宇宙物理研究室　Astrophysics Group, OIT　http://www.is.oit.ac.jp/~shinkai/

卒業研究中間報告

N体問題を用いたCPUとGPUの性能比較

ー行列の積 演算を用いた比較ー

富久　友樹

研究目的

- グラフィック用の演算目的に開発されたGPUは高い並列計算能力を持っているとされる。この能力を利用し、科学的数値計算において、従来のCPUを用いるシミュレーションに対し、どの程度CPUとGPUの性能が違うかを調べることを目的とする。
- また、CPUの性能をGPUの性能が上回る境界を調べることも目的とする。

基本用語①
GPUとは

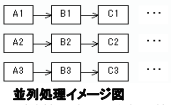
- 3Dグラフィックスの表示に必要な計算処理を行うプロセッサ(Graphics Processing Unit)
- 高い並列計算能力を持っており、この能力をグラフィック描画だけでなく汎用の数値計算にも利用することに近年注目が集まっている。
- 本研究で使用したGPUは"GeForce GTX 285(2009年製)"で倍精度にも対応している。コア数は240個、メモリは1GBである。

基本用語②
CUDAとは

- NVIDIA社が提供するGPU向けのC言語の統合開発環境であり、コンパイラやライブラリなどから構成されている。
- CUDAでGPU向けのプログラムを開発するにはNVIDIA社のビデオカードと対応しているOS(例えばWindows/Linux/Mac)が必要である。
- http://developer.nvidia.com/object/cuda_3_2_toolkit_rc.html より開発ツールキットがダウンロード可。開発環境のOSのツールキットが必要である。

基本用語③
並列処理

- 複数のマイクロプロセッサなどに処理を分散して、同時に演算を行うことである。
- 並列処理をすれば必ずしも演算速度が早くなるというわけではない。データ通信時間やプロセッサ同期にも時間を要するからである。
- 並列処理は演算が同一プロセッサ内のメモリで処理できる場合に力を発揮する。



並列処理イメージ図

出典http://www.jpo.go.jp/shiryou/s_sonota/map/denki17/4/4-2-3.htm

プログラミングの特徴①
CUDAの概念

- GPUを用いたプログラミングでは演算処理単位を考えたコードを書く必要がある。
- CUDAの概念上、一つのデータを処理する単位を「スレッド」という。
- 「ブロック」(block)とは同じサイズのスレッドをまとめたものを言う。最大3次元のスレッドの配列をブロックとすることができる。
- 「グリッド」(grid)とは同じサイズのブロックをまとめたものを言う。最大2次元のブロックの配列を含むことができる。

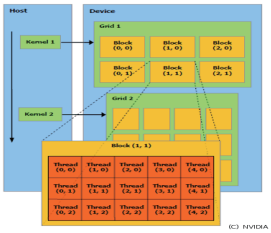
プログラミングの特徴②
実際のプログラムの例

```
dim3 grid(WIDTH/BLOCK, WIDTH/BLOCK, 1);   //ブロックを生成
dim3 threads(BLOCK, BLOCK, 1);            //スレッドを生成
Kernel1<<< grid, threads >>>(d_a, d_b, d_c);   //Kernel1関数を呼び出し

__global__ void Kernel1(float *A, float *B, float *C)
{
   //blockIdxはブロック番号,blockDimはスレッド数,threadIdはスレッド番号を示す
   int x=blockIdx.x*blockDim.x + threadIdx.x;
   int y=blockIdx.y*blockDim.y + threadIdx.y;
   float tmp=0.0;
   for(int k=0; k<WIDTH; k++) {
   int row=k+y*WIDTH;
   int col=x+k*WIDTH;
   tmp+=A[row]*B[col];
   }

   C[x+y*WIDTH]=tmp;
}
```

プログラミングの特徴③
CUDAのプログラミングモデル

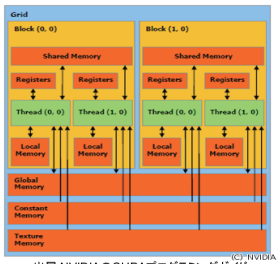


出典:NVIDIAのCUDAプログラミングガイド

プログラミングの特徴④
CUDAのメモリモデル

- グローバルメモリはCPU側からのデータ入力、GPU側からの計算結果の出力に使われ、演算に用いることもできる。
- シェアードメモリはブロック間で共有でき、グローバルメモリに比べて非常に高速な演算が可能である。
- GPU上で演算させる場合、シェアードメモリにデータを移してから演算させる方が高速に演算を行うことができる。

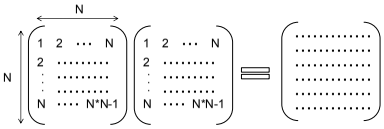
プログラミングの特徴⑤
CUDAのメモリモデル図



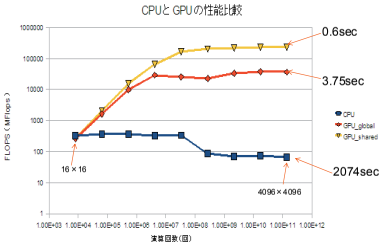
出展:NVIDIAのCUDAプログラミングガイド

性能比較テスト①
行列計算

- N＊Nの行列同士の積を計算した。(Nは2のべき乗)
- 行列計算を用いた理由は並列計算に向いているため、演算速度の違いを比較するのに向いていると考えたからである。



性能比較テスト②
行列計算による性能比較図



性能比較テスト③
性能比較図の解説

- GPU_globalはグローバルメモリを使用して演算したことを示し、GPU_sharedはシェアードメモリを使用して演算したことを示している。
- 演算回数が7936回の時はCPUの方が演算速度が速いのでGPUの演算速度が必ずCPUより速くなるとは限らない。
- GPUは演算回数が約3348万回を超えたあたりから演算速度の伸びが悪くなる。このことから、GPUの実力を限界まで使えたとわかった。

現時点までにわかったこと

- GPUの演算速度がCPUの演算速度を必ず上回るわけではない。上回るにはある程度演算回数が多くなくてはいけない。今回試した行列計算問題では32×32以上の行列計算で上回る。
- GPUをうまく使えれば、科学的数値計算において、大幅な演算時間の短縮が可能にと考えられる。ただし、プログラミング上の工夫やメモリ使用のチューニングが必要である。

今後の課題

- N体問題を用いてCPUとGPUの性能比較を行ない、それをグラフ化する。
- また、より速くするためにはどのようにプログラムを書けばいいのかを研究する。

大阪工業大学　情報科学部

2010/11/3 北山祭