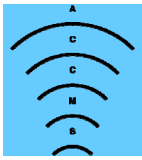


計算科学が拓く世界

スーパーコンピュータは何故スーパーか

学術情報メディアセンター
中島　浩

http://www.para.media.kyoto-u.ac.jp/jp/
username=super　password=computer

講義の概要

- 目的
  - 計算科学に不可欠の道具スーパーコンピュータが
    - どういうものか
    - なぜスーパーなのか
    - どう使うとスーパーなのか

    について雰囲気をつかむ
- 内容
  - スーパーコンピュータの歴史を概観しつつ
  - スーパーである基本原理を知り
  - どういう計算が得意であるかを学んで
  - それについてレポートを書く

スーパーコンピュータ (スパコン) とは (1)

- パソコンの数千倍～数万倍の規模・性能を持つ巨大な超高速コンピュータ
  - 世界最大・最高速マシン ≒パソコン x 10万
  - 京大スーパーコンピュータ ≒パソコン x 5千

  ➔パソコンで丸1日かかる計算＝1秒～数10秒

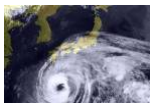
  （ただしスパコン向きの問題をうまくプログラムしたら）
- スパコンが高速な理由
  - 個々の部品（CPU, メモリなど）≒パソコン
  - 非常に多数のパソコン（のようなもの）の集合体
    - パソコン = 1~4 CPU
    - 京大スパコン = 6,656 CPU
    - 世界最高速スパコン = 86,016 CPU + 100,352 GPU
    - 世界最大規模スパコン = 294,912 CPU

スーパーコンピュータ (スパコン) とは (2)

- スパコンが得意な計算＝大量CPUによる分担計算＝超大量のデータを対象とする計算
  - 地球全体の気象・気候・海洋現象の予測

    ➔1km^2 あたり1データ

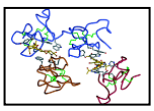
    ➔データ数≒5億（x 高さ方向）



  - 生体物質・化学物質・材料の解析

    ➔膨大な分子・原子数

    （e.g. 水1ml = 3.3 x 1兆 x 100億）



  - 自動車の空力・衝突解析

    ➔1mm^2 or 1cm^3 あたり1データ

    ➔データ数＝1~10億

  - Web 文書の解析

    （←自動翻訳用データ作成など）

    ➔文書数＝数億～数10億

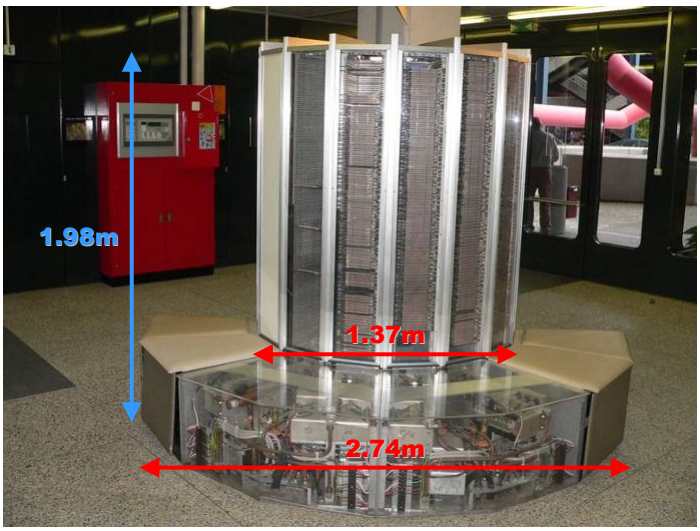
スーパーコンピュータの歴史

そもそもの始まり：ベクトルマシン (1)

- 1976年：最初のスパコンCray-1登場
  - 動作周波数=80MHz (< 携帯電話)
  - 演算性能=160MFlops (< 携帯電話)
    - Flops: Floating-point Operation Per Second
    = 10進16桁精度の数値(10^{-308}~10^{308})の加減乗算回数／秒
    ➔160MFlops = 毎秒1億6千万回の加減乗算
  - 消費電力=115kW
  - 大量の数値データ(ベクトル)に対する同種演算が得意
- 1976年(中島=20歳)での「スーパー」度
  - 最速@京大(富士通 F230-75) < 5MFlops
  - 最速@京大情報工学科(日立 H8350) < 1MFlops
  - Intel 8086/87(1978/80) ≒ 50KFlops

スーパーコンピュータの歴史

そもそもの始まり：ベクトルマシン (2)



source: http://en.wikipedia.org/wiki/Image:Cray-1-p1010221.jpg

ベクトル計算の原理 (1)

- 大量数値データの同種演算を高速に行う方法

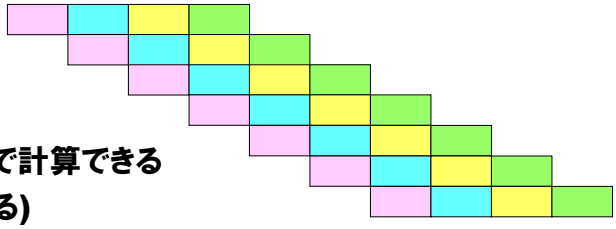
  例: $z_i = x_i \times y_i \quad (i = 1, 2, \ldots)$

  - 1つの乗算をいくつか(たとえば4つ)の小さい操作に分ける

    $z_i = x_i \times y_i$ 

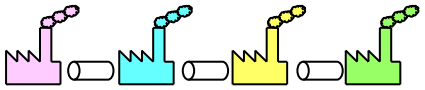
  - 多数の乗算を1小操作ずつずらして行う

    $z_1 = x_1 \times y_1$
    $z_2 = x_2 \times y_2$
    $z_3 = x_3 \times y_3$
    $z_4 = x_4 \times y_4$

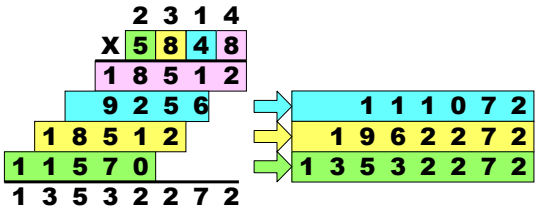
  ➔4倍の速度で計算できる
  (ように見える)

  ➔(演算)パイプライン処理

ベクトル計算の原理 (2)

- 乗算を4分割してずらす考え方(たとえ話≠真実)



2314x5848=13532272

2314x8	+2314x4	+2314x8	+2314x5

7872x6752=53151744

7872x2	+7872x5	+7872x7	+7872x6

1778x7142=12698476

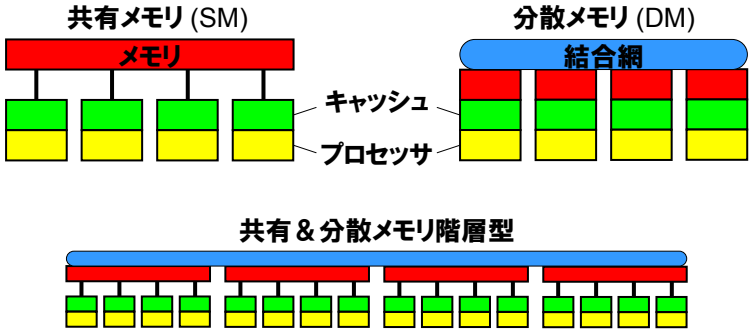
1778x2	+1778x4	+1778x1	+1778x7

8485x1651=13843635

8385x1	+8385x5	+8385x6	+8385x1

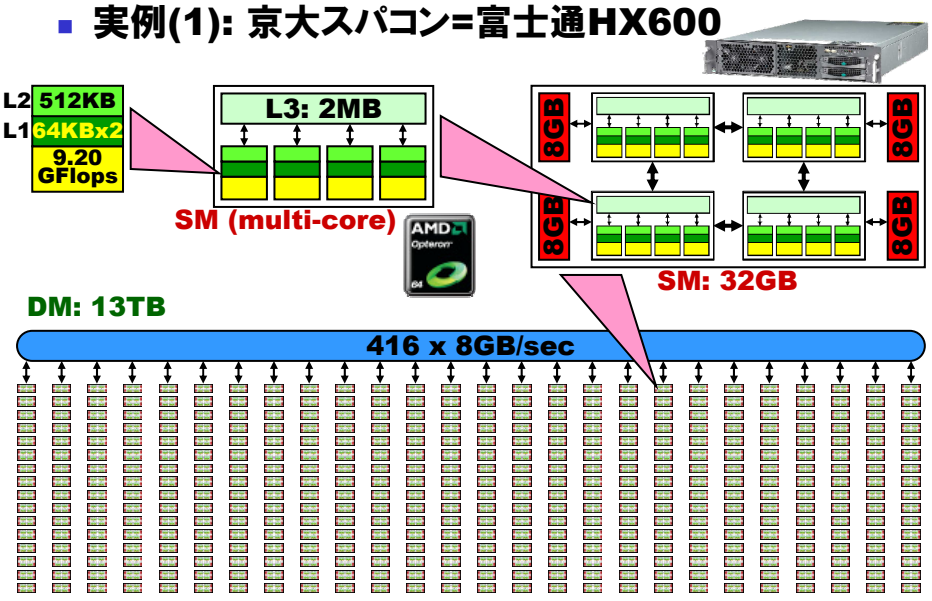
スーパーコンピュータの歴史(に戻って)
もう一つの方法:並列マシン(1)

- 1980年代: スカラーマルチプロセッサ台頭
  - 多数のパソコン(のようなもの)の集合体
  - Sequent Balance : 20 x NS32016 ('84)
  - Intel iPSC/1: 128 x i80286 ('85)



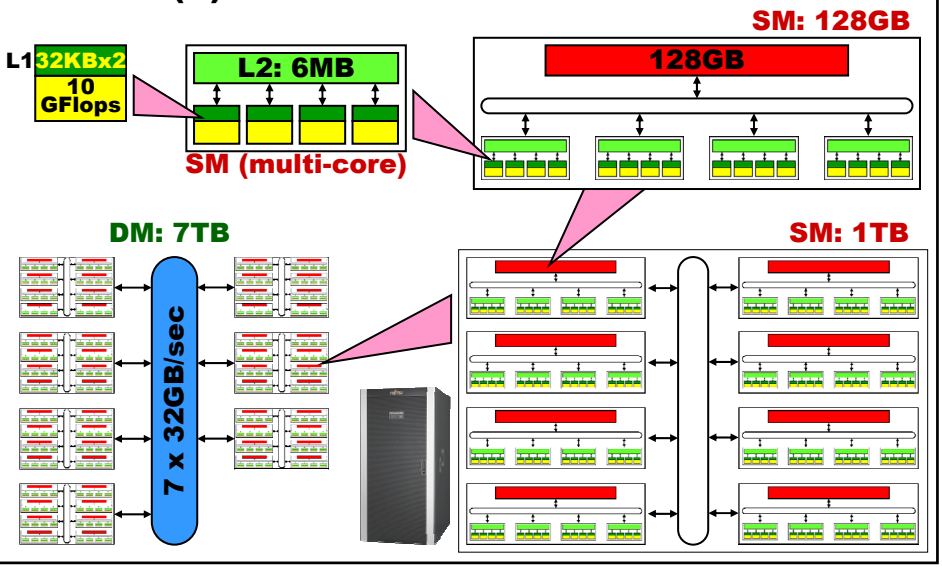
スーパーコンピュータの歴史
もう一つの方法:並列マシン(2)

- 実例(1): 京大スパコン=富士通HX600



スーパーコンピュータの歴史
もう一つの方法：並列マシン(3)

- 実例(2): 京大スパコン=富士通SE M9000



スーパーコンピュータの歴史
もう一つの方法：並列マシン(4)

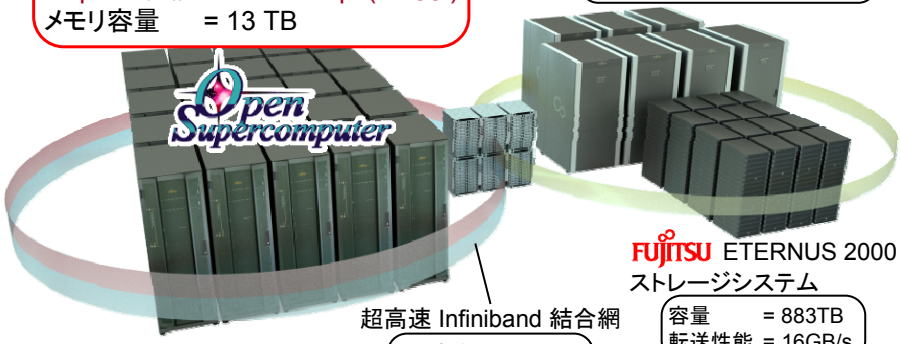
- 京大スパコンの全体像

FUJITSU HX600 クラスタ

ノード数 = 416
コア数 = 16 x 416 = 6656
ピーク性能 = 61.2 TFlops
Linpack 性能 = 50.5 TFlops(#158)
メモリ容量 = 13 TB

FUJITSU SPARC Enterprise M9000
fat node サブシステム

ノード数 = 7
コア数 = 128 x 7 = 896
ピーク性能 = 8.96 TFlops
メモリ容量 = 1TB x 7 = 7 TB



FUJITSU ETERNUS 2000
ストレージシステム

容量 = 883TB
転送性能 = 16GB/s

超高速 Infiniband 結合網
通信性能 = 3.3TB/s

また少し寄り道：スーパーコンピュータの原理
並列計算の原理

- **大量CPUによる同じ(ような)計算の分担方法**

  例： $z_i = x_i \times y_i \quad (i = 1, 2, \ldots, n)$

  - n 組のデータを p 個のCPUに均等に分割する
  - それぞれのCPUが割当てられた計算をする
  - おしまい

$$z_1 = x_1 \times y_1,\ z_2 = x_2 \times y_2,\ \vdots,\ z_{n/4} = x_{n/4} \times y_{n/4}$$

$$z_{n/4+1} = x_{n/4+1} \times y_{n/4+1},\ z_{n/4+2} = x_{n/4+2} \times y_{n/4+2},\ \vdots,\ z_{2n/4} = x_{2n/4} \times y_{2n/4}$$

$$z_{2n/4+1} = x_{2n/4+1} \times y_{2n/4+1},\ z_{2n/4+2} = x_{2n/4+2} \times y_{2n/4+2},\ \vdots,\ z_{3n/4} = x_{3n/4} \times y_{3n/4}$$

$$z_{3n/4+1} = x_{3n/4+1} \times y_{3n/4+1},\ z_{3n/4+2} = x_{3n/4+2} \times y_{3n/4+2},\ \vdots,\ z_n = x_n \times y_n$$

スーパーコンピュータの歴史（にまた戻って）
ベクトル vs 並列 (1)

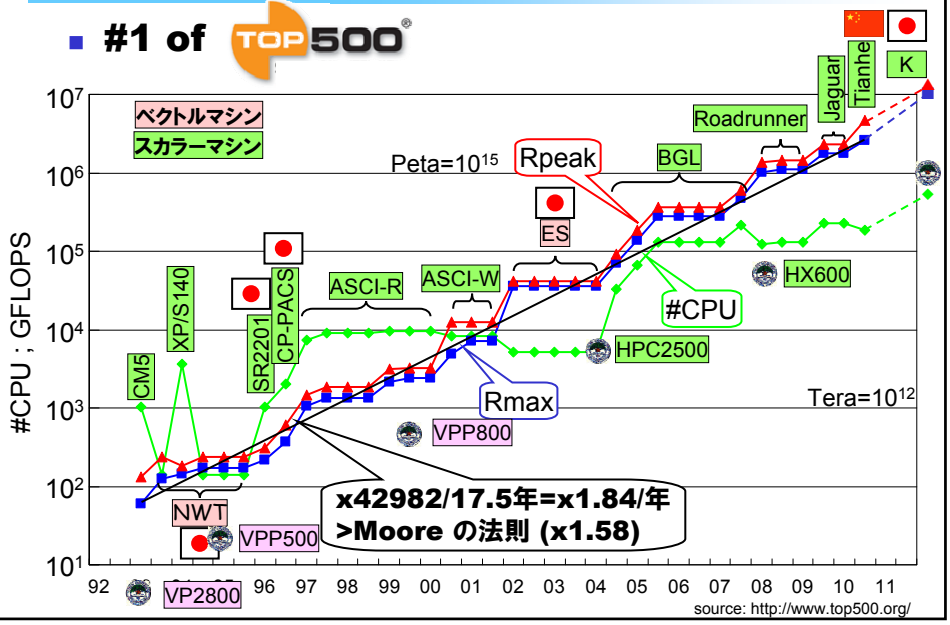
- **1990年代：ベクトル並列 vs スカラー並列**
  - **TOP-10 of TOP500® @ 1993.6**

machine	#proc	Rmax	Rpeak
TMC CM-5	1024	59.7	131.0
TMC CM-5	544	30.4	69.6
TMC CM-5	512	30.4	65.5
TMC CM-5	512	30.4	65.5
NEC SX-3	4	23.2	25.6
NEC SX-3	4	20.0	22.0
TMC CM-5	256	15.1	32.8
Intel Delta	512	13.8	20.5
Cray Y-MP	16	13.7	15.2
Cray Y-MP	16	13.7	15.2

- 巨大で(>100万元)密な連立一次方程式の求解性能に基づく世界中のスパコン順位表
- 1993.6から毎年2回発表 (6月&11月)
- Rmax: 求解性能
  Rpeak: 理論最大性能
  (単位GFlops：毎秒10億演算)

スーパーコンピュータの歴史
ベクトル vs 並列 (2)

- #1 of TOP500



source: http://www.top500.org/

スーパーコンピュータの原理
（いきなり&とりあえず）まとめ

- ベクトルマシン
  - 1つの演算を k 個の小さい操作に分割する
  - 多数の同種演算を1小操作ずつずらして行う

  ➔k 倍の速度で計算できる(ように見える)

  ➔大量 ($\gg k$) の同種演算が得意
- 並列マシン
  - 多数の同じ(ような)演算を p 個のCPUに分割
  - それぞれのCPUが割当てられた計算をする

  ➔p 倍の速度で計算できる(ように見える)

  ➔大量 ($\gg p$) の同じ(ような)演算が得意

➔スパコンは大量の同じ(ような)演算(や処理)が得意

大量同種演算は何でも得意か? (1)

- **超得意**

  $z_i = x_i + y_i$

- **普通に得意**

  $z_i = (x_{i-1} + 2x_i + x_{i+1}) / 4$

- **微妙に得意**

  $z = x_1 + x_2 + \cdots + x_n$

- **何とかなる**

  $z_i = x_{f(i)}$ s.t. $z_1 \le z_2 \le \cdots \le z_n$

- **全然ダメ**

  $z_1 = f(x_1, 0),\ z_i = f(x_i, z_{i-1})$

大量同種演算は何でも得意か? (2)

- **超得意**

  $z_i = x_i + y_i$

- **普通に得意**

  $z_i = (x_{i-1} + 2x_i + x_{i+1}) / 4$

- **微妙に得意**

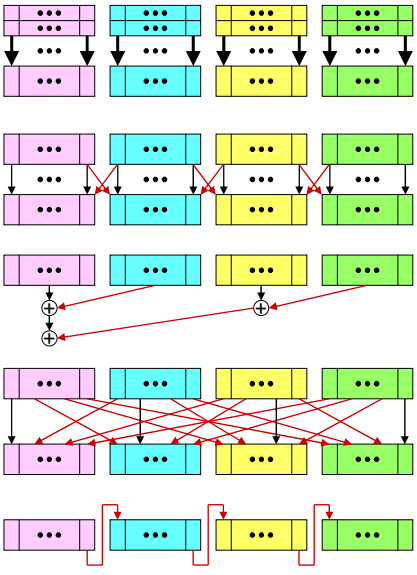
  $z = x_1 + x_2 + \cdots + x_n$

- **何とかなる**

  $z_i = x_{f(i)}$ s.t. $z_1 \le z_2 \le \cdots \le z_n$

- **全然ダメ**

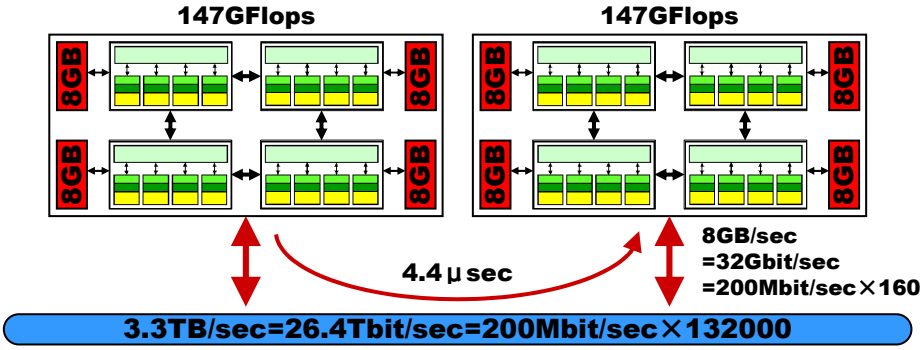
  $z_1 = f(x_1, 0),\ z_i = f(x_i, z_{i-1})$



スーパーコンピュータの原理

大量同種演算は何でも得意か? (3)

- 京大スパコンの通信速度



  - 1個の数値(8B)の通信時間＝4.4μsec
    ＝40480回の演算時間
  - 10億個の数値(8GB)の通信時間＝1秒
    ＝1470億回の演算時間

まとめ＆課題

- スーパーコンピュータは …
  - 大量の同じ(ような)演算(や処理)が得意
  - ただし演算どうしの依存性が少ないことが必要

➔そんな都合のよい問題はあるのか？

- そこでレポート課題
  (できればスパコンに適する大規模な)並列計算に
  より高い性能が期待できる実際的な問題を一つ挙げ、
  なぜその問題が並列計算に適するのかを説明せよ。