

# 作业 1

---

18308045 谷正阳

May 4, 2021

## Contents

<b>1</b>	<b>文件组织</b>	<b>2</b>
<b>2</b>	<b>CUDA 实现</b>	<b>2</b>
2.1	一维 VS 二维 . . . . .	2
2.2	线程块大小对性能的影响 . . . . .	3
2.3	每个线程计算的元素数量对性能的影响 . . . . .	4
2.4	以上配置在处理不同大小的矩阵时 . . . . .	4
2.4.1	1000x1000 . . . . .	4
2.4.2	8192x8192 . . . . .	6
<b>3</b>	<b>OpenMp 实现</b>	<b>7</b>

## 1 文件组织

- test\_a: 一维 VS 二维, 计算 EPOCHS 轮, 取平均时间当作效率, 矩阵大小 1024x1024
- test\_b: 线程块大小对性能的影响, 计算 EPOCHS 轮, 取平均时间当作效率, 矩阵大小 1024x1024
- test\_c: 每个线程计算的元素数量对性能的影响, 计算 EPOCHS 轮, 取平均时间当作效率, 矩阵大小 1024x1024
- test\_d: 以上配置在处理不同大小的矩阵时, 性能可能的差异
  - 1000\_1000: 矩阵大小 1000x1000
    - \* test\_a
    - \* test\_b
    - \* test\_c
  - 8192\_8192: 矩阵大小 8192x8192
    - \* test\_a
    - \* test\_b
    - \* test\_c
- hw1.h: CUDA 实现用到的函数, 函数已写注释
- openmp.cpp: OpenMp 实现, 一维数组表示矩阵, 4 线程, 每个线程对数组中相邻数据做运算。计算 EPOCHS 轮, 取平均时间当作效率, 在大小为 1000x1000, 1024x1024, 8192x8192 的矩阵上分别讨论效率

## 2 CUDA 实现

实现细节见 hw1.h 的代码及注释。

### 2.1 一维 VS 二维

```
> .\test_a\test_a.exe
2D array with 2D grid and block:
Correctness: 1
Efficiency: 0.000205009 us

1D array with 2D grid and block:
Correctness: 1
Efficiency: 0.000183559 us

1D array with 1D grid and block:
Correctness: 1
Efficiency: 0.000165959 us
```

一维数组，grid，block 效率最高。原因是二维数组  $A[y][x]$  内存访问次数多于  $A[y * m + x]$ ，效率低；二维 grid，block 使用一维数组，同一个 block 内线程空间不连续，效率低。

## 2.2 线程块大小对性能的影响

```
Block size: 16
Correctness: 1
Efficiency: 0.000496224 us

Block size: 32
Correctness: 1
Efficiency: 0.000294398 us

Block size: 63
Correctness: 1
Efficiency: 0.000186562 us

Block size: 64
Correctness: 1
Efficiency: 0.000173661 us

Block size: 65
Correctness: 1
Efficiency: 0.00020307 us

Block size: 128
Correctness: 1
Efficiency: 0.000195373 us
```

16、32 效果差，64、128 效果接近，block size 是否是 32 整数倍影响不是很大（有时 128、63、65 用时比 64 少）。block size 最好 32 整数倍的原因是，一个 SM 执行一个 block 中的所有线程，而 SM 负责执行和调度 warp 即以 32 个线程为一个单位，因而要最大化利用 warp，block size 需要是 32 的整数倍。

## 2.3 每个线程计算的元素数量对性能的影响

```
1 elements:
Correctness: 1
Efficiency: 0.000179102 us

2 elements:
Correctness: 1
Efficiency: 0.000183869 us

3 elements:
Correctness: 1
Efficiency: 0.000274386 us

4 elements:
Correctness: 1
Efficiency: 0.000285788 us
```

元素数量越多，性能越差。因为处理多个元素时是串行的，而且运用了循环结构，GPU 不太适用于复杂控制流。

## 2.4 以上配置在处理不同大小的矩阵时

### 2.4.1 1000x1000

```
> .\test_d\1000_1000\test_a\test_a.exe
2D array with 2D grid and block:
Correctness: 1
Efficiency: 0.000203698 us

1D array with 2D grid and block:
Correctness: 1
Efficiency: 0.000191802 us

1D array with 1D grid and block:
Correctness: 1
Efficiency: 0.000175984 us
```

```
> .\test_d\1000_1000\test_b\test_b.exe
Block size: 16
Correctness: 1
Efficiency: 0.000472141 us

Block size: 32
Correctness: 1
Efficiency: 0.000288033 us

Block size: 63
Correctness: 1
Efficiency: 0.000181777 us

Block size: 64
Correctness: 1
Efficiency: 0.000183408 us

Block size: 65
Correctness: 1
Efficiency: 0.000193309 us

Block size: 128
Correctness: 1
Efficiency: 0.000189215 us
```

```
> .\test_d\1000_1000\test_c\test_c.exe
1 elements:
Correctness: 1
Efficiency: 0.00017097 us

2 elements:
Correctness: 1
Efficiency: 0.00017723 us

3 elements:
Correctness: 1
Efficiency: 0.000279466 us

4 elements:
Correctness: 1
Efficiency: 0.000293152 us
```

矩阵变小时，上述结论基本没有变化。

## 2.4.2 8192x8192

```
> .\test_d\8192_8192\test_a\test_a.exe
2D array with 2D grid and block:
Correctness: 1
Efficiency: 0.00993368 us

1D array with 2D grid and block:
Correctness: 1
Efficiency: 0.00884773 us

1D array with 1D grid and block:
Correctness: 1
Efficiency: 0.00816648 us
```

```
> .\test_d\8192_8192\test_b\test_b.exe
Block size: 16
Correctness: 1
Efficiency: 0.0227857 us

Block size: 32
Correctness: 1
Efficiency: 0.0118128 us

Block size: 63
Correctness: 1
Efficiency: 0.00822816 us

Block size: 64
Correctness: 1
Efficiency: 0.00816871 us

Block size: 65
Correctness: 1
Efficiency: 0.00840791 us

Block size: 128
Correctness: 1
Efficiency: 0.00817399 us
```

```
> .\test_d\8192_8192\test_c\test_c.exe
1 elements:
Correctness: 1
Efficiency: 0.00824548 us

2 elements:
Correctness: 1
Efficiency: 0.0082873 us

3 elements:
Correctness: 1
Efficiency: 0.0099824 us

4 elements:
Correctness: 1
Efficiency: 0.0105736 us
```

矩阵变大时，上述结论基本没有变化。

### 3 OpenMp 实现

```
> .\openmp.exe  
Shape: 1000 x 1000  
Efficiency: 0.00367045 us  
  
Shape: 1024 x 1024  
Efficiency: 0.003952 us  
  
Shape: 8192 x 8192  
Efficiency: 0.24844 us
```

都比 CUDA 实现效率低，对于更大的矩阵效率更低，原因是线程数远少于 CUDA 实现，矩阵越大并行程度越低。