

HW4 report

F74081129 吳信葆

What have you done

Busy-waiting and a Mutex

在完成平行運算之前，都會有一個 barrier，而最後一個進 barrier 的 thread，要 `swap(BMPSaveData, BMPData);` 交換指標。

state 當前平滑運算次數是奇數還是偶數，為的就是有 thread 算太快，所以把它擋住。

Cache

由於多個 thread 共用 address space，所以每個 thread 存取位置越近越好。每個 thread 只算列數之餘數是自己的記憶體，這樣就確保在一定時間內我算的地方是相鄰幾行。

Analysis on your result

測試環境是我的筆電 wsl，在老師的主機上也可以。

```
基本速度:    2.40 GHz
插槽:        1
核心數目:    4
邏輯處理器:  8
模擬:        已啟用
L1 快取:     256 KB
L2 快取:     1.0 MB
L3 快取:     8.0 MB
```

我的 CPU 資訊。

編譯:

```
g++ h4_problem1.cpp -o h4_problem1.out -lpthread
```

執行:

Serial

```
paul@LAPTOP-4VQ86L63:/mnt/c/Users/Paul/OneDrive/桌面/平行/hw4$ time ./Smooth.out
Read file successfully!!
Save file successfully!!

real    1m13.526s
user    1m13.475s
sys     0m0.000s
```

2 threads

```
paul@LAPTOP-4VQ86L63:/mnt/c/Users/Paul/OneDrive/桌面/平行/hw4$ ./h4_problem1.out 2
Read file successfully!!
The execution time = 41.334
Save file successfully!!
```

4 threads

```
paul@LAPTOP-4VQ86L63:/mnt/c/Users/Paul/OneDrive/桌面/平行/hw4$ ./h4_problem1.out 4
Read file successfully!!
The execution time = 30.8714
Save file successfully!!
```

8 threads

```
paul@LAPTOP-4VQ86L63:/mnt/c/Users/Paul/OneDrive/桌面/平行/hw4$ ./h4_problem1.out 8
Read file successfully!!
The execution time = 21.667
Save file successfully!!
```

16 threads

```
paul@LAPTOP-4VQ86L63:/mnt/c/Users/Paul/OneDrive/桌面/平行/hw4$ ./h4_problem1.out 16
Read file successfully!!
The execution time = 31.9939
Save file successfully!!
```

如果沒有指定 thread 個數則是 8 thread.

執行時間:

核心數	Serial	2	4	8	16
Hw2(MPI)	73.526	41.883	29.8583	20.0588	31.9939
Hw4(pthread)	73.526	41.334	30.8714	21.667	69.2191

本想做 MPI、pthread 的 efficiency 比較，但發現執行時間差不多所以算了，但可以看到如果超出電腦核心數，MPI 效能降地較為明顯，可能就是要模擬 processor 溝通，而非單純處理 shared data。

Efficient(S/p) of Hw4:

核心數	Serial	2	4	8	16
S/p	1	0.8894	0.5954	0.42418	0.06639

雖然多 thread 的確能提升執行速度，可以看到 efficient 越來越低，可能原因除了 create thread 的 latency 外，還有不斷切換 thread 的 latency，因為同一 thread 不一定會由同一 processor 執行，也不保證每一 thread 每時每刻都在 running。

h4_proble1.cpp 在 ~/hw4/ 資料夾裡。

Any difficulties?

Barrier 要放的位置以及運作方式需要思考一下。