

Parallel Programming Homework4

106065509

林庭宇

1. Implementation

1.1. How do you divide your data

- Openmp: 這邊先透過 $\text{round} = \text{ceil}(n, B)$ 算出整張圖總共有幾個blocks要分，再上下切一半，各分給device 0,1。
- Mpi: 一樣先透過 $\text{round} = \text{ceil}(n, B)$ 算出整張圖總共有幾個blocks要分，接著根據 MPI_Comm_size 決定一個process負責的block數。

1.2. How do you implement the communication?

- Openmp: 為了方便解說將負責上半圖的thread稱作thread0，使用device0，負責下半圖的則是thread1，使用device1。

Thread0和thread1都會跑round次blocked FW。每個round的phase 1和phase 2都是做一樣的事，然而在phase 3則指做自己負責的那幾個blocks就好。當每一輪的phase 3結束，便會判斷下一輪(r+1)的phase 1落在整張圖中的上半或下半，若是在下半，thread0就需要將device1中第(r+1)個block所在的整個row都複製到device0中。

- Mpi: 使用 MPI_Graph_create 來建和openmp做一樣的事情，但因為同時會有許多process要和(r+1)所在的process拿資料，所以我使用的是 MPI_Bcast 。

而由於我們的MPI不支援Cuda-Aware MPI，傳資料時只好先用 cudaMemcpy2D 把data從gpu拉到host的send_buff，再傳出去。

1.3. AWhat's your configuration

- Cuda/Openmp/Mpi:
 - blocking factor = 64,
 - #blocks = $\text{ceil}(n, B)$,
 - #threads = (32, 32)