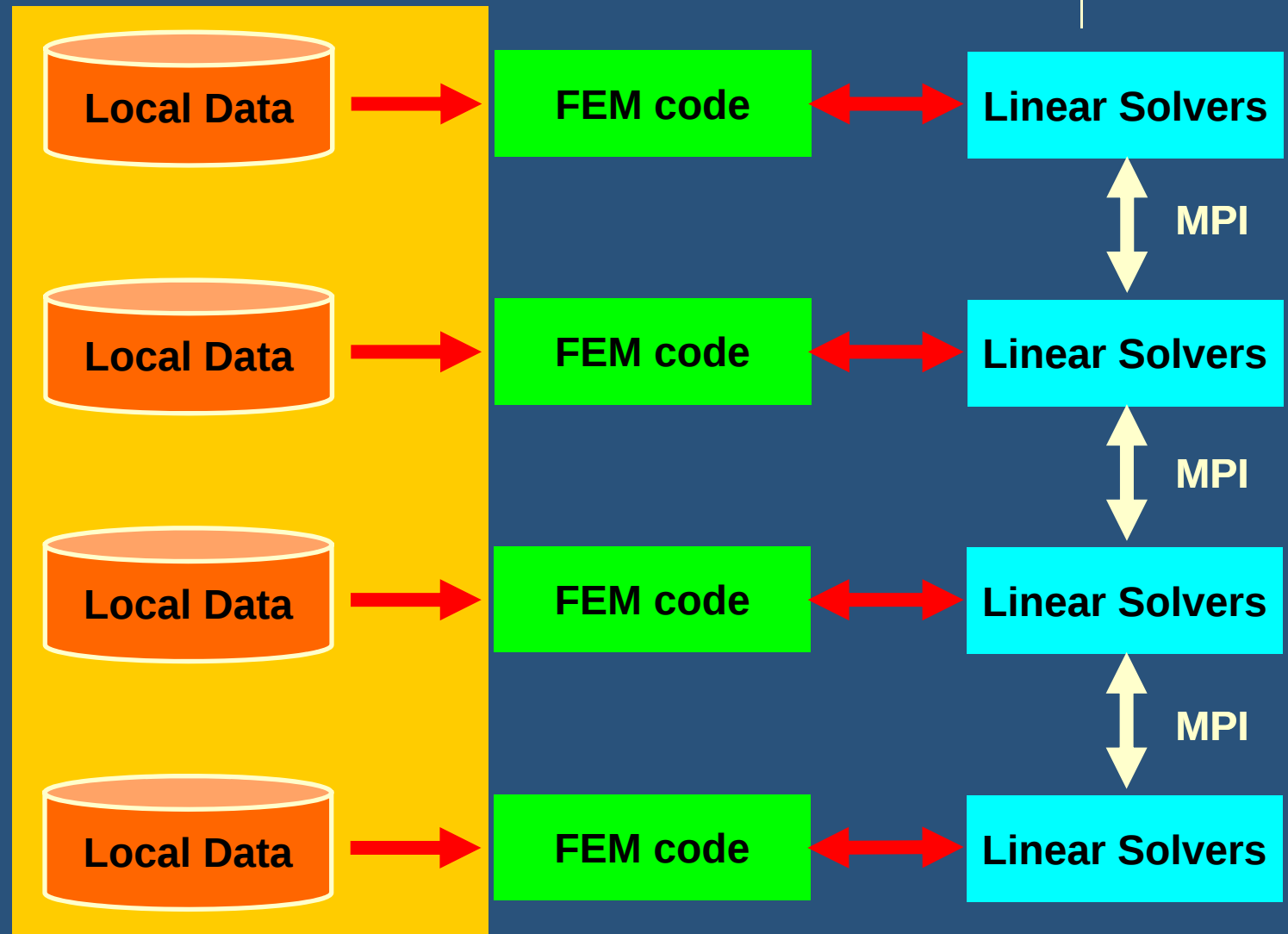
Parallel Computing in FEM

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すること
- 「通信テーブル」にその情報が含まれている