實驗十 Memory Hierarchy of CUDA Programming

9617145 資工 4C 許晏峻
9617167 資工 4C 蔡孟儒

1. 實驗目的

了解 CUDA 的 memory hierarchy，global memory 和 local shared memory 的不同，還有速度的差異。

2. 步驟過程

2.1. 基本題

- No shared memory

將原本程式(org)的 body_track function 中的 i for 迴圈和 j for 迴圈展開成 blocksPerGrid*threadsPerBlock 個，因此會變成：

```
int i = (blockIdx.x*threadsPerBlock + threadIdx.x) / frame_width;
int j = (blockIdx.x*threadsPerBlock + threadIdx.x) % frame_width;
if(i >= (frame_height-body_height+1) || j >= (frame_width-body_width+1)){
    return;
}
```

而其後不需做其他更動。

- With shared memory

將 no shared memory 版本程式的 body 改成 shared memory 來存取，因此先宣告一個 __shared__變數取代 body：

```
__shared__ int sub_body[1024];
```

接者讓每個 thread 進來 body_track 後分工將 shared body 填滿(初始化)，並且做一個 synchronization：

```
int part_size = body_height*body_width/threadsPerBlock;
for(int m=threadIdx.x*part_size; m < (threadIdx.x+1)*part_size; ++m){
    sub_body[m]=body[m];
}
__syncthreads();
```

其他部分則和 no shared memory 版本程式相同。

2.2. 進階題

2.3. 特別加分題

3. 數據結果

3.1. 基本題

實驗環境：

- OS：Ubuntu 10.04.01 64bits
- CPU：Intel(R) Core(TM)2 Quad CPU Q8200 @ 2.33GHz
- Mem：2 GB
- Graphic card：NVIDIA GeForce 9600GT

	org	no shared memory	with shared memory
execution time	11.08 seconds	2.32 seconds	0.08 seconds

4. 結論心得

因為這個禮拜參加畢業旅行的緣故，沒有上到課程，因此很多概念比較抽象，所以花了滿多時間再做嘗試。

由基本題中，可以了解到 shared memory 和沒有 shared memory 的差異，的確快了許多；因為在 no shared memory 版本的程式中，每個 thread 都必須自行存取整個 body 一次，所以總共存取了 blocksPerGrid*threadsPerBlock 次；在 shared memory 版本的程式中，先存取整個 body 一次將 body 放到 shared memory，而往後的 blocksPerGrid*threadsPerBlock 次存取皆是存取 shared memory，因此效能有明顯得提升。