1.h2_problem1:

1.What have you done?

MPI 變數方面我宣告了:

(1)local_rank : 現在在哪個 rank

(2)comm_size : number of process

(3)int *global_bmp_size : 一個紀錄 bmp size 在每個 process 的 array

(4)int *global_bmp_displacements : 一個紀錄 bmp displacement for

each process 的 array

(5)local_bmp_height : Height of the local_bmp

(6)RGBTRIPLE **local_bmp_data : 暫存 local bmp data 的地方

(7) RGBTRIPLE **local_bmp_save_data :存 local bmp data

(8) RGBTRIPLE **local_bmp_upper_temp : upper storage
(9) RGBTRIPLE **local_bmp_lower_temp : lower storage

接著 read_bmp，讀完 bmp 後會把 bmpInfo.biHeight 和

bmpInfo.BiWidth ‘’Broadcast’’ 給每個 process

之後算 local_bmp_height 和 local_bmp_size，公式為:

```
local_bmp_height = (bmpInfo.biHeight / comm_size) +
                   ((bmpInfo.biHeight % comm_size >= local_rank + 1) ?
                    1 : 0);
int local_bmp_size = bmpInfo.biWidth * local_bmp_height;
```

接著把每個 process 被分割的資料 ’’Gather” 起來

然後計算 displacement，公式為:

```
global_bmp_displacements[i] = ((i != kProcessRoot) ?
                                    (global_bmp_displacements[i - 1] +
                                        global_bmp_size[i - 1]) :
                                    0);
```

分配記憶體空間給 local_bmp_data、local_bmp_save_data、

local_bmp_upper_temp、local_bmp_lower_temp

再把圖片 “”Scatterv”” 給每個 process

開始 1000 次平滑。每次平滑都會計算 left_partner 和 right_partner，

然後分別從左向右、從右向左平移數據，運算完才開始進行 smoothing.

接著 1000 次平滑完成後，把照片”Gatherv”給 root process，接著 saveBMP.

2.Analysis on your result

(1)分別用 4、8、12 個 process 的執行時間: 發現 4 個 process 花的時間幾乎是 8 個 process 的兩倍，而且 8 個 process 花的時間比 12 個 process 花的時間短，推測 12 process 要 send、receive 所花的時間太多會導致執行速度並未隨著 process 數增加而提升。

```
H540084078@pn1:~> mpiexec -n 4 ./h2_problem1.out
Read file successfully!!
The execution time = 26.007
Save file successfully!!
H540084078@pn1:~> mpiexec -n 8 ./h2_problem1.out
Read file successfully!!
The execution time = 13.7467
Save file successfully!!
H540084078@pn1:~> mpiexec -n 12 ./h2_problem1.out
Read file successfully!!
The execution time = 17.5149
Save file successfully!!
H540084078@pn1:~>
```

(2)平滑次數 ：10 vs. 1000 vs. 10000 次，output.bmp 照片比較

(8process)

原圖:

10 次:

1000 次:

10000 次:

=>平滑次數越多次，output 照片越模糊

3.Any difficulties?

這個作業要先弄懂如何影像分割和傳資料，我覺得蠻難的，不過搞清楚後，會覺得很有條理，很佩服前人的智慧。

4.(optional) Feedback to Tas

我覺得作業 2 跟作業 1 都很難，不知道以後能不能把作業的 deadline 延長?

2.h2_problem2:

1.What have you done

我定義 3 個函式:

(1)arrayCompare : 比較兩個值

(2)swap : 交換兩個值

(3)mergeArray : 把兩個 array 合併

然後我宣告

1.total_num : total number of array
2.*global_num_array : global array

3.*global_num_array_displacement : 在 global array 裡每個 process 的 displacement

4.*global_num_array_size : 在 global array 裡每個 process 的 array 的 size

MPI 變數方面宣告:

1.comm_size : number of process

2.local_rank : 目前的 rank

3.local_remainder_check : check 目前的 process 需不需要增加 1 或多個 value 到 array

4.*local_num_array : local number array

5.local_num_array_size : local number array 的 size

6. *recv_num_array : receiver 的 number array

7.recv_num_array_size : receiver's number array 的 size

接著可以接收使用者輸入的數字

然後把 total_num ”Bcast” 給每個 process

檢查目前的 process 需不需要增加 1 或多個 value 到 array 並計算出 local_num_array_size

之後使用亂數種子讓 Each process 可以 create 出 a local list of n / comm_sz 個 ints.

之後就用 qsort 排序 local array

然後把 array length “Gather” 到 global_num_array_size，然後算出 array 的 displacement 再把 local array “Gatherv”給 global_num_array 並印出各個 process 裡的 local array

接下來進行 odd-even-sort(執行 comm_sz 次) :

1.如果是 odd phase, local partner= local_rank + ((local_rank % 2 == 0) ? -1 : 1)
2.如果是 even phase, local_partner = local_rank + ((local_rank % 2 == 0) ? 1 : -1)

之後在 partner 之間會進行大小比較，並且儲存需要的值在 local array. 較小的 rank 存 the smaller value 而較大的 rank 存 the larger value(呼叫 mergeArray function)，直到做完 number of process(comm_sz)次，odd-even-sort 結束。

odd-even-sort 結束後，把 local array “Gatherv” 到 global array，並印

出 global array.

2.Analysis on your result

(1)排序越多數字，花的時間越長(以 8 process 為例):

排 10 個數字的時間<排 100 個數字的時間<排 1000 個數字的時間<排 10000 個數字的時間

10:

```
Enter how many number you want to calculate:10

Local array:

process 0 local array:
Rank:   0 , No.     1 , 846930886
Rank:   0 , No.     2 , 1804289383

process 1 local array:
Rank:   1 , No.     1 , 611911301
Rank:   1 , No.     2 , 677741240

process 2 local array:
Rank:   2 , No.     1 , 331330603

process 3 local array:
Rank:   3 , No.     1 , 1064858753

process 4 local array:
Rank:   4 , No.     1 , 1784236095

process 5 local array:
Rank:   5 , No.     1 , 1433930398

process 6 local array:
Rank:   6 , No.     1 , 22703042

process 7 local array:
Rank:   7 , No.     1 , 1817947203

Global Array:
No.     1 , 22703042
No.     2 , 331330603
No.     3 , 611911301
No.     4 , 611911301
No.     5 , 846930886
No.     6 , 1064858753
No.     7 , 1433930398
No.     8 , 1784236095
No.     9 , 1804289383
No.    10 , 1817947203

The execute Time: 0.007008
H54084078@pn1:~>
```

100:

```
No.    62 , 1259609597
No.    63 , 1277102819
No.    64 , 1350490027
No.    65 , 1365076974
No.    66 , 1396803471
No.    67 , 1433930398
No.    68 , 1474515219
No.    69 , 1475816657
No.    70 , 1528734569
No.    71 , 1545004831
No.    72 , 1553434310
No.    73 , 1609665824
No.    74 , 1649760492
No.    75 , 1650016564
No.    76 , 1664181092
No.    77 , 1681692777
No.    78 , 1710201796
No.    79 , 1714636915
No.    80 , 1717881694
No.    81 , 1721903022
No.    82 , 1775430624
No.    83 , 1784236095
No.    84 , 1791422769
No.    85 , 1804031483
No.    86 , 1804289383
No.    87 , 1817947203
No.    88 , 1836062090
No.    89 , 1842344224
No.    90 , 1886588038
No.    91 , 1893586918
No.    92 , 1956612032
No.    93 , 1957747793
No.    94 , 2083069270
No.    95 , 2099549627
No.    96 , 2104660314
No.    97 , 2106822048
No.    98 , 2120519271
No.    99 , 2123082299
No.   100 , 2138423562

The execute Time: 0.007907
H54084078@pn1:~>
```

1000:

```
No.   961 , 2083069270
No.   962 , 2084374456
No.   963 , 2084420925
No.   964 , 2085016816
No.   965 , 2087336188
No.   966 , 2089018456
No.   967 , 2099517135
No.   968 , 2099549627
No.   969 , 2102271570
No.   970 , 2103781290
No.   971 , 2104660314
No.   972 , 2106822048
No.   973 , 2111748931
No.   974 , 2112311639
No.   975 , 2113469436
No.   976 , 2114253222
No.   977 , 2114738097
No.   978 , 2115672838
No.   979 , 2116767386
No.   980 , 2116988433
No.   981 , 2117035652
No.   982 , 2117115292
No.   983 , 2117386918
No.   984 , 2120519271
No.   985 , 2120675190
No.   986 , 2123082299
No.   987 , 2124641190
No.   988 , 2128538674
No.   989 , 2128617938
No.   990 , 2129613864
No.   991 , 2131132462
No.   992 , 2132805286
No.   993 , 2137335155
No.   994 , 2137786030
No.   995 , 2138423562
No.   996 , 2141135886
No.   997 , 2142393982
No.   998 , 2145174067
No.   999 , 2146790443
No.  1000 , 2146967231

The execute Time: 0.009520
H54084078@pn1:~>
```

10000:

```
No.  9958 , 2138232043
No.  9959 , 2138295400
No.  9960 , 2138352249
No.  9961 , 2138423562
No.  9962 , 2138959392
No.  9963 , 2138982933
No.  9964 , 2139072608
No.  9965 , 2139442706
No.  9966 , 2139865461
No.  9967 , 2140064406
No.  9968 , 2141102332
No.  9969 , 2141111200
No.  9970 , 2141135886
No.  9971 , 2141350397
No.  9972 , 2141548647
No.  9973 , 2141558545
No.  9974 , 2141696612
No.  9975 , 2141701042
No.  9976 , 2141775443
No.  9977 , 2141821581
No.  9978 , 2141906483
No.  9979 , 2141943612
No.  9980 , 2142393982
No.  9981 , 2142626740
No.  9982 , 2142757034
No.  9983 , 2143124030
No.  9984 , 2143404004
No.  9985 , 2143520255
No.  9986 , 2143841581
No.  9987 , 2144187616
No.  9988 , 2144824565
No.  9989 , 2145035110
No.  9990 , 2145098855
No.  9991 , 2145174067
No.  9992 , 2145323337
No.  9993 , 2145432356
No.  9994 , 2145809671
No.  9995 , 2145879112
No.  9996 , 2146211838
No.  9997 , 2146790443
No.  9998 , 2146967231
No.  9999 , 2147156862
No. 10000 , 2147469841

The execute Time: 0.149213
H54084078@pn1:~>
```

(2)不同 process 數量，執行時間差異(10 number):

12 process 的執行時間>8 process 的執行時間>4 process 的執行時間

4 process:

```
H54084078@pn1:~> mpiexec -n 4 ./h2_problem2.out
Enter how many number you want to calculate:10

Local array:

process 0 local array:
Rank:   0 , No.     1 , 846930886
Rank:   0 , No.     2 , 1681692777
Rank:   0 , No.     3 , 1804289383

process 1 local array:
Rank:   1 , No.     1 , 516687479
Rank:   1 , No.     2 , 611911301
Rank:   1 , No.     3 , 677741240

process 2 local array:
Rank:   2 , No.     1 , 197953680
Rank:   2 , No.     2 , 331330603

process 3 local array:
Rank:   3 , No.     1 , 868075535
Rank:   3 , No.     2 , 1064858753

Global Array:
No.     1 , 197953680
No.     2 , 331330603
No.     3 , 516687479
No.     4 , 611911301
No.     5 , 677741240
No.     6 , 846930886
No.     7 , 868075535
No.     8 , 1064858753
No.     9 , 1681692777
No.    10 , 1804289383

The execute Time: 0.000138
H54084078@pn1:~>
```

8 process:

```
H54084078@pn1:~> mpiexec -n 8 ./h2_problem2.out
Enter how many number you want to calculate:10

Local array:

process 0 local array:
Rank:   0 , No.     1 , 846930886
Rank:   0 , No.     2 , 1804289383

process 1 local array:
Rank:   1 , No.     1 , 611911301
Rank:   1 , No.     2 , 677741240

process 2 local array:
Rank:   2 , No.     1 , 331330603

process 3 local array:
Rank:   3 , No.     1 , 1064858753

process 4 local array:
Rank:   4 , No.     1 , 1784236095

process 5 local array:
Rank:   5 , No.     1 , 1433930398

process 6 local array:
Rank:   6 , No.     1 , 22703042

process 7 local array:
Rank:   7 , No.     1 , 1817947203

Global Array:
No.     1 , 22703042
No.     2 , 331330603
No.     3 , 611911301
No.     4 , 611911301
No.     5 , 846930886
No.     6 , 1064858753
No.     7 , 1433930398
No.     8 , 1784236095
No.     9 , 1804289383
No.    10 , 1817947203

The execute Time: 0.006713
H54084078@pn1:~>
```

12 process:

```
H54084078@pn1:~> mpiexec -n 12 ./h2_problem2.out
Enter how many number you want to calculate:10

Local array:

process 0 local array:
Rank:   0 , No.     1 , 1804289383

process 1 local array:
Rank:   1 , No.     1 , 677741240

process 2 local array:
Rank:   2 , No.     1 , 331330603

process 3 local array:
Rank:   3 , No.     1 , 1064858753

process 4 local array:
Rank:   4 , No.     1 , 1784236095

process 5 local array:
Rank:   5 , No.     1 , 1433930398

process 6 local array:
Rank:   6 , No.     1 , 22703042

process 7 local array:
Rank:   7 , No.     1 , 1817947203

process 8 local array:
Rank:   8 , No.     1 , 1466417113

process 9 local array:
Rank:   9 , No.     1 , 1131482220

Global Array:
No.     1 , 22703042
No.     2 , 331330603
No.     3 , 677741240
No.     4 , 1064858753
No.     5 , 1131482220
No.     6 , 1433930398
No.     7 , 1466417113
No.     8 , 1784236095
No.     9 , 1804289383
No.    10 , 1817947203

The execute Time: 0.009496
```

推測 process 越多，傳遞資料花的時間>平行運算減少的時間

3.Any difficulties?

花了很多時間了解 odd-even-sort，實作過程中也遇到很多困難

4.(optional) Feedback to Tas

希望以後作業繳交 deadline 可以延長