標題：HW3-report
姓名：葉怡君
學號：110062529

# Implementation

## Hw3-1

```
for(int k = 0;k < vertices;k++){
    #pragma omp parallel for num_threads(ncpus)
        for(int i = 0; i < vertices; i++){
            for(int j = 0; j < vertices; j++){
                if(board[i][j] > (board[i][k] + board[k][j]))
                    board[i][j] = board[i][k] + board[k][j];
            }
        }
}
```

我選用了 FLOYD-WARSHALL ALGORITHM 去實作，只將 i 的那層 loop parallize。

## Hw3-2

一開始先 cudaMalloc 一個 d_Dist 的陣列，在 input 的 function 用 unroll 的概念 read 至 Dist 以減少 I/O time，再用 cudaMemcpy 至 d_Dist，再根據 blocking factor(=32)去分 block，blocking factor = 32 是因為 warp size = 32，32*32 threads/block 是因為 blocking factor = 32，每個 grid 有多少個 block 取決於當時所在的 phase 及正在計算的 row 或 column，ex.正在 phase 2，計算 pivot block 右邊的 row，則將 blocks/grid 設成 blockDim.x = round - r - 1 及 blockDim.y = 1，接著，每次 calculate 都會先把 data load 到 shared memory 去做計算，計算完成後再存回去 device 的 global memory，以減少 memory access 的時間，在每個 phase 結束前都用 cudaDeviceSynchronize()同步，才可開始下個階段，最後 cudaMemcpy 至 Dist 做 output。

## Hw3-3

在 hw3-3，我有試著使用 omp_thread 分給兩個 device 去跑，但發現並沒有跑得比較快，反而有些只用一個 gpu 跑、有 accepted 的測資，分給兩個 gpu 跑的時候卻超時，因此 hw3-3 的 code 跟 3-2 是一樣的。

# Profiling(hw3-2)

實驗設備：課堂 hades
測資：c06.1

- occupancy

```
[pp21s49@hades01 ~]$ srun -p prof -N1 -n1 --gres=gpu:1 nvprof --metrics achieved_occupancy ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==893021== NVPROF is profiling process 893021, command: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==893021== Profiling application: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
devices count: 1
I/O time: 2.000000 ms
kernel time: 190.000000 ms
mem copy time: 0.000000 ms
==893021== Profiling result:
==893021== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 1080 (0)"
    Kernel: cal_1(int, int, int, int*, int)
          4                        achieved_occupancy                        Achieved Occupancy    0.304458    0.496400    0.448160
    Kernel: cal_3(int, int, int, int*, int)
         10                        achieved_occupancy                        Achieved Occupancy    0.316964    0.484857    0.434347
    Kernel: cal_col(int, int, int, int*, int)
          6                        achieved_occupancy                        Achieved Occupancy    0.137417    0.496208    0.397101
    Kernel: cal_row(int, int, int, int*, int)
          6                        achieved_occupancy                        Achieved Occupancy    0.315967    0.496649    0.466300
```

- sm efficiency

```
[pp21s49@hades01 ~]$ srun -p prof -N1 -n1 --gres=gpu:1 nvprof --metrics sm_efficiency ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==893047== NVPROF is profiling process 893047, command: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==893047== Profiling application: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
devices count: 1
I/O time: 2.000000 ms
kernel time: 128.000000 ms
mem copy time: 0.000000 ms
==893047== Profiling result:
==893047== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 1080 (0)"
    Kernel: cal_1(int, int, int, int*, int)
          4                             sm_efficiency                   Multiprocessor Activity       1.74%       3.69%       3.20%
    Kernel: cal_3(int, int, int, int*, int)
         10                             sm_efficiency                   Multiprocessor Activity       1.83%      19.79%       7.42%
    Kernel: cal_col(int, int, int, int*, int)
          6                             sm_efficiency                   Multiprocessor Activity       3.24%       9.91%       6.10%
    Kernel: cal_row(int, int, int, int*, int)
          6                             sm_efficiency                   Multiprocessor Activity       3.60%      11.23%       6.43%
```

- shared memory load/store throughput

```
[pp21s49@hades01 ~]$ srun -p prof -N1 -n1 --gres=gpu:1 nvprof --metrics shared_load_throughput,shared_store_throughput ./hw3-2 /home/pp21/share/hw3-2/ca
ses/c06.1 tmp.out
==893072== NVPROF is profiling process 893072, command: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==893072== Profiling application: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
devices count: 1
I/O time: 4.000000 ms
kernel time: 170.000000 ms
mem copy time: 0.000000 ms
==893072== Profiling result:
==893072== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 1080 (0)"
    Kernel: cal_1(int, int, int, int*, int)
          4                      shared_load_throughput             Shared Memory Load Throughput  3.0268GB/s  58.401GB/s  50.192GB/s
          4                     shared_store_throughput            Shared Memory Store Throughput  1.8626GB/s  10.499GB/s  8.2010GB/s
    Kernel: cal_3(int, int, int, int*, int)
         10                      shared_load_throughput             Shared Memory Load Throughput  14.234GB/s  446.21GB/s  177.78GB/s
         10                     shared_store_throughput            Shared Memory Store Throughput  2.5636GB/s  33.528GB/s  9.8766GB/s
    Kernel: cal_col(int, int, int, int*, int)
          6                      shared_load_throughput             Shared Memory Load Throughput  9.9684GB/s  115.06GB/s  75.351GB/s
          6                     shared_store_throughput            Shared Memory Store Throughput  2.2095GB/s  10.638GB/s  7.6691GB/s
    Kernel: cal_row(int, int, int, int*, int)
          6                      shared_load_throughput             Shared Memory Load Throughput  8.8052GB/s  165.19GB/s  95.834GB/s
          6                     shared_store_throughput            Shared Memory Store Throughput  2.8670GB/s  33.262GB/s  15.474GB/s
```

- global load/store throughput

```
[pp21s49@hades01 ~]$ srun -p prof -N1 -n1 --gres=gpu:1 nvprof --metrics gst_throughput ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
srun: job 420762 queued and waiting for resources
srun: job 420762 has been allocated resources
==727290== NVPROF is profiling process 727290, command: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
devices count: 1
I/O time: 6.000000 ms
kernel time: 58.000000 ms
mem copy time: 0.000000 ms
==727290== Profiling application: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==727290== Profiling result:
==727290== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 1080 (0)"
    Kernel: cal_1(int, int, int, int*, int)
          4                             gst_throughput                  Global Store Throughput  52.982MB/s  632.85MB/s  540.70MB/s
    Kernel: cal_3(int, int, int, int*, int)
         10                             gst_throughput                  Global Store Throughput  50.863MB/s  17.243GB/s  3.2566GB/s
    Kernel: cal_col(int, int, int, int*, int)
          6                             gst_throughput                  Global Store Throughput  109.34MB/s  1.2028GB/s  843.83MB/s
    Kernel: cal_row(int, int, int, int*, int)
          6                             gst_throughput                  Global Store Throughput  131.54MB/s  1.2789GB/s  795.89MB/s
```

```
[pp21s49@hades01 ~]$ srun -p prof -N1 -n1 --gres=gpu:1 nvprof --metrics gld_throughput ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==893638== NVPROF is profiling process 893638, command: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
==893638== Profiling application: ./hw3-2 /home/pp21/share/hw3-2/cases/c06.1 tmp.out
devices count: 1
I/O time: 10.000000 ms
kernel time: 196.000000 ms
mem copy time: 0.000000 ms
==893638== Profiling result:
==893638== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 1080 (0)"
    Kernel: cal_1(int, int, int, int*, int)
          4                             gld_throughput                   Global Load Throughput  290.34MB/s  468.70MB/s  334.33MB/s
    Kernel: cal_3(int, int, int, int*, int)
         10                             gld_throughput                   Global Load Throughput  625.17MB/s  6.2106GB/s  2.3562GB/s
    Kernel: cal_col(int, int, int, int*, int)
          6                             gld_throughput                   Global Load Throughput  456.98MB/s  1.7048GB/s  959.78MB/s
    Kernel: cal_row(int, int, int, int*, int)
          6                             gld_throughput                   Global Load Throughput  443.71MB/s  1.7305GB/s  935.53MB/s
```

## Experiment & Analysis

實驗設備：課堂 hades
測資：c21.1
時間計算方式： std::chrono::steady_clock
Computing time：在執行 block_FW 所花的時間
Memory copy time: 進行 cudaMemcpy - HostToDevice & DeviceToHost 所花的時間
I/O time：在 read 跟 write 檔案時所花的時間

- **Time distribution(hw3-2)：**

Computing time: 334.000000 ms
Memory copy time: 30.000000 ms
I/O time: 712.000000 ms

- **Optimization(hw3-2)**

在 input 函式中使用 unroll 的概念來存進 Dist。

使用 blockIdx & threadIdx 平行化(CUDA 2D alignment)得把需要用來判斷的 block 的 data load 至 shared memory，再利用 shared memory 中的 data 來計算 shortest path，計算完成後再存回 Dist(device's global variable)。

## Experience & Conclusion

這次作業我學到如何寫 cuda，以及更深的了解 GPU 的 grid、block、thread 的概念，還有 device 的 global memory 跟 shared memory access 的時間真的差很多，這次作業難度很高，在 cuda 的撰寫遇到了很多困難，在如何設置 blockIdx 和 threadIdx 這個地方就花了很多時間，即便做了 blockIdx 和 threadIdx 的平行化及使用 shared memory，在 hw3-2 p19 以後的測資還是沒過，最後在 hw3-3，我有試著使用 omp_thread 分給兩個 device 去跑，但發現並沒有跑得比較快，蠻想知道兩個 gpu 的寫法。