

スーパーコンピュータと統計数理研究所

統計数理研究所

統計科学技術センター センター長

中野純司



目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

スーパーコンピュータの定義(？)

- 「科学技術における高性能計算 (high-performance computing、HPC) を主要目的とする大規模コンピュータ」
- 「その時代の最新技術が投入された最高クラスの性能の計算機」
- 「平成25年度版 政府調達における我が国の施策と実績 -世界に開かれた政府調達へ- -平成26年5月-」の付属資料「別紙2:スーパーコンピュータ導入手続」では「この手続は**50TFLOPS以上の理論的最高性能を有するスーパーコンピュータ**の導入に適用されるが、この対象範囲は必要に応じ見直すこととする。」とある。
 - TFLOPS=Tera Floating-point Operations Per Second
 - 1TFLOPS=1秒間に浮動小数点数演算が1テラ(=10¹², 一兆)回できる
 - 同じ資料で、平成平成17年5月1日から平成26年4月までは「**1.5TFLOPS以上**」であった。

98.3 TFLOPS



207 TFLOPS



28.7 TFLOPS



*Supercomputer System
for Data Assimilation*



*Supercomputer System
for Statistical Science*



*Communal Cloud
Computing System*



2014/09/12

一般知識：コンピュータの基本構成要素

- CPU (Central Processing Unit, 中央処理装置)
 - 演算装置: データの演算を行う装置で、演算に使用する少量の記憶装置(レジスタ)を含む
 - 制御装置: 読み込んだプログラムに沿ってデータを操作する装置
 - 昔はこれらの機能(コア)を一組だけ持っていた。最近は複数のコアを持つ。
- 記憶装置: データを記憶しておく装置
 - 主記憶装置: メモリ
 - 補助記憶装置(ストレージ): ハードディスク、SSD(Solid State Drive)、USBメモリなど
- 入力装置
 - 外部からコンピュータへのデータ入力を行うための装置: キーボード、マウス、イメージスキャナなど
- 出力装置
 - 処理の結果を出力する装置: ディスプレイ(モニタ)、プリンタなど
- ネットワーク装置
 - コンピュータ間を接続するための装置(イーサネット接続が普通)

スーパーコンピュータはどこがすごい？

- CPU: いっぱい持つ
- 記憶装置: いっぱい持つ
- (内部)ネットワーク: 非常に早い
- 入出力装置: 高価なものもある

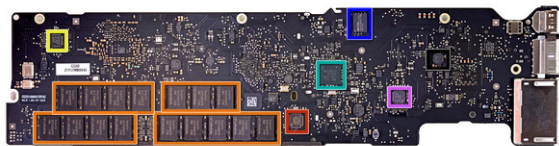
目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

MacBook Air (2012)



基盤裏面



メモリ



MacBook Air (2012)の性能

- CPU: デュアルコア Intel Core i5 最大45GFLOPSくらい
- 記憶装置
 - メモリ 最大 8GB
 - SSD(ハードディスクの代わり) 最大 512GB
- 入力装置
 - キーボード、タッチパッド
 - カメラ
 - マイク
- 出力装置
 - ディスプレイ
 - スピーカー
- (外部)ネットワーク
 - 無線LAN

(注意1) 単位について

キロ(k)= 10^3 (千)、メガ(M)= 10^6 (百万)

ギガ(G)= 10^9 (十億)、テラ(T)= 10^{12} (一兆)

ペタ(P)= 10^{15} (千兆)、エクサ(E)= 10^{18} (百京)

(注意2) 情報の単位

bit=情報の最小単位(0または1)

Byte=8bit (256個の区別ができる)

(例) データ同化スーパーコンピュータシステム Aの性能

- CPU: Intel Xeon、トータルで 5120 コア
最大98.3TFLOPS
- 記憶装置
 - メモリ 128TB
- ハードディスク 816TB
- 入力装置
 - キーボード
- 出力装置
 - ディスプレイ
- 内部ネットワーク
 - NUMalink, InfiniBand

一般知識：コンピュータのソフトウェア

- オペレーティングシステム (OS)
 - (例) Windows, MacOS, Unix, Linux
 - ハードウェアを操作するための基本的なもの
 - 最小限のアプリケーションが含まれる
 - OSは複数の仕事を(複数のコアで)同時に行うことができる
- アプリケーション
 - (例) Word, Excel, PowerPoint, 計算機言語 (C, Fortran, R)
 - OSの上で動き特定の仕事をするためのもの
 - 現在のアプリケーションは並列処理ができるものが多い
- これらに関してはスパコンもパソコンとあまり変わらない

パソコンとスパコン

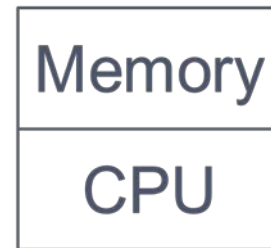
- これまでCPUを含めて「集積回路上のトランジスタ数は18か月(=1.5年)ごとに倍になる」(ムーアの法則と言われることがある)にしたがって発展してきた。すなわち、5年で10倍、10年で100倍である。
- ところが、現在ではいろいろな理由でそのような急速な進歩は難しくなってきた
- さらに、民生用の技術(パソコンを含む)が非常に進み、スパコンと言えど、それらを使う方向になっている
- そこでスパコンでは“超”並列計算を行う

目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

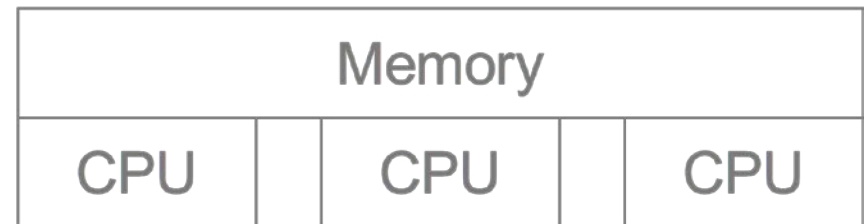
並列計算システム(1)

- 並列型でない“普通の”
計算機

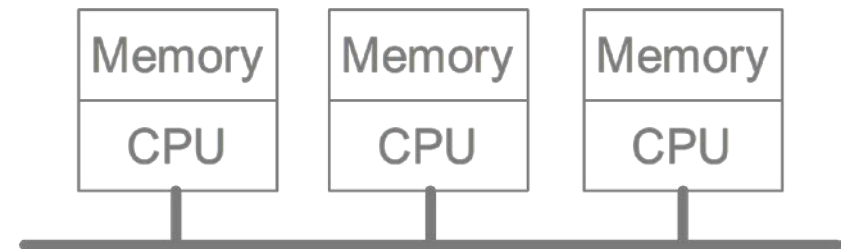


(注意) こ
こでの CPU
は1 core の
場合を考え
る

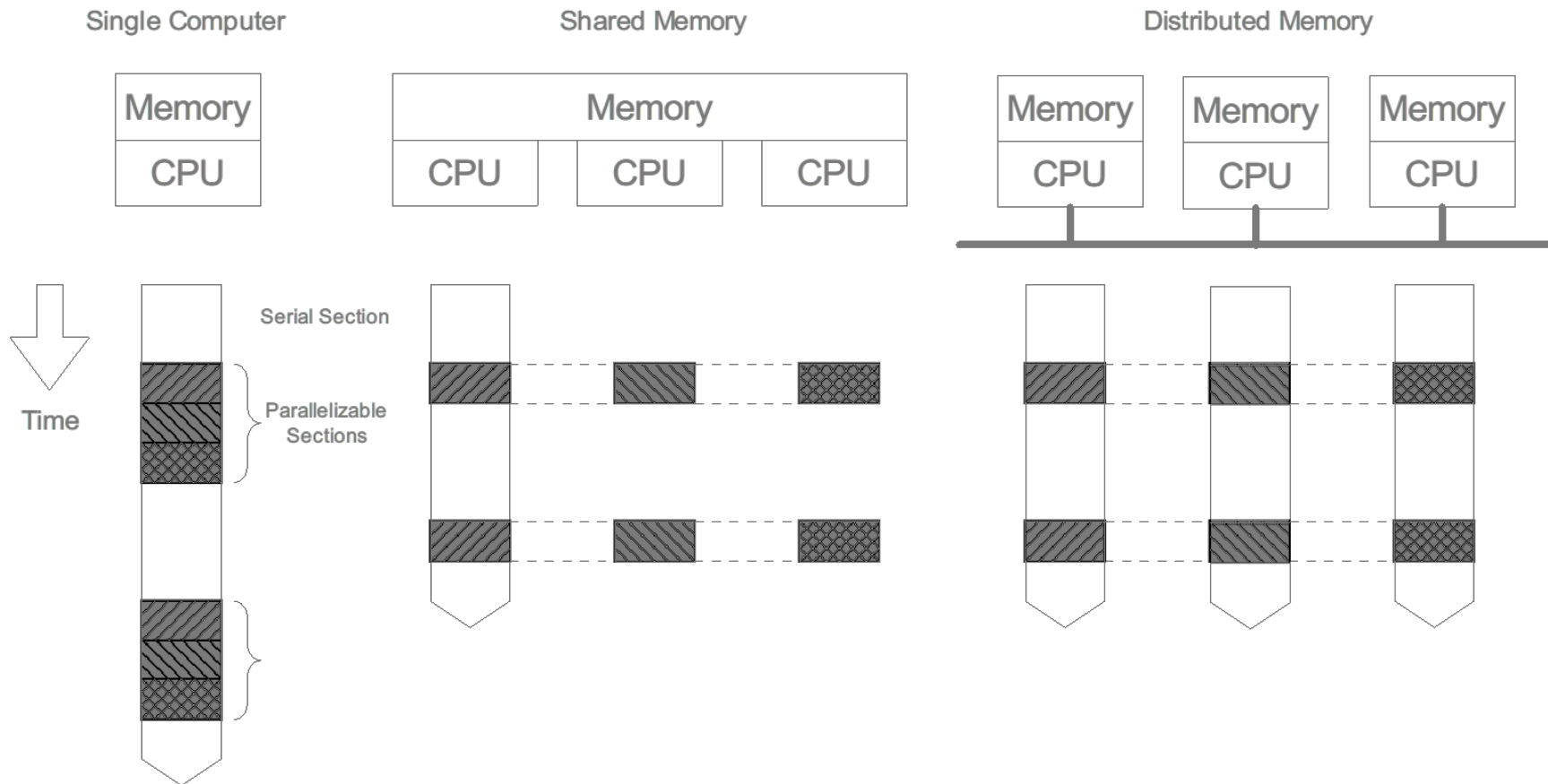
- 共有メモリ型並列システ
ム (Shared memory)



- 分散メモリ型並列システ
ム (Distributed memory)



並列計算システム(2)



並列化の“法則”

- アムダールの法則

- プログラムの並列化不可能な部分の割合を f ($0 < f < 1$)とすると、 n 並列のときのスピードアップ $S(n)$ は

$$S(n) = \frac{1}{f + (1-f)/n}$$

- つまり、並列化不可能な部分が少なくないと効果なし

- グスタフソンの法則

- 例えばシミュレーションを行うとき、準備に s 時間、実行に p 時間かかるとする。 n 回の実行を行うときには、1プロセスでは $s + np$ 、 n プロセスでは $s + p$ かかるのでスピードアップ $S(n)$ は

$$S(n) = \frac{s+np}{s+p}$$

- つまり、並列数を増やすと大規模な解析ができる

並列計算のためのソフトウェア

- 共有記憶システム

- マルチプロセス（プロセス＝一つの仕事）

- Unixが元からサポート

- シングルコアではタイムシェアリングで実現

- マルチスレッド（スレッド＝一つの仕事の一部）

- **OpenMP** (Open Multi-Processing)

- 普通のプログラム(シングルコア用)に指示行を加えることによって並列化する

- ただし、実際にはプログラムを大きく書き直さなければならないことも多い

- 分散記憶システム

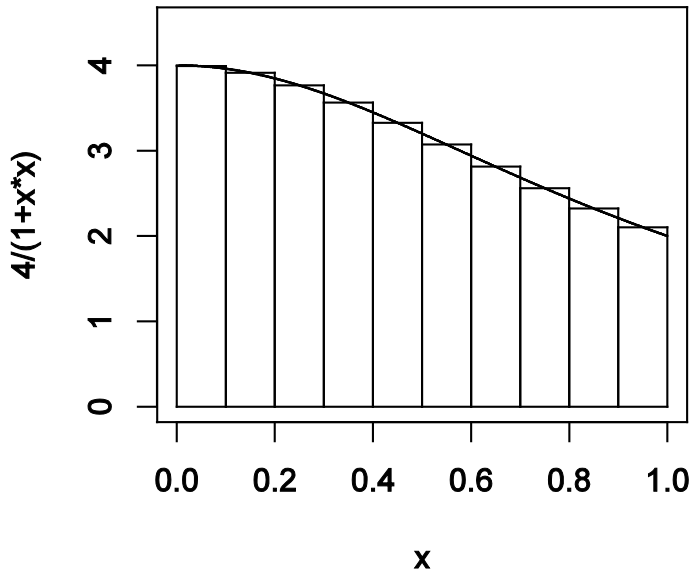
- **MPI** (Message Passing Interface)

- 最初から並列計算を意識してプログラムを組む

並列計算の例

- 簡単な(数値)積分

$$(\pi =) \int_0^1 \frac{4}{1+x^2} dx \sim \frac{1}{n} \sum_{i=1}^n \frac{4}{1 + \left(\frac{i-0.5}{n}\right)^2}$$



曲線で囲まれた面積を求めるためには、小さい長方形の集まりで近似する

普通のFortranプログラム

```
integer n, i
double precision d, s, x, pi
write(*,*) 'n?'
read(*,*) n
d = 1.0/n
s = 0.0
do i=1, n
  x = (i-0.5)*d
  s = s+4.0/(1.0+x*x)
enddo
pi = d*s
write(*,100) pi
100 format(' pi = ', f20.15)
end
```

OpenMP Fortranプログラム

```
integer n, l
double precision d, s, x, pi
write(*,*) 'n?'
read(*,*) n
d = 1.0/n
s = 0.0
!$OMP PARALLEL PRIVATE(x), SHARED(d)
!$OMP& REDUCTION(+: s)
!$OMP DO
  do i = 1, n
    x = (i-0.5)*d
    s = s+4.0/(1.0+x*x)
  end do
!$OMP END DO
!$OMP END PARALLEL
pi = d*s
write(*,100) pi
100 format(' pi = ', f20.15)
end
```

MPI Fortranプログラム

```
include 'mpif.h'
integer n, l
double precision d, s, x, pi, temp
integer myid, numprocs, ierr, status(3)
integer sumtag, sizetag, master
call MPI_INIT(ierr)
call MPI_COMM_SIZE(MPI_COMM_WORLD,numprocs,ierr)
call MPI_COMM_RANK(MPI_COMM_WORLD,myid,ierr)
sizetag = 10
sumtag = 17
master = 0
if (myid .eq. master) then
  write(*,*) 'n?'
  read(*,*) n
  do i = 1, numprocs-1
    call MPI_SEND(n,1,MPI_INTEGER,i,sizetag,
$ MPI_COMM_WORLD,ierr)
  enddo
else
  call MPI_RECV(n,1,MPI_INTEGER,master,sizetag,
$ MPI_COMM_WORLD,status,ierr)
endif
```

```
d = 1.0/n
s = 0.0
do i = myid+1, n, numprocs
  x = (i-0.5)*d
  s = s+4.0/(1.0+x*x)
enddo
pi = d*s
if (myid .ne. master) then
  call MPI_SEND(pi,1,MPI_DOUBLE_PRECISION,
$ master,sumtag,MPI_COMM_WORLD,ierr)
else
  do i = 1, numprocs-1
    call MPI_RECV(temp,1,MPI_DOUBLE_PRECISION,
$ i,sumtag,MPI_COMM_WORLD,status,ierr)
    pi = pi+temp
  enddo
endif
if (myid .eq. master) then
  write(*, 100) pi
100 format(' pi = ', f20.15)
endif
call MPI_FINALIZE(ierr)
end
```

目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

統計科学は

- 不確実性を伴う現象の
 - 客観的な記述を行う(モデリング)
 - データを処理する(データ科学)
 - 理論的考察を行う(数理・推論)
- これらは相互に影響を与え合う

科学の記述手段

- 自然言語（日本語、英語など）
 - もっとも古い
 - あいまいなところがあり冗長である、（が）柔軟である
 - 例： 社会科学、法律、哲学
- 数学
 - 近代では主流（ニュートン以後最近まで）
 - あいまいなところがなく冗長でない、（が）柔軟でない
 - 例： 物理学、理論経済学、数理統計学
- 計算機
 - もっとも新しい
 - あいまいなところはなく少し冗長である、（が）適度に柔軟である
 - 例： 計算 * * 学、計算（機）統計学

統計科学において計算機は

- データの計算、表示のための道具
 - (大量の)データには(大量の)計算処理が必要
 - グラフ(またはグラフィックス)による可視化
- モデルの記述手段
 - プログラムは数式と並び、「複雑」な現象をより「簡単」に記述するための(統計)言語
 - 仮想的な実験ができる: (統計的)シミュレーション
 - 乱数の利用

なぜ統計数理研究所でスーパーコンピュータ？

- データの爆発的な増加
 - データの自動収集
 - インターネット
- データの構造の複雑化
 - ゲノムデータ
 - ネットワーク構造のデータ
 - 地球規模のデータ
- 必要な計算量も爆発的に増大
- 結局、いつでもより強力な計算機が必要になる

本研究所のスパコンで計算されているもの

- データ同化
 - 地球物理的データ
 - 海洋、気候
 - 地震、津波
 - 宇宙物理的データ
 - 地球周辺の電磁気圏の解析、オーロラ
 - その他
- 経済、ファイナンスデータ解析
- 生命科学
 - 分子系統樹
 - ゲノム科学

目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- **統計数理研究所の(スーパー)コンピュータ**
 - 必要性
 - **導入の歴史**
 - 現在の統数研スパコン： A, I, C

大型コンピュータの時代

・前史

- 1956年 国産初の富士通信機製造(現富士通)製商用リレー式計算機 FACOM 128A の1号機が納入された
- 1963年 3月30日に電子計算機が設置された(機種不明)
- 1971年 2月に新型電子計算機が稼働を開始(機種不明)

・統計科学計算機システム

- 1980年 HITACH M280H (主記憶 24MB)
- 1989年 HITACH M682H (主記憶 256MB)
HITACH M660D (主記憶 64MB)
- 1994年 HITACH M880/180 (主記憶 512MB)
HITACH S-3600 (主記憶 256MB,理論最大性能 205MFLOPS)

スーパーコンピュータの時代

- ・ 統計科学スーパーコンピュータシステム

1999年 HITACHI SR8000 (20ノード、主記憶 160GB、理論最大性能 160GFLOPS)

2004年 SGI Altix3700 (256CPU、主記憶 1920GB、理論最大性能 1.3TFLOPS)

2010年 Fujitsu SPARC Enterprise M9000(2システム

64CPU(256コア)、主記憶2TB、理論最大性能 2.9TFLOPS、
24CPU(96コア)、主記憶1TB、理論最大性能 1.1TFLOPS)

Fujitsu PRIMERGY RX200S5(360ノード(2880コア)、主記憶 12.4TB、
理論最大性能 33.75TFLOPS)

- ・ 計算統計学支援システム

1996年 IBM RS/6000 SP(48ノード 主記憶12GB、理論最大性能 12.7GFLOPS)

2000年 Origin2000(64CPU、主記憶48GB、R12000(300MHz)
理論最大性能 38.4GFLOPS)

2006年 HP XC4000 (ProLiant DL 145G2 128ノード、Opteron 2.6GHz × 2、
主記憶640GB、理論最大性能 1.3TFLOPS)

データ同化スーパーコンピュータシステム “A”



統計科学スーパーコンピュータシステム “I”



共用クラウド計算システム “C”



*Supercomputer System
for Data Assimilation*



*Supercomputer System
for Statistical Science*



*Communal Cloud
Computing System*

(注意) 統計学において重要なモデル選択規準として赤池情報量規準
(An Information Criterion, **AIC**)がある

2014/09/12

目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

データ同化スーパーコンピュータシステム “A”



- シングルシステムとして世界最大の共有メモリを搭載した SGI 社製大規模共有メモリ型サーバ SGI UV 2000 2台で構築
- 1パーティション 2560 コア、メモリ容量 64TB
- 1パーティションをHPCI (High Performance Computing Infrastructure) システムに資源提供している

「京」と全国の大学や研究機関に設置されたスパコンを高速ネットワーク (SINET4) で結び、多様なユーザーニーズに応える革新的な共用計算環境を実現しています。

A 主要スペック

CPU: Intel Xeon E5-4650v2 5120 コア

主記憶: 128TB

ストレージ: Lustre サーバ 816TB

(アクセラレータ Intel Xeon Phi 5110P)

理論性能値: 98.3TFLOPS

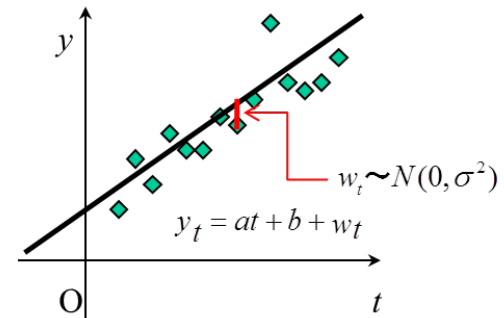


データ同化とは？(1)

要するに、
観測データにモデルをあてはめること

$$p(y_t | \theta = [a, b, \sigma^2])$$

$$E = \sum_{t=1}^T \{y_t - (a \cdot t + b)\}^2 / \sigma^2$$



データに直線をあてはめる

データにシミュレーションモデルをあてはめる

= 「シミュレーションモデルにデータを同化する」


データ同化

複雑なモデルを扱うことになるのが特徴

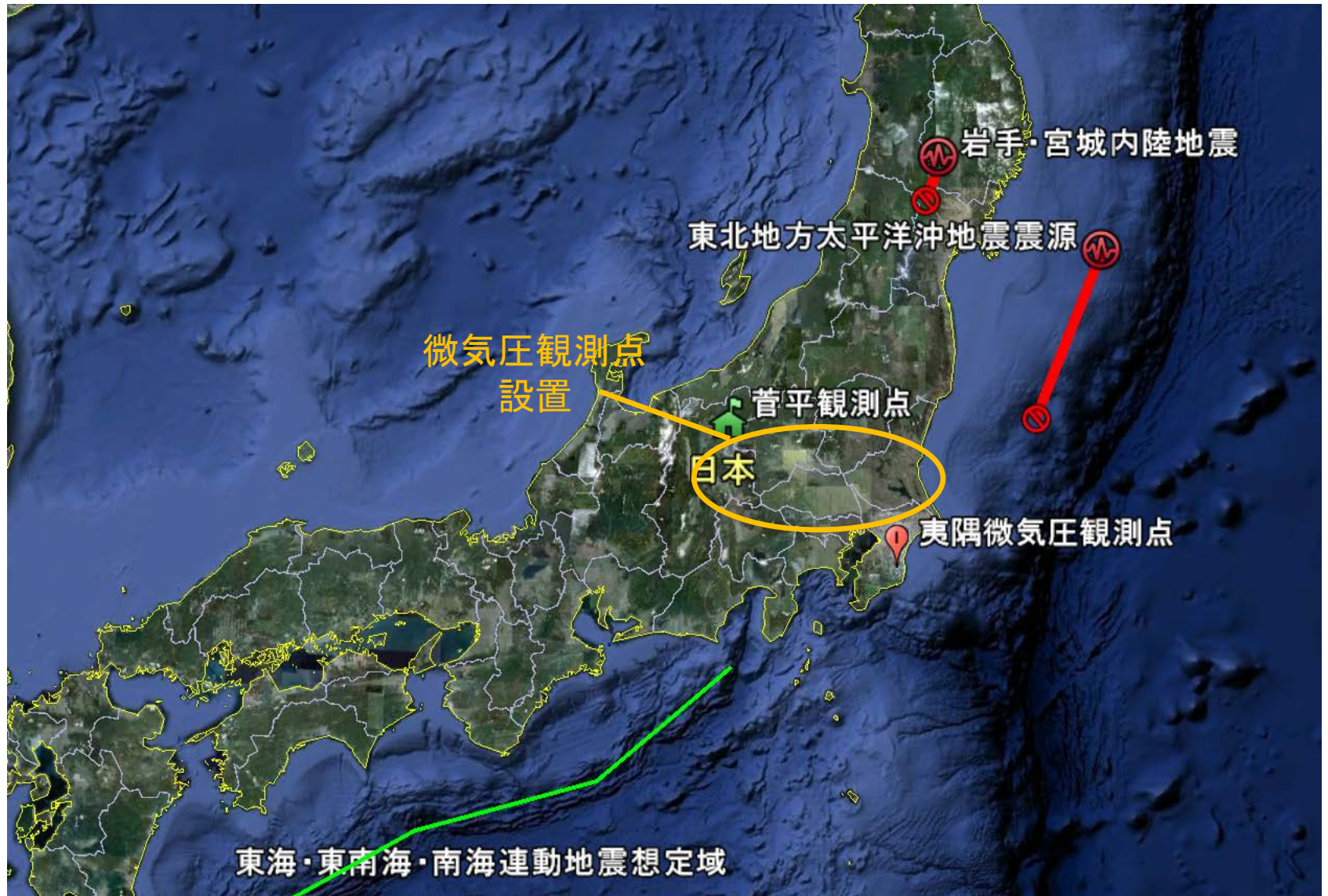
気象・海洋分野での“方言”

データ同化とは？(2)

- 気象学・海洋学の分野で発達（1990年代中頃から）
- 物理数値シミュレーションモデルと実際の観測を統合する手法
 - シミュレーションのみでは適切に現実の物理現象を再現できない
 - シミュレーションモデルには、モデルの不完全性や境界条件が正確にはわからないなどの不確かさが存在
 - (たとえば) 正確な気象予測には適切な初期条件の構成が必要
 - 観測データは物理的・社会的制約により得られる情報に限界がある

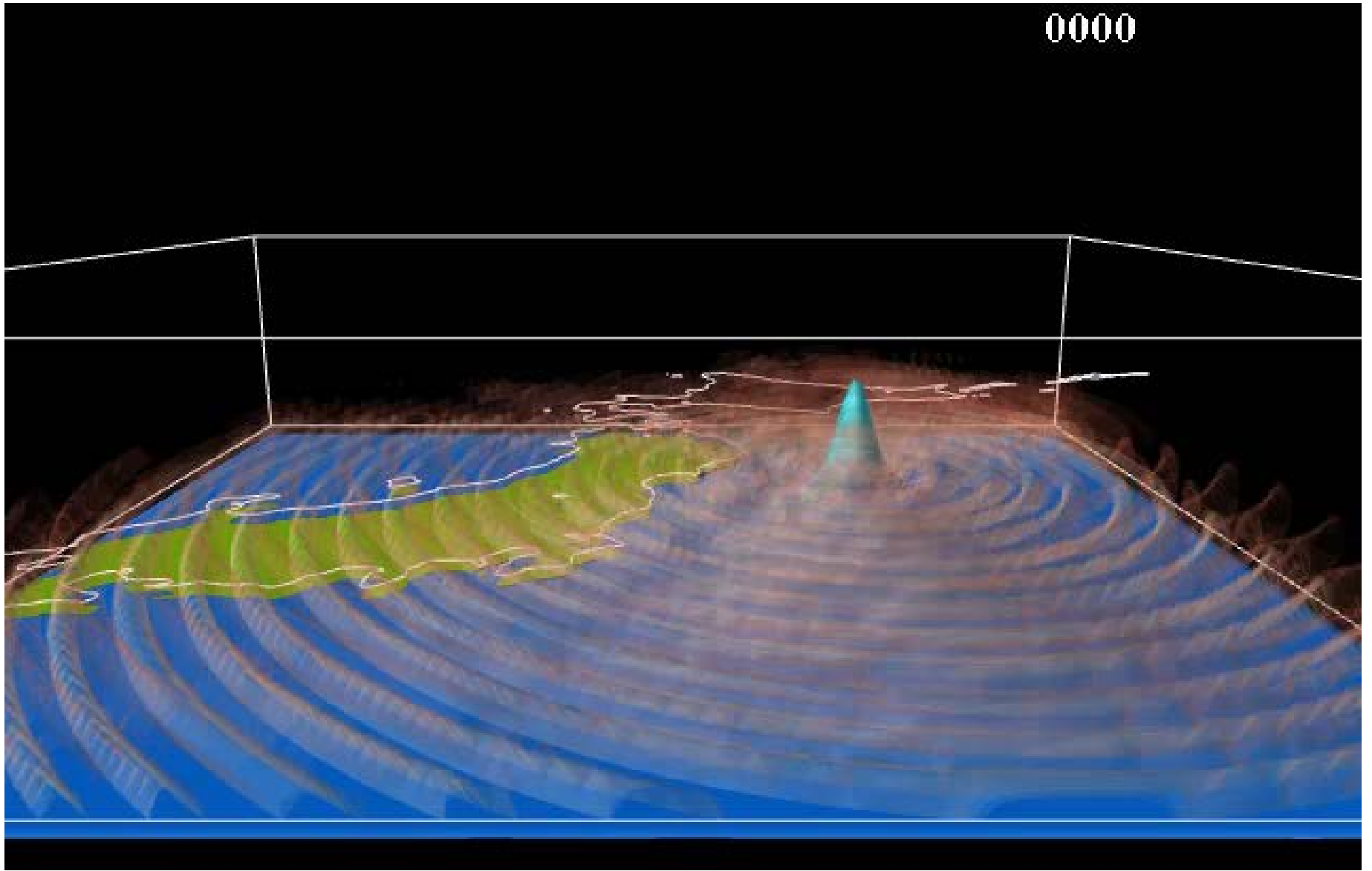
 観測データを用い数値シミュレーション内の変数を修正
= データ同化

地震音波検出を目的とした微気圧観測



東日本大震災による地震音波強度や伝搬特性を評価

0000



目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

統計科学スーパーコンピュータシステム “I”

Supercomputer System
for Statistical Science

- **分散メモリ型**のスパコンで本研究所最高速
- SGI 社製 ICE X を中心とする
 - 400 の計算ノード (CPU: Intel Xeon E5-2697v2 x 2、主記憶: 128GB)
 - アクセラレータ (Intel Xeon Phi) 付きのノードは 32 ノード
- 物理乱数発生装置が備わっている
 - 秒間 528MB の発生速度を持つ 3 台のサーバで構成される
- なお、2015 年度中に増強される (以下の表参照)



I 主要スペック (増強後)

CPU	Intel Xeon E5-2697 9600 コア (12960 コア)
主記憶	50TB (100TB)
ストレージ	Lustre サーバ 2.5PB
アクセラレータ	Intel Xeon Phi 5110P
理論性能値	207TFLOPS (336TFLOPS)

目次

- スーパーコンピュータとは
 - いったい何？
 - 本当に「スーパー」？： ノートパソコンとの比較
 - どのように使う？： 仕組みとソフトウェア
- 統計数理研究所の(スーパー)コンピュータ
 - 必要性
 - 導入の歴史
 - 現在の統数研スパコン： A, I, C

共用クラウド計算システム “C”



- データ解析HPCを目的とするプライベートクラウドシステム
- HPC向けのチューニングを行い、データ解析ソフトウェアがすぐ使えるパーソナルなHPC環境を提供する
- 標準で提供する環境は
 - 8 core CPU 4ノード
 - 各ノードでメモリは64GB
 - ストレージは5TB
 - MPIや統計解析環境Rがインストールされていて、すぐ使い始めることができる



C 主要スペック

CPU	Intel Xeon E5-2680v2 1380 コア
主記憶	16.4TB
ストレージ	GPFS サーバ 364TB
理論性能値	28.7TFLOPS

一般知識：クラウド（またはクラウドコンピューティング）

- ネットワーク、特にインターネットをベースとしたコンピュータ資源の利用形態
- ユーザーは、コンピュータによる処理やデータの格納などをネットワーク経由で、サービスとして利用する。
- コンピュータ本体およびネットワークの購入・管理運営費用が軽減される
- 必要な時に使えばよいので、資源を遊ばせる無駄がない

HPCとクラウド

- 現在のクラウド(ソフトウェア)は主としてビジネス用途を考えて設計されている
 - 対話的使用
 - 人間が考えている間はCPUはほとんど待ち状態
 - そこで一つのCPUに複数人のプロセスを割り当てても良い
 - ひとりが1つのCPUを使うのは無駄
- HPCはその対極
 - バッチ使用
 - 仕事を計算機に投げたら計算が終わるまで人間は関与しない
 - CPUはほとんどフル稼働状態
- したがってクラウドでHPCを行うためにはまだまだ研究が必要
 - 現状では実機と比べてパフォーマンスがよくない
 - しかし、チューニング次第で90%を超えることも可能