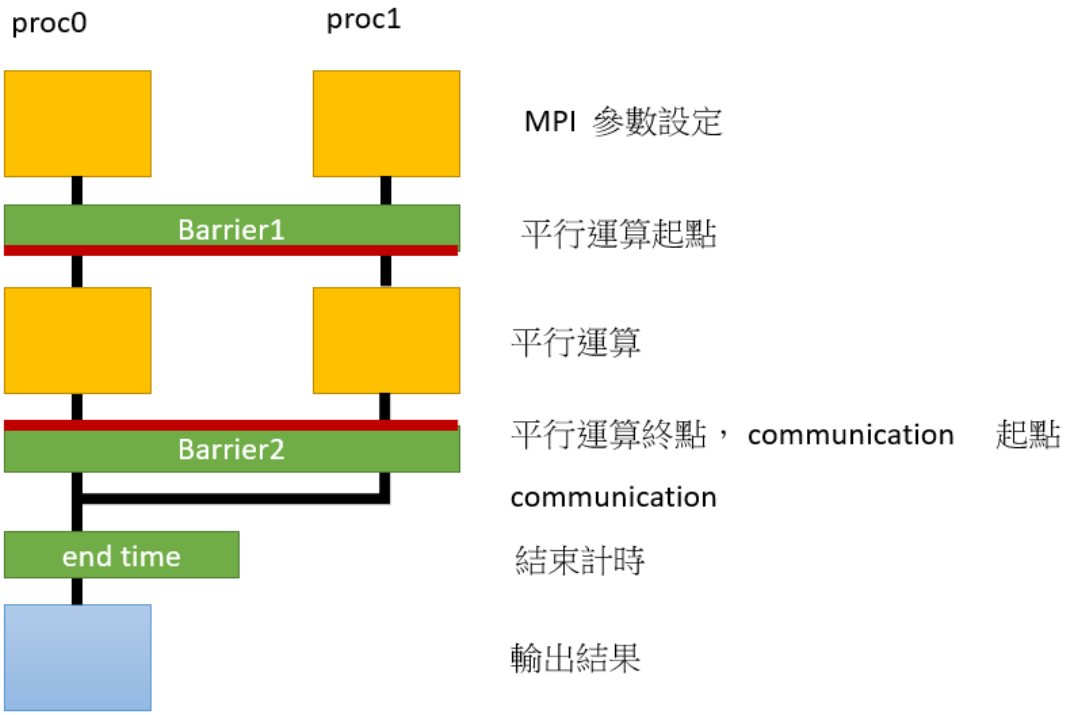
平行程式設計 作業 1

F74081129 資訊 112 吳信葆

What have you done

程式架構

我的兩個題目都使用了以下程式碼架構，方便我去測量時間，



在這個架構中，我會設計如果輸入的 argv 包含 'a'，就是 all_test 模式，我就會在每個 proc 進行平行運算的分開測量(barrier1 到 barrier2 (紅線部分))，並且會讓 proc0 進行 communication 時間之測量(barrier2 到 end time)。

否則我只由 proc0 測量總時間(barrier1 到 end time)。

問題二一開始要廣播 number of tossed，我是放在平行運算區塊裡。

注意!!!
end time block 只由 proc0 運算，且此為沒參數版本之 end time。
barrier 之後才進行起始時間運算，結束時間則是在起始時間之前。

Tree Communication

皆由自製函式

```
int My_Sum(int id, int val,int n)
```

完成，參數分別是

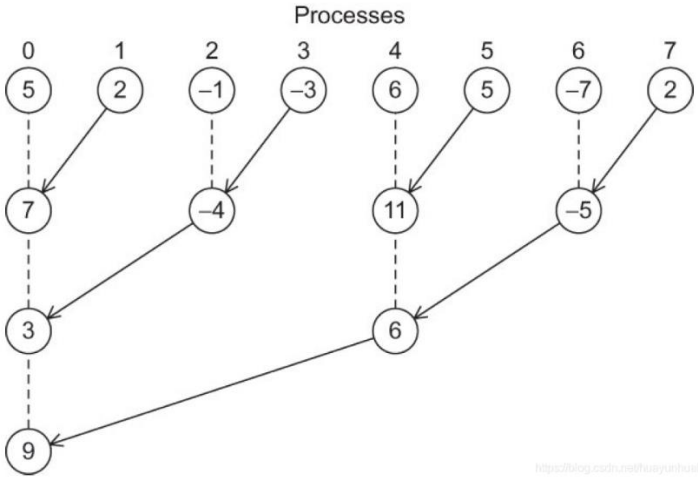
Id: proc id

Val: proc 各自結果

N: number of procs

回傳與最後自己 partner 總和(proc 0 因此會是所有 proc 之和)。

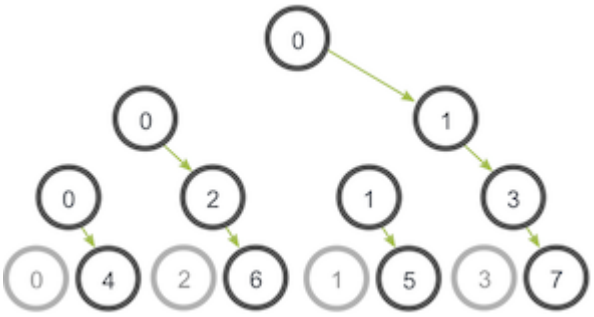
採用的方式是 tree communication，迴圈寫法，詳細看 problem1.c 程式碼註解，以下是流程圖。



作業二新增

```
long long int My_Bcast(int id, long long int val, int n)
```

採用的方式是 tree communication，迴圈寫法，詳細看 problem2.c 程式碼註解，以下是參考圖。



Analysis on your result

效能

第一題

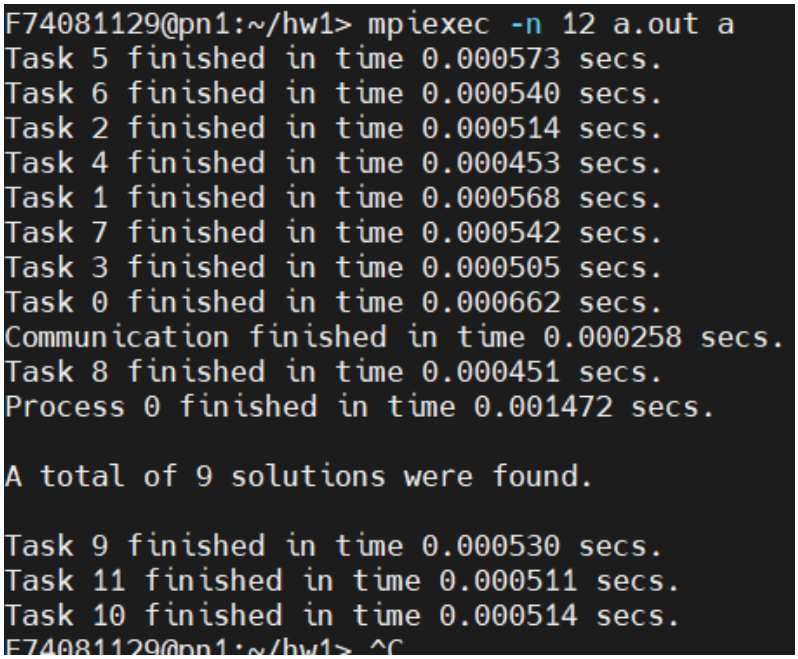
以下 Monte Carlo method 的 sample 點個數為 1,000,000。

針對不同核心數執行 10 次取執行時間平均。

核心數	1	2	4	12	24
	0.011721	0.004822	0.001954	0.00221	0.000447
	0.011024	0.005763	0.001681	0.000578	0.000801
	0.009245	0.004299	0.001884	0.000938	0.00032
	0.00884	0.003878	0.00188	0.000688	0.001247
	0.011107	0.004856	0.001881	0.002039	0.001507
	0.013832	0.004462	0.002259	0.00163	0.000957
	0.009796	0.004736	0.001902	0.001196	0.01203
	0.008864	0.005948	0.002253	0.000737	0.000567
	0.011976	0.004964	0.002079	0.000689	0.000446
	0.014204	0.005522	0.002275	0.000499	0.000441
平均時間(s)	0.011061	0.004925	0.002005	0.00112	0.001876
標準差(s)	0.00183	0.000621	0.000192	0.000594	0.003405

我們發現 12 顆核心效果最好，因為 server 的核心只有 12 顆，12 顆全開便能榨乾三台伺服器效能。

於是我們在看 12 顆處理器完成 task 所需時間之平均(in all_test mode)。



經過計算大概是 0.00053025s ，而最後的 communication time 是 0.000258s。

我發現 Task 執行時間與 communication 們差多少，可能不太適合做 mpi 。

第二題

針對不同核心數執行 12 次取執行時間平均。

核心數	1	2	4	12	24
	0.029804	0.019988	0.010804	0.003441	0.008791
	0.036325	0.022945	0.008418	0.003628	0.006936
	0.030427	0.024072	0.010307	0.003445	0.004433
	0.035092	0.017684	0.00798	0.00369	0.008137
	0.030177	0.0162	0.010166	0.003409	0.003063
	0.033345	0.019034	0.010248	0.003585	0.008516
	0.041213	0.01994	0.008095	0.003438	0.006364
	0.031041	0.02139	0.010168	0.003453	0.004319
	0.03872	0.022127	0.010182	0.003645	0.005949
	0.032876	0.024141	0.012149	0.003428	0.010786
平均時間	0.033902	0.020752	0.009852	0.003516	0.006729

可以看到一樣是 12 顆核心最好。

於是我們在看 12 顆處理器完成 task 所需時間之平均(in all_test mode)。

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out a
Task 0 finished in time 0.003529 secs.
Communication finished in time 0.000016 secs.
pi is approximately 3.1437959671020508, Error is 0.0022033135122577
wall clock time = 0.004078
Task 1 finished in time 0.002311 secs.
Task 5 finished in time 0.002357 secs.
Task 6 finished in time 0.002330 secs.
Task 2 finished in time 0.002343 secs.
Task 4 finished in time 0.002490 secs.
Task 7 finished in time 0.002365 secs.
Task 3 finished in time 0.002509 secs.
Task 8 finished in time 0.002532 secs.
Task 9 finished in time 0.002483 secs.
Task 10 finished in time 0.002371 secs.
Task 11 finished in time 0.002340 secs.
F74081129@pn1:~/hw1>
```

平均時間是 0.002496667 ，communication 時間是 0.000016。

Monte Carlo method 結果

我們知道，當我們 toss 次數越多，理論上結果會越逼近 真實圓周率，我們來測試一下。

我會在 all test mode is false 且 12 顆核心去改變 toss 次數，得出 估計圓周率，我是以 proc rank 作為亂數種，所以只測一次。

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 10
pi is approximately 3.5999999046325684, Error is 0.4584072510427752
wall clock time = 0.000115
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 100
pi is approximately 3.3599998950958252, Error is 0.2184072415060321
wall clock time = 0.000246
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 1000
pi is approximately 3.0840001106262207, Error is -0.0575925429635724
wall clock time = 0.001741
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 10000
pi is approximately 3.1340000629425049, Error is -0.0075925906472882
wall clock time = 0.000990
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 100000
pi is approximately 3.1421599388122559, Error is 0.0005672852224627
wall clock time = 0.000397
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 1000000
pi is approximately 3.1437959671020508, Error is 0.0022033135122577
wall clock time = 0.002722
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 10000000
pi is approximately 3.1425664424896240, Error is 0.0009737888998309
wall clock time = 0.049761
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 100000000
pi is approximately 3.1420993804931641, Error is 0.0005067269033709
wall clock time = 0.273015
```

```
F74081129@pn1:~/hw1> mpiexec -n 12 a.out 1000000000
pi is approximately 3.1419978141784668, Error is 0.0004051605886737
wall clock time = 3.167992
```

結果符合理論。

Any difficulties ?

因為在 p1 p2 都沒有各自的問題，所以合併討論，我本次兩題都有用統計的方式去評測效能，但把結果一個一個複製很沒效率，所以下次應該會使用腳本去完成統計任務。

測試方式

環境: 老師的伺服器

```
F74081129@pn1:~/hw1> mpdtrace
pn1
pn4
pn2
```

第一題

編譯: *mpicc problem1.c -o problem1.out*

執行: *mpiexec -n 8 problem1.out*

```
F74081129@pn1:~/hw1> mpiexec -n 8 problem1.out
Process 0 finished in time 0.000727 secs.

A total of 9 solutions were found.
```

執行 all test mode: *mpiexec -n 8 problem1.out a*

```
F74081129@pn1:~/hw1> mpiexec -n 8 problem1.out a
Task 4 finished in time 0.000650 secs.
Task 1 finished in time 0.000767 secs.
Task 7 finished in time 0.000774 secs.
Task 3 finished in time 0.000703 secs.
Task 6 finished in time 0.000797 secs.
Task 5 finished in time 0.000820 secs.
Task 2 finished in time 0.000722 secs.
Task 0 finished in time 0.000874 secs.
Communication finished in time 0.000184 secs.
Process 0 finished in time 0.001472 secs.

A total of 9 solutions were found.
```

第二題

編譯: *mpicc problem2.c -o problem2.out*

執行: *mpiexec -n 8 problem2.out*

```
F74081129@pn1:~/hw1> mpiexec -n 8 problem2.out
pi is approximately 3.1433279514312744, Error is 0.0017352978414813
wall clock time = 0.005180
```

執行 all test mode: *mpiexec -n 8 problem2.out a*

```
F74081129@pn1:~/hw1> mpiexec -n 8 problem2.out a
Task 1 finished in time 0.003510 secs.
Task 3 finished in time 0.003899 secs.
Task 2 finished in time 0.003506 secs.
Task 0 finished in time 0.005157 secs.
Communication finished in time 0.000707 secs.
pi is approximately 3.1433279514312744, Error is 0.0017352978414813
wall clock time = 0.005879
Task 4 finished in time 0.003674 secs.
Task 7 finished in time 0.003459 secs.
Task 5 finished in time 0.003516 secs.
Task 6 finished in time 0.003647 secs.
```

執行自訂 toss 個數(default is 1,000,000，可與 all test mode 混合執行):

mpiexec -n 8 problem2.out 200000000

```
F74081129@pn1:~/hw1> mpiexec -n 8 problem2.out 200000000
pi is approximately 3.1420946121215820, Error is 0.0005019585317889
wall clock time = 1.017556
```