

**Problem 1**

這題因為大多數的程式碼已經有附上了所以比較難的是在算總合的地方利用 tree 的方式加總。在算每個 process 找到的數量只需要用題目給的程式碼並將迴圈的地方改成由 process id 開始且每隔總 process 數量跳著計算, 就可以將所有要計算的次數全部平分給每個 process 來分工完成, 這也代表越多台 process 分工每台跑的時間就越少。由下圖結果可以得知我使用 10 台 process 跑程式碼, 花費時間最長的一台只需 0.000315 秒, 當我換成 4 台 process 時則花費時間最短的一台就需要 0.000449 秒, 很明顯就可以看出差別。

```
H34096102@sivslab-pn1:~/HW1$ mpiexec -n 10 ./problem1
4) 1001100111110110
6) 1001101111110110
8) 1001110111110110
3) 1001100111110101
7) 1001101111110111
7) 1001110111110101
5) 1001100111110111
9) 1001110111110111
5) 1001101111110101
Process 6 finished in time 0.000193 secs. 1 solutions were found.
Process 8 finished in time 0.000188 secs. 1 solutions were found.
Process 7 finished in time 0.000269 secs. 2 solutions were found.
Process 2 finished in time 0.000274 secs. 0 solutions were found.
Process 3 finished in time 0.000285 secs. 1 solutions were found.
Process 9 finished in time 0.000286 secs. 1 solutions were found.
Process 5 finished in time 0.000303 secs. 2 solutions were found.
Process 1 finished in time 0.000310 secs. 0 solutions were found.
Process 4 finished in time 0.000305 secs. 1 solutions were found.
Process 0 finished in time 0.000315 secs. A total of 9 solutions were found.
H34096102@sivslab-pn1:~/HW1$ mpiexec -n 4 ./problem1
3) 1001100111110111
3) 1001101111110111
3) 1001110111110111
1) 1001100111110101
1) 1001101111110101
1) 1001110111110101
2) 1001100111110110
2) 1001101111110110
2) 1001110111110110
Process 3 finished in time 0.000449 secs. 3 solutions were found.
Process 1 finished in time 0.000462 secs. 3 solutions were found.
Process 2 finished in time 0.000457 secs. 3 solutions were found.
Process 0 finished in time 0.000461 secs. A total of 9 solutions were found.
H34096102@sivslab-pn1:~/HW1$
```

在計算 solution 總和是用 tree structure 的方法, 一開始設一個變數" process" 用來計算還沒被 tree structure 計算過 count 數目的 process 數量(也就是一開始等於總 process 數), 以及變數" offset" 用作分辨是否要重新進入迴圈繼續 send&receive message, 每次的迴圈都會處理掉一半的 process, 當仍有多個 process 尚未處理則由 offset 來判斷進入下一次的迴圈。

接著分為兩大類: 當 process id 小於(" process" /2)則做為接收方, 當 process id 小於" process" 且不小於 offset 則作為傳遞方。不停地將大 id 傳給小 id 直到剩下一個便停止。這樣的做法就可以利用 tree structure 完成點對點傳遞。

這次的作業較困難的地方應該是思考樹狀傳遞的那塊, 花了不少時間在研究上課的簡報, 除了對指令不熟悉之外也要特別搜尋那些傳遞參數該放甚麼比較正確, 其餘的部分題目以及上課的時候就講得蠻清楚所以在多平行處理機的分配上比較不花時間。

## Problem 2

這題的做法與上議題幾乎一樣, 不論是在使用 tree structure 來點對點傳送資料、加總數量, 或者用迴圈分配 process 跑程式分工合作都是使用相同的方法, 但這題我覺得比較有趣的是有多了一個可以分析的數據。從下方第一張圖可以很明顯的看出用 10 台 process 跑程式時間大約都需要花 0.0003~0.0004 秒, 但改成 20 台來跑之後僅需 0.0001~0.0002 秒即可完成, 這與上一題相同的分析結果。不過從第二張圖可以看到, 將 toss 的數量由 1000 增至 10000000, 計算出來的 pi 值可以差很多且更精確, 當然, 因為要計算的次數倍增所以花費的時間也會增加很多。

```
H34096102@sivslab-pn1:~/HW1$ mpicc problem2.c -o ./problem2
H34096102@sivslab-pn1:~/HW1$ mpirun -n 10 ./problem2
Process 5 finished in time 0.000413 secs.
Process 7 finished in time 0.000388 secs.
Process 9 finished in time 0.000414 secs.
Process 2 finished in time 0.000345 secs.
Process 8 finished in time 0.000326 secs.
Process 4 finished in time 0.000329 secs.
Process 6 finished in time 0.000344 secs.
Process 1 finished in time 0.000347 secs.
Process 0 finished in time 0.000335 secs. PI: 3.1335999999999999, number of tosses is: 100000
Process 3 finished in time 0.000345 secs.
H34096102@sivslab-pn1:~/HW1$ mpirun -n 20 ./problem2
Process 16 finished in time 0.000168 secs.
Process 12 finished in time 0.000199 secs.
Process 11 finished in time 0.000236 secs.
Process 14 finished in time 0.000185 secs.
Process 6 finished in time 0.000160 secs.
Process 15 finished in time 0.000181 secs.
Process 5 finished in time 0.000199 secs.
Process 10 finished in time 0.000182 secs.
Process 13 finished in time 0.000185 secs.
Process 19 finished in time 0.000187 secs.
Process 17 finished in time 0.000163 secs.
Process 9 finished in time 0.000162 secs.
Process 18 finished in time 0.000186 secs.
Process 8 finished in time 0.000169 secs.
Process 3 finished in time 0.000186 secs.
Process 7 finished in time 0.000164 secs.
Process 2 finished in time 0.000165 secs.
Process 4 finished in time 0.000198 secs.
Process 1 finished in time 0.000188 secs.
Process 0 finished in time 0.000198 secs. PI: 3.1688000000000001, number of tosses is: 100000
H34096102@sivslab-pn1:~/HW1$
```

```
H34096102@sivslab-pn1:~/HW1$ mpicc problem2.c -o ./problem2
H34096102@sivslab-pn1:~/HW1$ mpiexec -n 10 ./problem2
Process 6 finished in time 0.000006 secs.
Process 7 finished in time 0.000005 secs.
Process 2 finished in time 0.000005 secs.
Process 8 finished in time 0.000005 secs.
Process 3 finished in time 0.000006 secs.
Process 9 finished in time 0.000006 secs.
Process 5 finished in time 0.000006 secs.
Process 4 finished in time 0.000006 secs.
Process 1 finished in time 0.000006 secs.
Process 0 finished in time 0.000006 secs. PI: 2.9600000000000000, number of tosses is: 1000
H34096102@sivslab-pn1:~/HW1$ mpicc problem2.c -o ./problem2
H34096102@sivslab-pn1:~/HW1$ mpiexec -n 10 ./problem2
Process 9 finished in time 0.050016 secs.
Process 4 finished in time 0.071624 secs.
Process 7 finished in time 0.083937 secs.
Process 2 finished in time 0.047322 secs.
Process 6 finished in time 0.074091 secs.
Process 1 finished in time 0.066613 secs.
Process 5 finished in time 0.100106 secs.
Process 8 finished in time 0.094358 secs.
Process 3 finished in time 0.071551 secs.
Process 0 finished in time 0.072013 secs. PI: 3.1412439999999999, number of tosses is: 10000000
H34096102@sivslab-pn1:~/HW1$
```