

高性能コンピューティング特論 講義メモ(2)

「最も重要なのはバンド幅」

高橋大介

daisuke@cs.tsukuba.ac.jp

筑波大学大学院システム情報工学研究科
計算科学研究センター

2010/12/8

高性能コンピューティング特論

1

講義内容

- プロセッサへのデータ供給能力の指標
- キャッシュメモリ
- 記憶階層
- バンク・メモリ
- Byte/Flopの概念

2010/12/8

高性能コンピューティング特論

2

プロセッサへのデータ供給能力の指標

- 科学技術計算を行う計算機システムにおいて最も重要なのは「プロセッサへのデータ供給能力」である。
- プロセッサの演算器にデータが供給されないと、計算を行うことができない。
- プロセッサの演算性能は、データ供給能力に大きく左右される。
- データ供給能力の指標として、「バンド幅」を用いる。

2010/12/8

高性能コンピューティング特論

3

メモリ性能

- プロセッサにおいて、連続的に結果を得るためには、連続的に途切れなくデータを演算器に流し込み、かつ、結果を連続的に取り出すことが必要。
- このためには、プロセッサの性能に見合っただけの演算器に対するデータの転送能力(スループット)が必要となる。

```
for (i = 0; i < n; i++)  
    z[i] = x[i] + y[i];
```
- $x[i]$, $y[i]$, $z[i]$ がすべてメモリ上のベクトルであるとする
と、1クロックに、 $x[i]$, $y[i]$ の2つのデータのメモリからの読み出しと、結果 $z[i]$ の書き込みの能力が必要となる。
- つまり、3アクセス/クロックのデータ転送能力が必要となる。

2010/12/8

高性能コンピューティング特論

4

キャッシュメモリ(1/2)

- 初期の計算機は演算能力が低かったので、演算そのものに要する時間よりも主記憶からデータをもってくる時間が短かった。
- 演算能力が向上するに従って、主記憶のアクセス速度が無視できなくなってきた。
- そこでアーキテクチャからは透過な形で、主記憶のアクセスの遅さを緩和する手段としてキャッシュが考案された。
- 1968年にIBMのS/360モデル85で最初に用いられた。

2010/12/8

高性能コンピューティング特論

5

キャッシュメモリ(2/2)

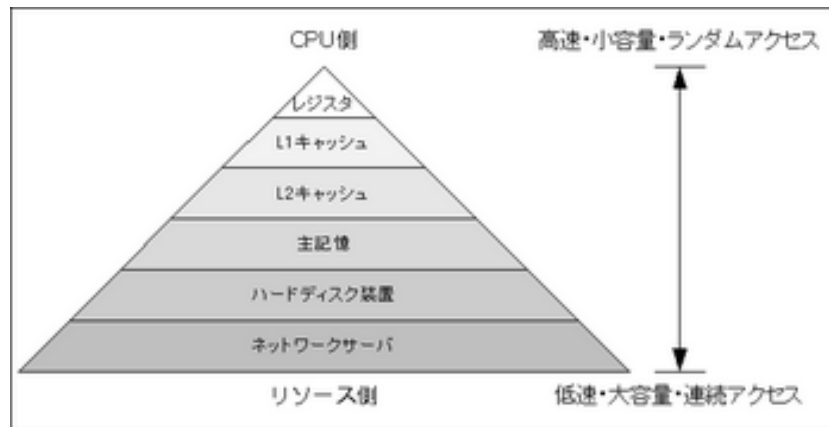
- プログラムから初めてアクセスされたデータ(実際にはキャッシュラインと呼ばれる)は、主記憶からキャッシュを経由して演算器に送られた後、キャッシュに残される。
- 続けてそのデータや、そのキャッシュラインに含まれるデータが使用されると、データは主記憶からではなく、キャッシュからアクセスされる。
- このアクセス時間は最初のアクセスよりもはるかに短い。
- キャッシュの管理は基本的にはハードウェアによって行われ、そのハードウェア上で実行されるプログラムからはキャッシュの存在は見えない。

2010/12/8

高性能コンピューティング特論

6

記憶階層(1/4)



出典: Wikipedia

記憶階層(2/4)

- データを保持する記憶装置のコストバランス
 - 「小容量 × 高速 ÷ 大容量 × 低速」が成り立つ
- 「小容量 × 高速」記憶装置
 - レジスタ
- 「大容量 × 低速」記憶装置
 - ハードディスクや磁気テープ
- 「大容量 × 高速」はコストパフォーマンスが悪く実現困難

2010/12/8

高性能コンピューティング特論

7

2010/12/8

高性能コンピューティング特論

8

記憶階層 (3/4)

- 記憶階層は記憶域に対するアクセスパターンの局所性 (locality) を前提に設計されている。
- 局所性には
 - 時間的局所性
 - ある一定のアドレスに対するアクセスは、比較的近い時間内に再発するという性質
 - 空間的局所性
 - ある一定時間内にアクセスされるデータは、比較的近いアドレスに分布するという性質

記憶階層 (4/4)

- これらの傾向は、事務計算などの非数値計算には当てはまることが多いが、数値計算プログラムでは一般的ではない。
- 特に大規模な科学技術計算においては、データ参照に時間的局所性がないことが多い。
- これが、科学技術計算でベクトル型スーパーコンピュータが有利であった大きな理由。

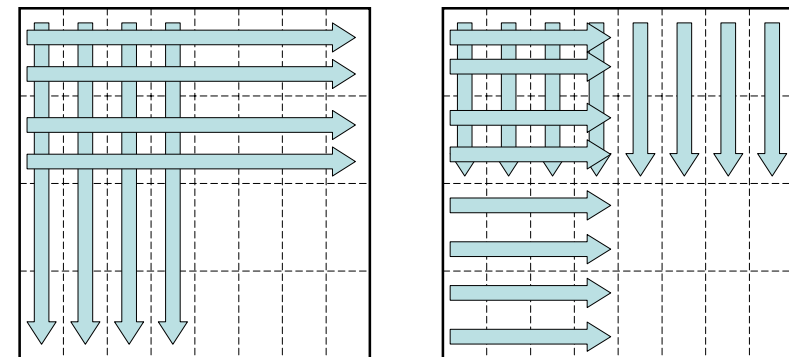
ブロック化 (1/2)

- メモリ参照を最適化するための有効な方法。
- キャッシュミスをできるだけ減らす。

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

double A[N][N], B[N][N], C;
for (i = 0; i < N; i += 4) {
  for (j = 0; j < N; j += 4) {
    for (ii = i; ii < i + 4; ii++) {
      for (jj = j; jj < j + 4; jj++) {
        A[ii][jj] += B[jj][i] * C;
      }
    }
  }
}
```

ブロック化 (2/2)



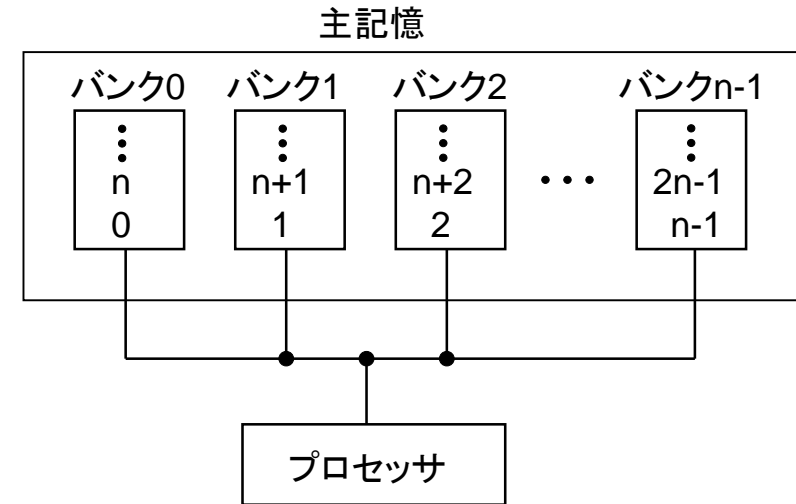
ブロック化しない場合の
メモリアクセスパターン

ブロック化した場合の
メモリアクセスパターン

バンク・メモリ

- 同一メモリにアクセスできる最小のアクセス間隔をメモリのサイクル時間(T_c)という。
- メモリが単一の装置から構成されていると、ベクトルの加算($z[i] = x[i] + y[i]$)の例では、3アクセス/クロックであるから、少なくとも $T_c = \text{クロック} / 3$ である必要がある。
- ところが主記憶に使用するメモリ素子(DRAM)は、大容量ではあるが、演算速度に比べ遅いのが普通。
- したがって、単一のメモリ装置では、高いデータ転送能力を出すのが不可能なので、バンクと呼ぶ複数個の独立して動作するモジュールで構成する。

メモリアンターリーブ方式



パディング

- バンク・コンフリクトを防ぐためには、以下のようなパディングが有効。
 - 特にサイズが2のべきとなる配列の場合
- 二次元配列の定義サイズを少し変えてみる。
- コンパイルオプションを指定すると行ってくれるものもある。

```
double A[N][N], B[N][N];
for (k = 0; k < N; k++) {
  for (j = 0; j < N; j++) {
    A[j][k] = B[k][j];
  }
}
```



```
double A[N][N+1], B[N][N+1];
for (k = 0; k < N; k++) {
  for (j = 0; j < N; j++) {
    A[j][k] = B[k][j];
  }
}
```

パディングを行う前

パディングを行った後

バンク・コンフリクト

- 1次元の連続アクセスでは、インターリーブが有効に働く。
- ところが、行列の列方向/対角方向などのアクセスでは、等間隔アクセスとなる。
- n-wayインターリーブ方式のメモリにおいて、アクセス間隔がnであると、常に同一メモリモジュールをアクセスするので、インターリーブの効果なくなる。
- これを、バンク・コンフリクトと呼ぶ。

Byte/Flopの概念

- 1回の浮動小数点演算を行う際に必要なメモリアクセ
ス量をByte/Flopで定義することができる。

```
void daxpy(int n, double a, double *x, double *y)
{
    int i;
    for (i = 0; i < n; i++)
        y[i] += a * x[i];
}
```

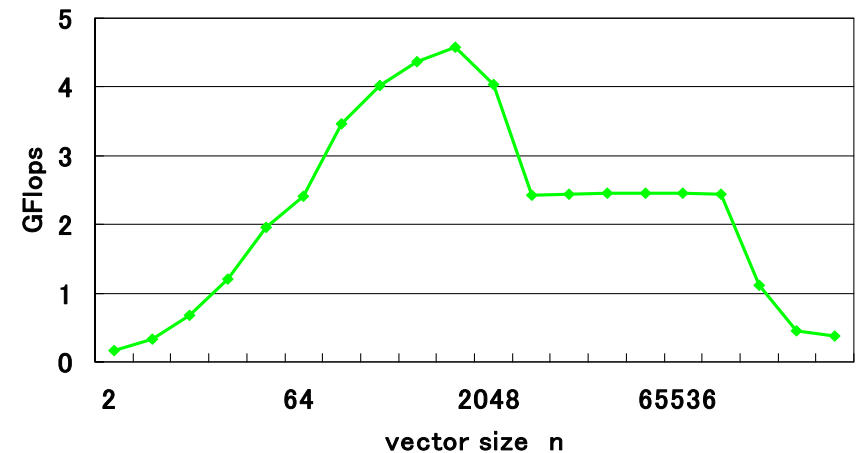
- daxpyでは、1回のiterationにつき、2回の倍精度浮動
小数点演算に対して3回の倍精度実数データ(合計
24Byte)のload/storeが必要。
 - この場合、24Byte/2Flop = 12Byte/Flopとなる。
- Byte/Flop値は、小さいほど良い。

2010/12/8

高性能コンピューティング特論

17

BLASの性能 (Woodcrest 2.4GHz 4MB L2 cache, Intel MKL 9.1)



2010/12/8

高性能コンピューティング特論

18

PCとベクトル型スパコンの メモリバンド幅

- Intel Quad-Core Xeon X5365 (Clovertown 3GHz,
FSB 1333MHz, 2ソケット)では、1ノード当たり
 - 理論ピーク性能は
 $12\text{GFlops} \times 4\text{コア} \times 2\text{ソケット} = 96\text{GFlops}$
 - メモリバンド幅は最大21GB/s
 - Byte/Flop値は $21/96 \doteq 0.219$
- NEC SX-9 (16CPU/ノード)では、1ノード当たり
 - 最大ベクトル性能は
 $102.4\text{GFlops} \times 16\text{CPU} = 1638.4\text{GFlops}$
 - メモリバンド幅は最大4TB/s
 - Byte/Flop値は $4096/1638.4 = 2.5$

2010/12/8

高性能コンピューティング特論

19

daxpyにおける理論性能の比較

- Intel Quad-Core Xeon X5365 (Clovertown
3GHz, FSB 1333MHz, 2CPU)
 - ワーキングセットがキャッシュの容量を超えた場合、
メモリバンド幅(21GB/s)が律速となるので、
 $(21\text{GB/s}) / (12\text{Byte/Flop}) = 1.75\text{GFlops}$ が限界。
 - 理論ピーク性能の約1.8%!
- NEC SX-9 (16CPU/ノード)
 - メモリバンド幅(4TB/s)が律速となるので、
 $(4\text{TB/s}) / (12\text{Byte/Flop}) \doteq 341.3\text{GFlops}$ が限界。
 - 最大ベクトル性能の約20.8%

2010/12/8

高性能コンピューティング特論

20