

2012年度
計算機システム演習 第13回

計算機システムTA 福田圭祐 (松岡研究室)

Q：スパコンってなあに？



松岡聡 教授

チエツク!



シーモア・クレイは失敗を恐れない人で、成功するまでとことん頑張っちゃうエライ人だよ。シーモア・クレイ賞ってのができるくらいだから、スパコンのダディでクールに違いなし。

みんなとつながる
- Friend Network -

©2011 Sony Computer Entertainment Inc.

日程確認

▶ PC組み立て演習

- ▶ 日時：7/20(金) 15:05~ (2時間程度)
- ▶ 場所：講義室に集合 → その後西7演習室
 - ▶ (福田は出張中につきTAは別の人です)

▶ 試験

- ▶ 日時：8/6(月) 3,4限
 - ▶ 場所：W833 (授業と異なるので注意)
-



1. プロセッサの性能：条件分岐

- ▶ 次のようなコードを考える（C言語を用いる）

double 型配列 `array` に、 $(0, 1)$ の一様分布に従う乱数列を代入しておく。

```
void initArray(void) {  
    int i;  
    for (i = 0; i < N; i++) {  
        array[i] = rand() * 1.0 / RAND_MAX;  
    }  
}
```



条件分岐する関数

- ▶ 次のような関数 `funcA`, `funcB` を考える

```
int funcA(double x) {  
    if (x < 0.5)          return 1;  
    else if (x < 0.75)   return 2;  
    else                  return 3;  
}
```

```
int funcB(double x) {  
    if (x >= 0.75)       return 3;  
    else if (x >= 0.5)   return 2;  
    else                 return 1;  
}
```



分岐命令の回数が多い関数

- ▶ さらに、funcC を考える

```
int funcC(double x) {
    if (x > 100)          return 1;
    else if (x < -100)   return 2;
    else if (x > 100)    return 3;
    else if (x < -99)    return 4;
    else                  return 5;
}
```



3つの関数の性能差

- ▶ これらの関数の実行時間について考察してみよう
 - ▶ 実行にかかる時間は、実行する命令数で決まる？
 - ▶ 命令は、**比較**と **return** だけ
 - ▶ return は同じだから、比較命令の回数が支配的？

- ▶ 比較命令が実行される回数の期待値は？



実行結果

- ▶ 実際の実行結果はどのようなであろうか？
 - ▶ 予測を裏付けるか？
- ▶ コンパイル方法(Mac・Linuxの場合)
 - ▶ `gcc -O0 branch.c -o branch`
 - ▶ コンパイラの最適化の影響を排除
 - ▶ `-O0` : 最適化(Optimize) レベルゼロ(最適化を行わない)
- ▶ 実行
 - ▶ `./branch`



プロセッサの性能（２）：メモリ階層

- ▶ FLOPS : FLoating OPerations per Second (諸説あり)
 - ▶ 主に加減算と乗算を対象とする

- ▶ 行列積 $C = AB$ を考えてみよう
 - ▶ 簡単のため A, B, C は全てN次正方行列とする

- ▶ この行列積演算は、何FLOPの演算か？



行列積のFLOPS

- ▶ $N=2$ とすると...

$$C = \begin{pmatrix} a_{11} \cdot b_{11} + a_{12} \cdot b_{21} & a_{11} \cdot b_{12} + a_{12} \cdot b_{22} \\ a_{21} \cdot b_{11} + a_{22} \cdot b_{21} & a_{21} \cdot b_{12} + a_{22} \cdot b_{22} \end{pmatrix}$$

- ▶ C の1要素を計算するために必要な計算回数
 $= 2N - 1$
 - ▶ C の全要素の計算に必要なFLOPは
 $= N^2 (2N - 1)$
-



行列積のFLOPS (cont.)

- ▶ 処理にかかった時間を T [sec] とすれば、

$$F_{matmul}(n) = \frac{2n^2(n-1)}{T}$$

- ▶ 実装によっては、浮動小数点の回数が若干多い場合もあるが、それは実装上のオーバーヘッドとみなす



行列積の計算コード

- ▶ 行列積は、3重ループで計算できる
- ▶ ここでは、行列のデータ構造として、1次元配列を用いる

```
for (i = 0; i < N; i++) {  
    for (j = 0; j < N; j++) {  
        for (k = 0; k < N; k++) {  
            C[i*N+j] += A[i*N+k] * B[k*N+j];  
        }  
    }  
}
```



行列式の計算順序

- ▶ このループ形式を **IJK** ループと呼ぶことにする
- ▶ 計算結果は、ループの順番には依存しない
 - ▶ **KJI, JIK, ...**

```
for (i = 0; i < N; i++) {  
    for (j = 0; j < N; j++) {  
        for (k = 0; k < N; k++) {  
            C[i*N+j] += A[i*N+k] * B[k*N+j];  
        }  
    }  
}
```



計算順序の違いによる速度差

- ▶ ループ順序の違いによって計算速度に差はでるだろうか？
- ▶ どれくらいの差があるか？



既成の行列積演算ルーチン

- ▶ 現実には、行列積のような基本的なルーチンを手で書くことは多くない
 - ▶ 商用もしくはフリーの数学演算ライブラリを使う
- ▶ TSUBAME2.0 では・・・
 - ▶ Intel社製のMath Kernel Library (MKL) が利用可能
 - ▶ GPUを用いて行列積を計算することも可能



課題

最終締め切りに注意！

課題1：分岐

- 1) `branch.c` をコンパイルして実行し、結果をまとめよ。コンパイル方法に注意すること。

このような `funcA`, `funcB`, `funcC` の挙動は、「分岐予測」という概念で(ある程度)説明できる。

- 2) 「分岐予測」についてまとめよ。どのような機能で、どのようなメリットがあるのか、講義内容を踏まえて解説せよ。(参考文献がある場合は明示すること)
- 3) `funcA`, `funcB`, `funcC` の挙動(速度差)について、上記を踏まえて説明せよ



課題2： 行列積

- 1) 行列積プログラム `matmul.c` を変更し、ループ順序6通りについて行列積を計算し、速度をグラフにせよ。
(実験を行った環境を明記すること)
- 2) 6通りの計算方法について、「空間的局所性」と「時間的局所性」について、表形式でまとめよ。
(乗算を単位として、演算の順序を考えてみよ)
- 3) 上記の結果に基づき、6通りの行列積計算法の速度差が生じる理由を考察せよ。

※ (2), (3) については、厳密に一致するわけではないので、大まかな傾向がわかればよい

課題2：行列積（補足&ヒント）

	IJK	IKJ	JKI	JIK	KIJ	KJI
Aの空間的局所性	○					
Aの時間的局所性	×					
Bの空間的局所性	×					
Bの時間的局所性	×					
Cの空間的局所性	○					
Cの時間的局所性	○					
行列積速度 [Gflops]						



課題提出

- ▶ 〆切: 8/6(Mon) 23:59 (最終締め切り)
 - ▶ 未提出の課題も上記の締め切りまでに提出すること
- ▶ 提出物: 以下のファイルを**圧縮したもの**
 - ▶ ドキュメント(pdf,plain txt,wordなんでも可)
 - ▶ 演習の授業に対する感想等
- ▶ 提出方法: Webから提出
- ▶ 注意: TSUBAMEは、停止期間が近づくと混みます

