

2007年度, 2008年度 理研ベンチマークコンテスト報告

2008年 3月, 2009年3月

東京工業大学 学術国際情報センター
青木 尊之

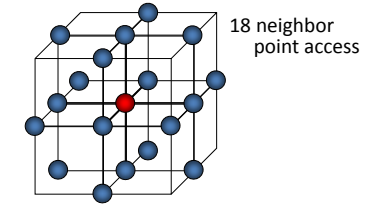
理研(姫野)ベンチマーク問題

Poisson Equation: $\nabla \cdot (\nabla p) = \rho$
(Generalized coordinate)

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} + \alpha \frac{\partial^2 p}{\partial xy} + \beta \frac{\partial^2 p}{\partial xz} + \gamma \frac{\partial^2 p}{\partial yz} = \rho$$

Discretized Form:

$$\begin{aligned} & \frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{\Delta x^2} + \frac{p_{i,j+1,k} - 2p_{i,j,k} + p_{i,j-1,k}}{\Delta y^2} + \frac{p_{i,j,k+1} - 2p_{i,j,k} + p_{i,j,k-1}}{\Delta z^2} \\ & + \alpha \frac{p_{i+1,j+1,k} - p_{i-1,j+1,k} - p_{i+1,j-1,k} + p_{i-1,j-1,k}}{4\Delta x\Delta y} \\ & + \beta \frac{p_{i+1,j,k+1} - p_{i-1,j,k+1} - p_{i+1,j-1,k} + p_{i-1,j-1,k}}{4\Delta x\Delta z} \\ & + \gamma \frac{p_{i,j+1,k+1} - p_{i,j-1,k+1} - p_{i,j+1,k-1} + p_{i,j-1,k-1}}{4\Delta y\Delta z} = \rho_{i,j,k} \end{aligned}$$



姫野ベンチの詳細

Read only 12 arrays : a, b, c, bnd, ...
Read-write 2 arrays : p, wrk2

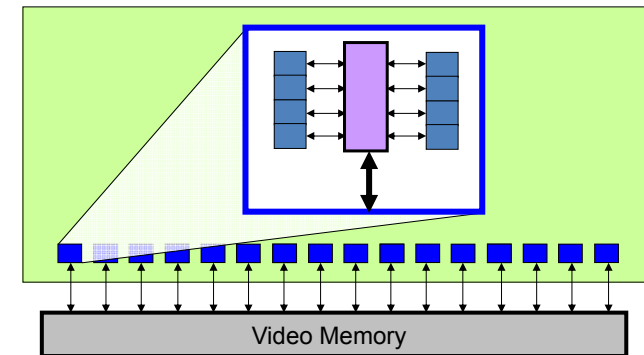
```
#define MIMAX 65
#define MJMAX 65
#define MKMAX 129

static float p[MIMAX][MJMAX][MKMAX];
static float a[4][MIMAX][MJMAX][MKMAX],
             b[3][MIMAX][MJMAX][MKMAX],
             c[3][MIMAX][MJMAX][MKMAX];
static float bnd[MIMAX][MJMAX][MKMAX];
static float wrk1[MIMAX][MJMAX][MKMAX],
             wrk2[MIMAX][MJMAX][MKMAX];

for(i=1; i<imax-1; i++)
  for(j=1; j<jmax-1; j++)
    for(k=1; k<kmax-1; k++){
      s0 = a[0][i][j][k] * p[i+1][j][k]
          + a[1][i][j][k] * p[i][j+1][k]
          + a[2][i][j][k] * p[i][j][k+1]
          + b[0][i][j][k] * ( p[i+1][j+1][k] - p[i+1][j-1][k]
                          - p[i-1][j+1][k] + p[i-1][j-1][k] )
          + b[1][i][j][k] * ( p[i][j+1][k+1] - p[i][j-1][k+1]
                          - p[i][j+1][k-1] + p[i][j-1][k-1] )
          + b[2][i][j][k] * ( p[i+1][j][k+1] - p[i-1][j][k+1]
                          - p[i+1][j][k-1] + p[i-1][j][k-1] )
          + c[0][i][j][k] * p[i-1][j][k]
          + c[1][i][j][k] * p[i][j-1][k]
          + c[2][i][j][k] * p[i][j][k-1]
          + wrk1[i][j][k];

      ss = ( s0 * a[3][i][j][k] - p[i][j][k] ) * bnd[i][j][k];
      wrk2[i][j][k] = p[i][j][k] + omega * ss;
    }
} /* end n loop */
```

GPU アーキテクチャ



- Global memory 256MB~1GB (video memoryとも呼称)
- Multiprocessor 16個(例えば9800GTX(G92))
- Shared memory 16 Kbyte
- Streaming Processor 1つのMultiprocessorに8個ずつ。計128個

GPU 計算の概要



jacobi() 関数の中の計算を全て GPU で処理させる。

initmt() 関数の中で、使用する全ての配列の初期値を GPU のdeviceメモリ(ビデオメモリ)に転送しておく。

Point Jacobi 計算を GPU の 128 個のストリーミング・プロセッサで並列計算させる。nVIDIA の CUDA (Compute unified device architecture) を使い、プログラミングを行った。計算領域を16x16x8格子単位で分割し、CUDAのblockと対応付ける。block内には256スレッドのジョブが並列で走り、計算全体では65536スレッドの並列計算となっている。1 スレッドは8格子点の計算を担当している。

GPUの計算を効率化させるため、チップ内にあるShared メモリという高速なメモリをキャッシュ的に使う。(圧力変数のload回数を減らすため) Shared メモリは上記block内のスレッドでのみ共有アクセスできる。

一時変数 wrk2[][][] と p[][][] のデータ内容の交換は、配列ポインターの交換のみとした。誤差計算は500回の反復計算の最後の1回のみで行う(承認済み)。ss*ss の総和計算はblock内ではShared メモリを使いreductionによる並列計算を行う。

GPUによる計算の最高性能の予測



A[129][65][65] : 2.18 MB × 14 変数

12変数: read only
1 変数: read-write
1 変数: write

1格子点当り = 34 浮動小数点演算

メモリ転送 1 Word 当りの演算数 = $34/14 = 2.4$

```
#define NIMAX      65
#define NIJMAX    65
#define NIKMAX    129

static float p[NIMAX][NIJMAX][NIKMAX];
static float a[4][NIMAX][NIJMAX][NIKMAX];
static float b[3][NIMAX][NIJMAX][NIKMAX];
static float c[3][NIMAX][NIJMAX][NIKMAX];
static float bnd[NIMAX][NIJMAX][NIKMAX];
static float wrk1[NIMAX][NIJMAX][NIKMAX];
static float wrk2[NIMAX][NIJMAX][NIKMAX];
```

GPU のビデオメモリの転送速度が
70 GB/sec (17.5 GWord/sec) の場合:

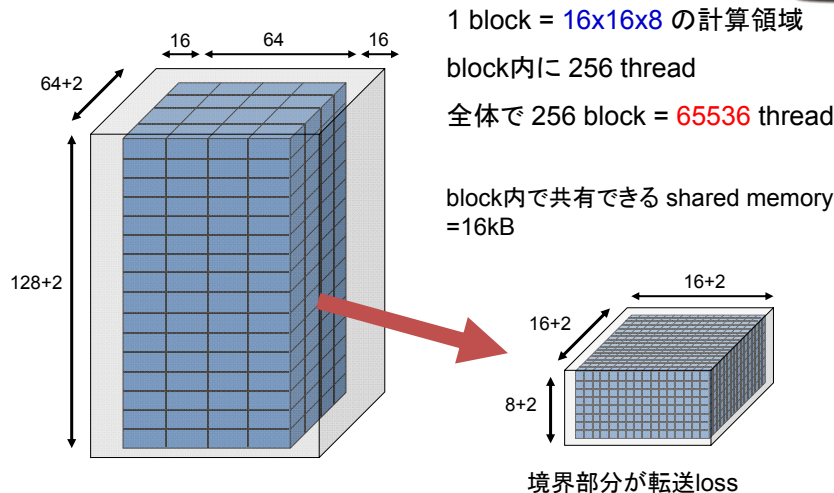
GPU の演算速度が無限大であったとしても、

$$17.5 \times 2.4 = 42 \text{ GFLOPS}$$

圧力の隣接格子点参照を毎回行くと、 $34/14 \rightarrow 34/(14+18) = 1.06$

$$17.5 \times 1.06 = 18.5 \text{ GFLOPS}$$

Shared-memory の利用



Specifications of GPU and Motherboard

		GeForce 8800Ultra(G80) (MSI)	GeForce 8800Ultra(G80) (ELSA)
GPU	Peak Performance [GFlops]*	414.2	384
	# of SP	128	128
	SP Clock(CoreCock)[MHz]	1618(660)	1500(612)
Video Memory	Transfer Rate[GB/s]	110.4	103.68
	Memory Bus width[bit]	384	384
	Data Rate[GHz]	2.3(GDDR3)	2.16(GDDR3)
	Capacity [MB]	768	768

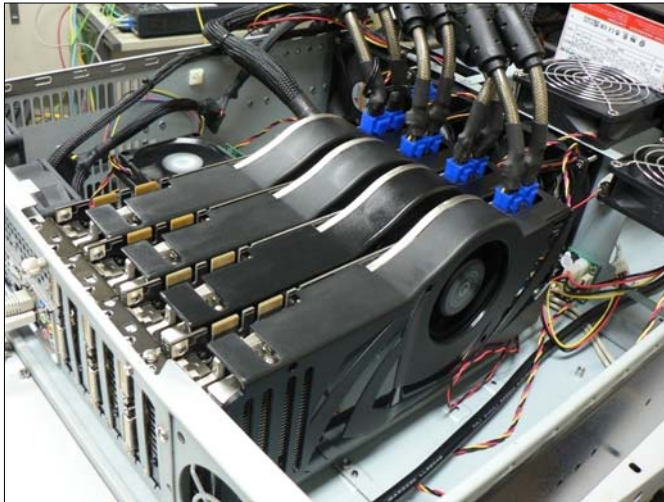
* 2 instruction issue



MSI K9A2 Platinum
AMD 790FX + AMD SB600
4 PCI-Express x16-type slots
•2 slot Support up to PCI-Express 2.0 x 16
•2 slot Support up to PCI-Express 2.0 x 8

When using 4 GPU card, it works as PCI-Express x 8 for each GPU

MSI Phenom Motherboard + 4 GPU GeForce 8800 Ultra x 4



4 GPU の並列計算について



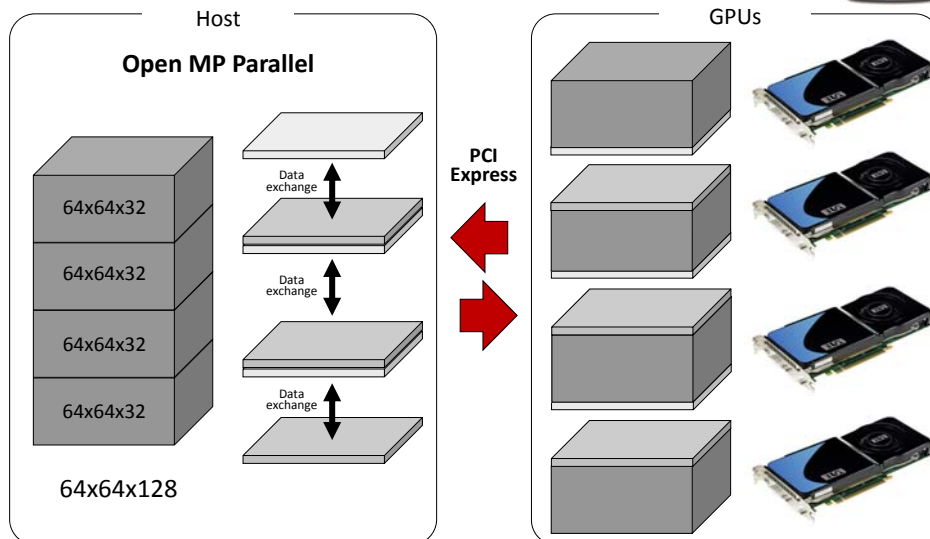
理研ベンチマークテストをさらに高速化するため、PCI Express x16 のスロットが4つ付いているマザーボードのマシンを準備し、GPU 並列計算を行った。

GPU 上のメモリは他の GPU から見えないため、分散メモリでの並列計算と同じ状況になる。z方向に $129-1=128$ 格子を短冊型に4分割し、4つのGPUが正味 $64 \times 64 \times 32$ の計算領域を並列で計算する。

GPU の制御は、CPU 側が jacobi 計算の部分になったら Open MP の4スレッド並列で実行し、それぞれのCPU スレッドが並列に GPU のカーネル関数を実行させる。

領域間の糊しる部分のデータ交換は、CPUとGPU双方に転送バッファを用意し、糊しる領域をGPU → CPUに転送した後、CPU側で領域交換を行い、再びCPU → GPUへデータ転送する。このデータ交換はPCI Express Busを通して毎回反復計算終了後に行われる。1回の転送量は32kBしかなく、GPU→CPU転送のオーバーヘッドが特に大きく、転送速度が 800MB/sec 程度しか出ていないため、並列計算の足を引っ張る結果となっている。

Parallelization using 4 GPU



浮動小数点の計算誤差について



理研ベンチマークテストを CPU で実行したときと、GPU を用いて計算したときの計算結果 gosa の値が4桁目から異なる。

これは、以下の2点の原因による。

- GPU の浮動小数点演算の精度がCPUの演算精度と若干異なる。正しい計算プログラムであることを確認するために、CUDA のエミュレーション・モードで計算を実行し(実際の実行はCPUで行う)、gosa が等しくなることを確認した。
- 単精度変数の総和計算では、総和計算の順序により結果がことなる。CPUで理研ベンチマークテストを行う場合も、 $ss*ss$ の総和計算のループの順序を入れ替えるだけで計算結果はことなる。GPUでの総和計算は $16 \times 16 \times 8$ 格子からなるblock内で先に総和計算するため、元のベンチマークテストの総和計算の順序と異なっている。

計算結果は、元のベンチマークテストの方も総和計算を GPU と同じ順序にし、GPU での計算をエミュレーションで行った結果と一致することを確認した。

RESULT



0.976 GFLOPS (8.431sec) → 51.91 GFLOPS (0.158sec)

× 53.1

• Before

```

mimax = 65 mjmax = 65 mkmax = 129
imax = 64 jmax = 64 kmax = 128
Start rehearsal measurement process.
Measure the performance in 3 times.

MFLOPS: 941.082902 time(s): 0.052496 3.288628e-03

Now, start the actual measurement process.
The loop will be excuted in 500 times
This will take about one minute.
Wait for a while

Loop excuted for 500 times
Gosa : 9.673350e-04
MFLOPS measured : 976.566479      cpu : 8.431426
Score based on Pentium III 600MHz : 11.909347

```

• After

```

[INFO] Number of host available CPU : 4
[INFO] Number of CUDA devices      : 4

4-GPU OpenMP execution . . . . .

mimax = 65 mjmax = 65 mkmax = 129
imax = 64 jmax = 64 kmax = 128
Start rehearsal measurement process.
Measure the performance in 3 times.

MFLOPS: 34717.560084 time(s): 0.001423 3.295089e-03

Now, start the actual measurement process.
The loop will be excuted in 500 times
This will take about one minute.
Wait for a while

Loop excuted for 500 times
Gosa : 9.672065e-04
MFLOPS measured : 51909.594689      cpu : 0.158619
Score based on Pentium III 600MHz : 633.043838

```

平成19年度 RIKEN ベンチマークコンテスト

(平成20年度3月13日)



1 PC 部門 優勝



東京工業大学 学術国際情報センター
青木 尊之, 小川 慧

GeForce 8800 Ultra × 4枚

0.976 GFLOPS (8.431sec) → 51.91 GFLOPS (0.158sec)

53.1倍の加速

Parallel Performance



Sモデル [65x65x129]

1 GPU (no data transfer)	30.6 GFLOPS (0.269sec)
2 GPU (16kB transfer)	42.5 GFLOPS (0.193sec)
4 GPU (32kB transfer)	51.9 GFLOPS (0.158sec)

Reference

Mモデル [129x129x257]

1 GPU (no data transfer)	29.4 GFLOPS (2.328sec)
2 GPU (66kB transfer)	53.7 GFLOPS (1.275sec)
4 GPU (131kB transfer)	83.6 GFLOPS (0.819sec)

Lモデル [257x257x512]

1 GPU (no data transfer)
2 GPU (262kB transfer)
4 GPU (524kB transfer)	93.6 GFLOPS (5.974sec)

その後



情報処理学会 HPC研究会117

富士通研究所 成瀬 彰氏ら

Global Memory へのアクセスを最適化チューニング

GeForce GTX 280 × 1枚

Performance: 41.2 → 70.1 GFLOPS

平成20年度 RIKEN ベンチマークコンテスト

(平成21年度3月12日)



1PC部門 → 実効性能部門

如何に性能を引き出すか

東京工業大学 学術国際情報センター
額田 彰, 青木 尊之

優勝

GeForce GTX 280 × 1枚

Performance: **71.64** GFLOPS

(Texture Memory の利用)

