Hw1

測試環境

OS

physical machine ubuntu 20.04

CPU

Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz

Thread(s) per core: 2
Core(s) per socket: 6
Socket(s): 1

```
yellow@yellow-arg7:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   39 bits physical, 48 bits virtual
CPU(s):                          12
On-line CPU(s) list:             0-11
Thread(s) per core:              2
Core(s) per socket:              6
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           158
Model name:                      Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
Stepping:                        10
CPU MHz:                         800.006
CPU max MHz:                     4600.0000
CPU min MHz:                     800.0000
BogoMIPS:                        6399.96
L1d cache:                       192 KiB
L1i cache:                       192 KiB
L2 cache:                        1.5 MiB
L3 cache:                        12 MiB
NUMA node0 CPU(s):               0-11
```

memory size

```
yellow@yellow-arg7:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:            31Gi       5.1Gi        15Gi       319Mi        10Gi        25Gi
Swap:          2.0Gi          0B       2.0Gi
```

kernel version

5.11.0-37-generic

How to run

編譯所有程式，執行所有測試並生成csv和png。

`$ source run_hw1.sh`

Output

multi-process (integer)

```
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./multiprocess_int.out 99999999 1
test setting: datasize=99999999, process_num=1
time cost is 123750 ms  ,Integer 13 occurs 177 times in the array
time cost is 124223 ms  ,Integer 13 occurs 177 times in the array
time cost is 124549 ms  ,Integer 13 occurs 177 times in the array
time cost is 125141 ms  ,Integer 13 occurs 177 times in the array
time cost is 125551 ms  ,Integer 13 occurs 177 times in the array
time cost is 124969 ms  ,Integer 13 occurs 177 times in the array
time cost is 122873 ms  ,Integer 13 occurs 177 times in the array
time cost is 122819 ms  ,Integer 13 occurs 177 times in the array
time cost is 125530 ms  ,Integer 13 occurs 177 times in the array
time cost is 124825 ms  ,Integer 13 occurs 177 times in the array
average time cost is 124423 ms
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./multiprocess_int.out 99999999 12
test setting: datasize=99999999, process_num=12
time cost is 26675 ms  ,Integer 13 occurs 612 times in the array
time cost is 25586 ms  ,Integer 13 occurs 806 times in the array
time cost is 28929 ms  ,Integer 13 occurs 573 times in the array
time cost is 27496 ms  ,Integer 13 occurs 640 times in the array
time cost is 28601 ms  ,Integer 13 occurs 971 times in the array
time cost is 26723 ms  ,Integer 13 occurs 729 times in the array
time cost is 32035 ms  ,Integer 13 occurs 936 times in the array
time cost is 30029 ms  ,Integer 13 occurs 515 times in the array
time cost is 26460 ms  ,Integer 13 occurs 619 times in the array
time cost is 25500 ms  ,Integer 13 occurs 473 times in the array
average time cost is 27803 ms
```

multi-thread (integer)

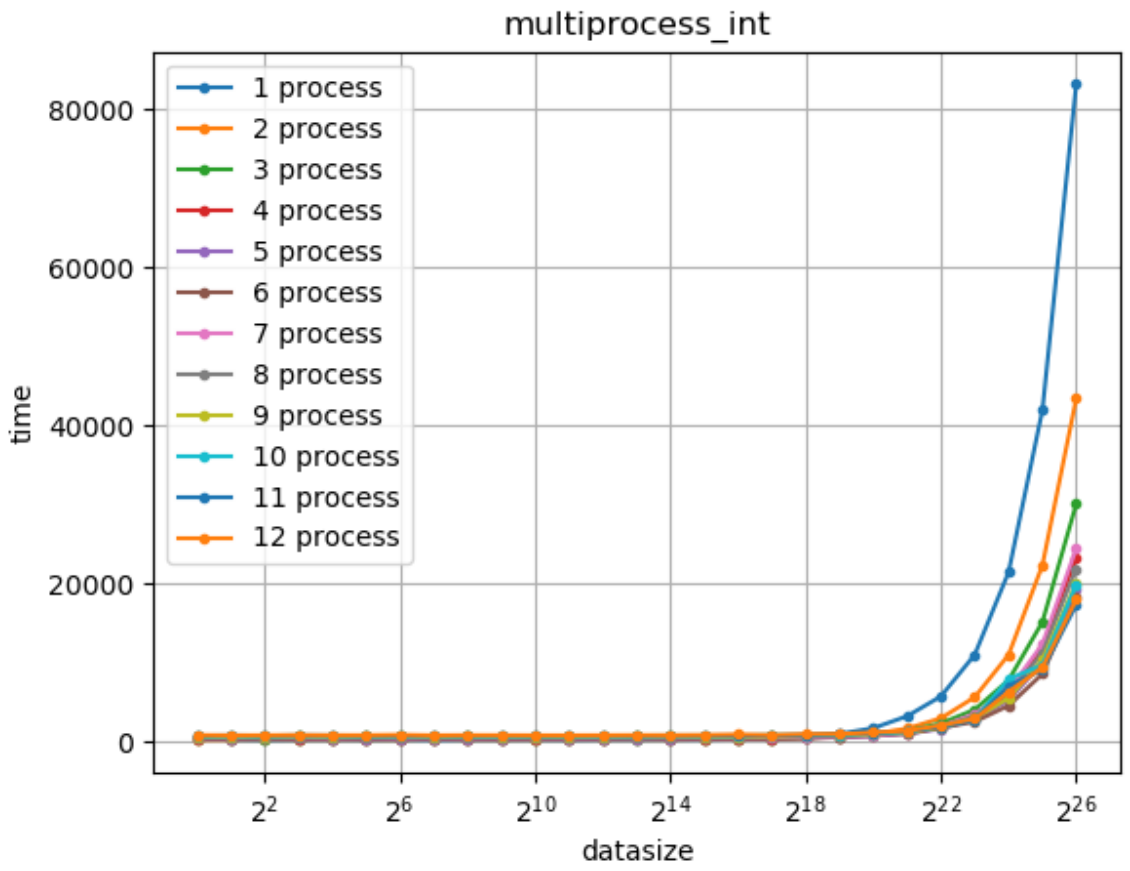
```
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./multithread_int.out 99999999 1
test setting: datasize=99999999, thread_num=1
time cost is 161373 ms  ,Integer 13 occurs 390577 times in the array
time cost is 167021 ms  ,Integer 13 occurs 390577 times in the array
time cost is 163521 ms  ,Integer 13 occurs 390577 times in the array
time cost is 165999 ms  ,Integer 13 occurs 390577 times in the array
time cost is 165961 ms  ,Integer 13 occurs 390577 times in the array
time cost is 166295 ms  ,Integer 13 occurs 390577 times in the array
time cost is 166959 ms  ,Integer 13 occurs 390577 times in the array
time cost is 165683 ms  ,Integer 13 occurs 390577 times in the array
time cost is 165643 ms  ,Integer 13 occurs 390577 times in the array
time cost is 165892 ms  ,Integer 13 occurs 390577 times in the array
average time cost is 165434 ms
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./multithread_int.out 99999999 12
test setting: datasize=99999999, thread_num=12
time cost is 19581 ms  ,Integer 13 occurs 390577 times in the array
time cost is 20308 ms  ,Integer 13 occurs 390577 times in the array
time cost is 24225 ms  ,Integer 13 occurs 390577 times in the array
time cost is 21924 ms  ,Integer 13 occurs 390577 times in the array
time cost is 20401 ms  ,Integer 13 occurs 390577 times in the array
time cost is 27082 ms  ,Integer 13 occurs 390577 times in the array
time cost is 26092 ms  ,Integer 13 occurs 390577 times in the array
time cost is 29986 ms  ,Integer 13 occurs 390577 times in the array
time cost is 22181 ms  ,Integer 13 occurs 390577 times in the array
time cost is 19584 ms  ,Integer 13 occurs 390577 times in the array
average time cost is 23136 ms
```

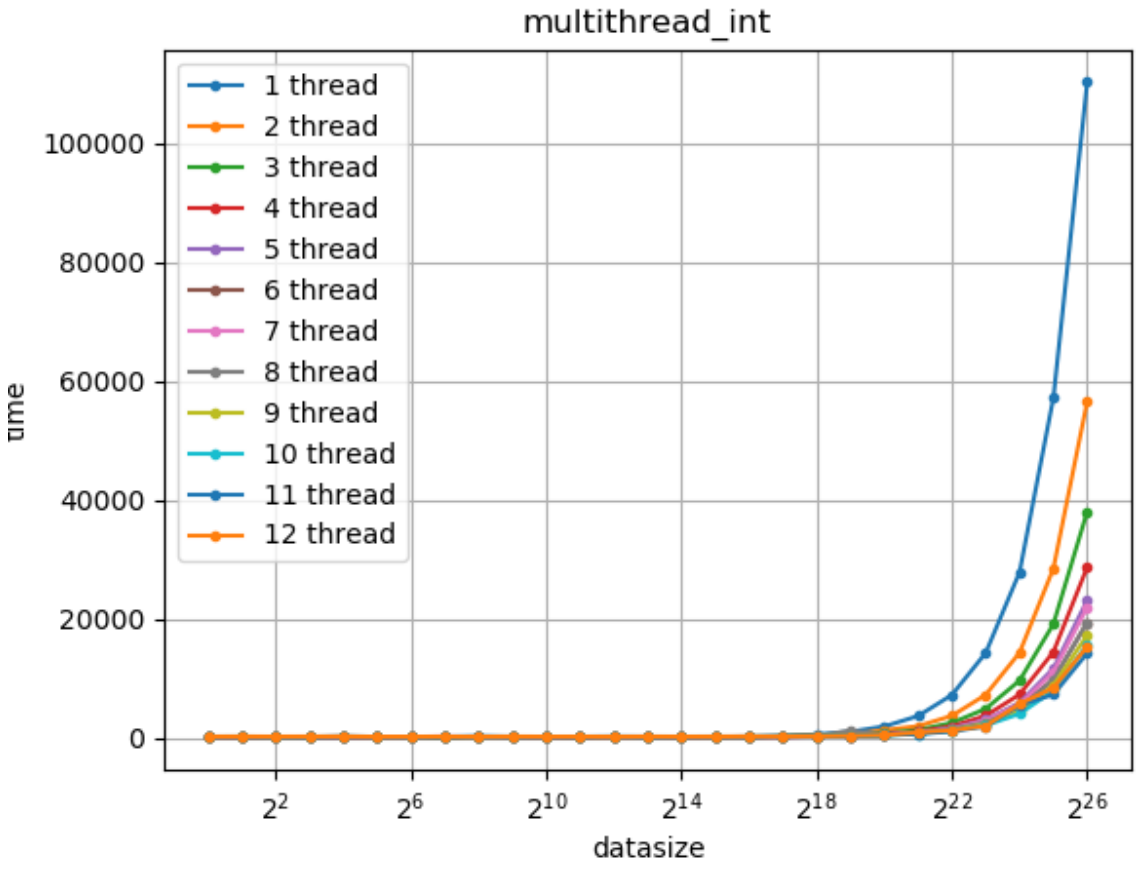
實驗結果

time 單位是 ms，data size是指宣告的array之長度。

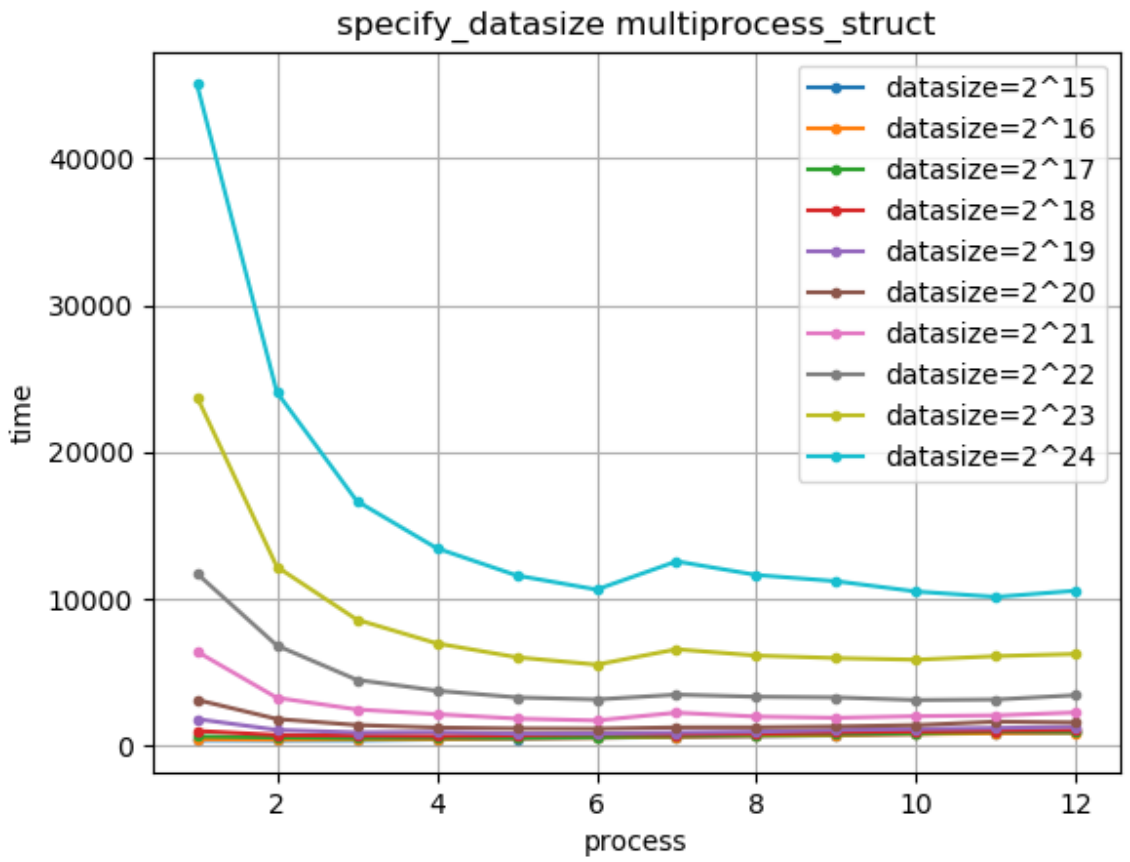
Integer

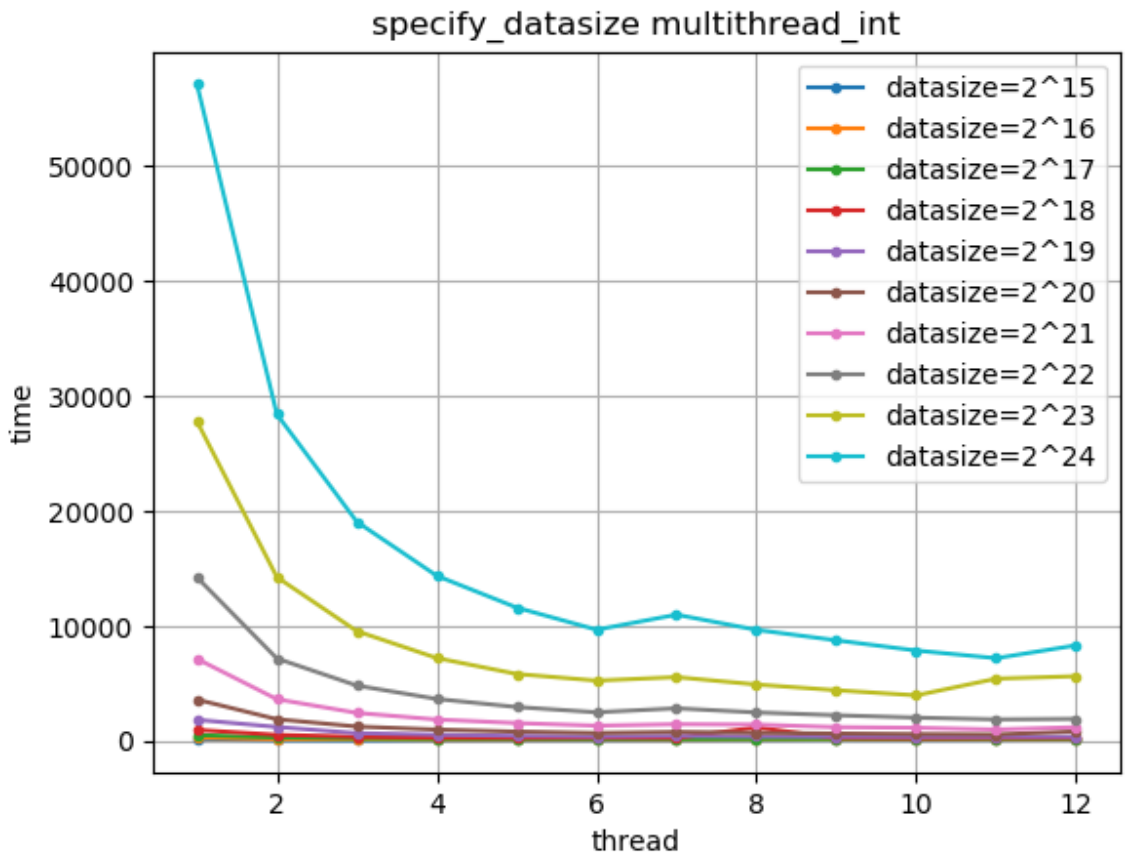
特定 process/thread 數目


multiprocess_int
1 process
2 process
3 process
4 process
5 process
6 process
7 process
8 process
9 process
10 process
11 process
12 process
80000
60000
40000
20000
0
time
2^2
2^6
2^{10}
2^{14}
2^{18}
2^{22}
2^{26}
datasize


multithread_int
1 thread
2 thread
3 thread
4 thread
5 thread
6 thread
7 thread
8 thread
9 thread
10 thread
11 thread
12 thread
100000
80000
60000
40000
20000
0
ume
2^2
2^6
2^{10}
2^{14}
2^{18}
2^{22}
2^{26}
datasize

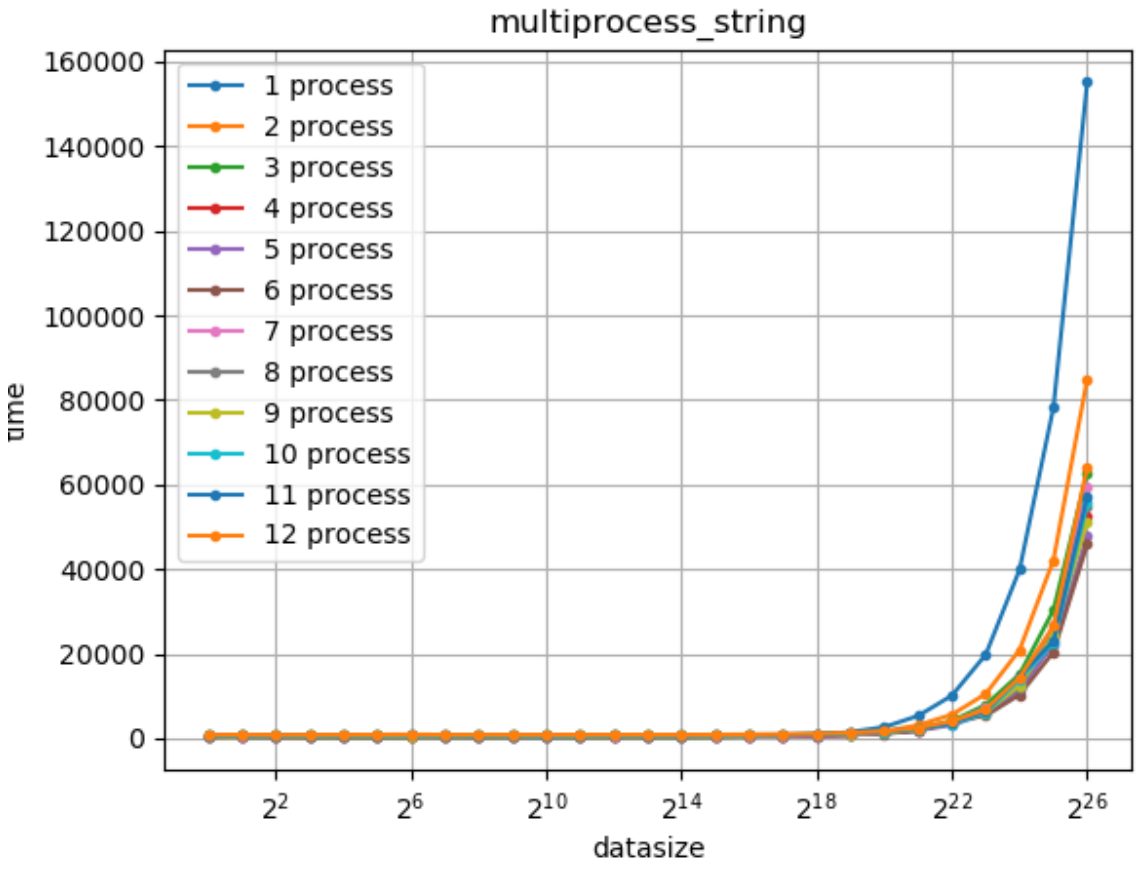
特定 data size

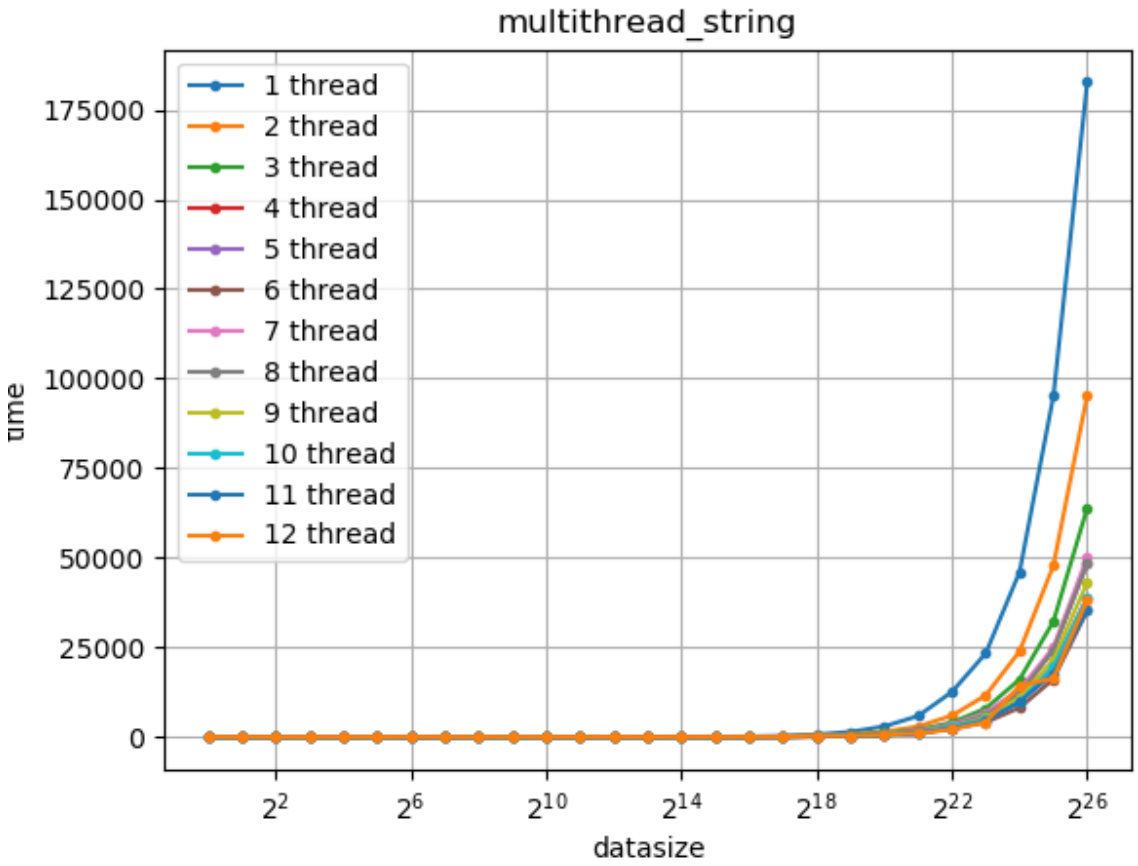

specify_datasize multiprocess_struct
datasize=2^15
datasize=2^16
datasize=2^17
datasize=2^18
datasize=2^19
datasize=2^20
datasize=2^21
datasize=2^22
datasize=2^23
datasize=2^24
time
process
0
10000
20000
30000
40000
2
4
6
8
10
12



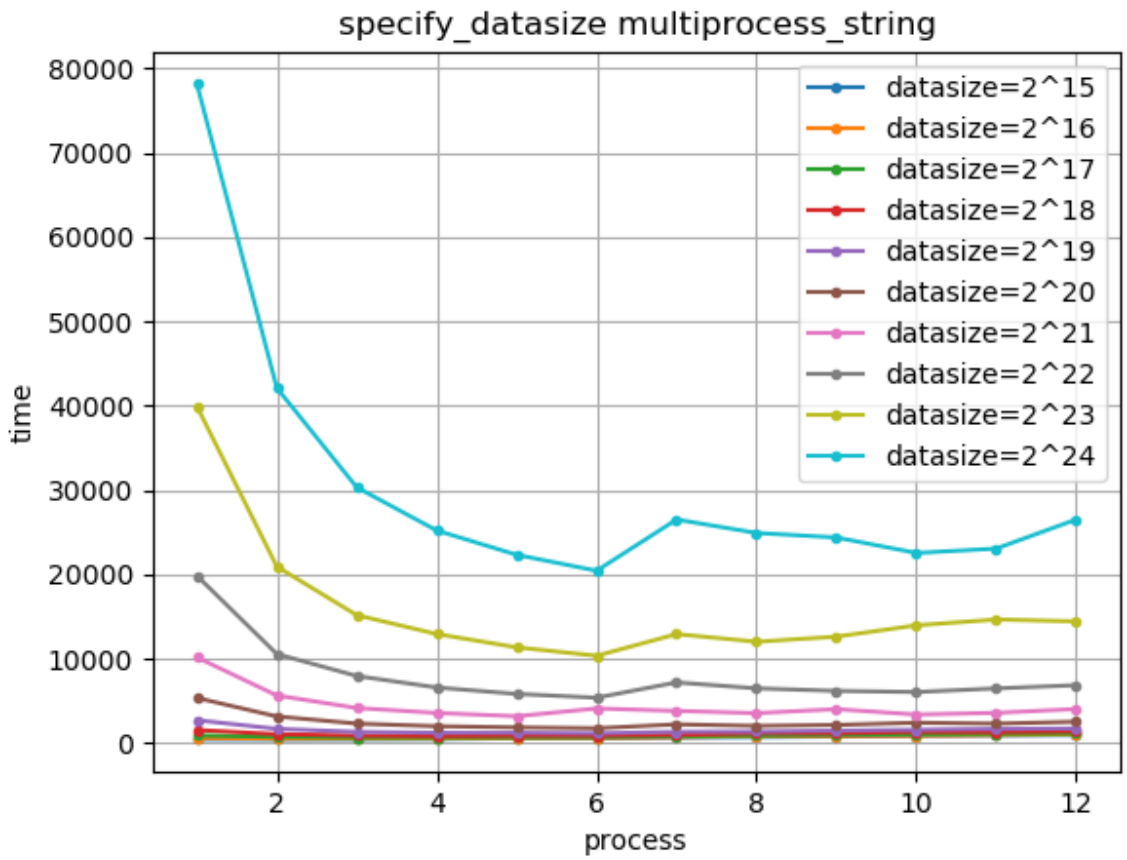
String

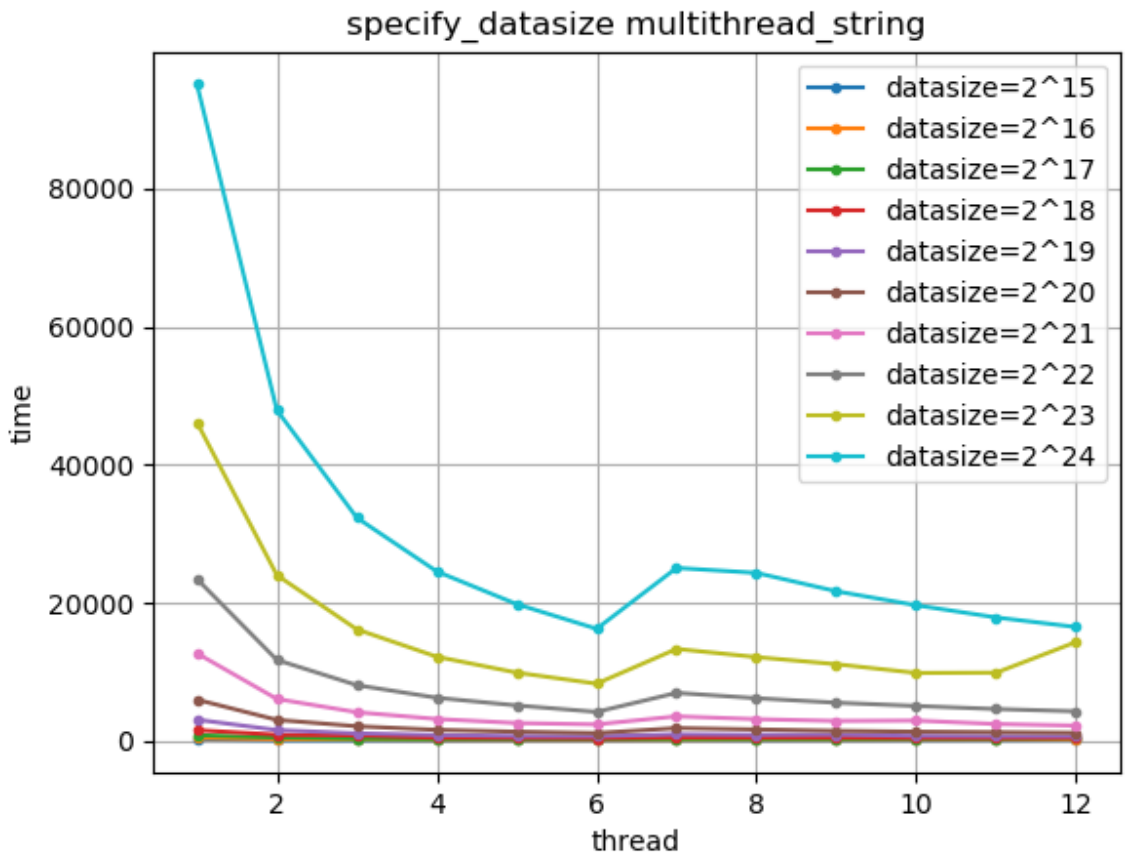
特定 process/thread 數目


multiprocess_string
1 process
2 process
3 process
4 process
5 process
6 process
7 process
8 process
9 process
10 process
11 process
12 process
160000
140000
120000
100000
80000
60000
40000
20000
0
ume
2^2
2^6
2^{10}
2^{14}
2^{18}
2^{22}
2^{26}
datasize


multithread_string
1 thread
2 thread
3 thread
4 thread
5 thread
6 thread
7 thread
8 thread
9 thread
10 thread
11 thread
12 thread
175000
150000
125000
100000
75000
50000
25000
0
ume
2^2
2^6
2^{10}
2^{14}
2^{18}
2^{22}
2^{26}
datasize

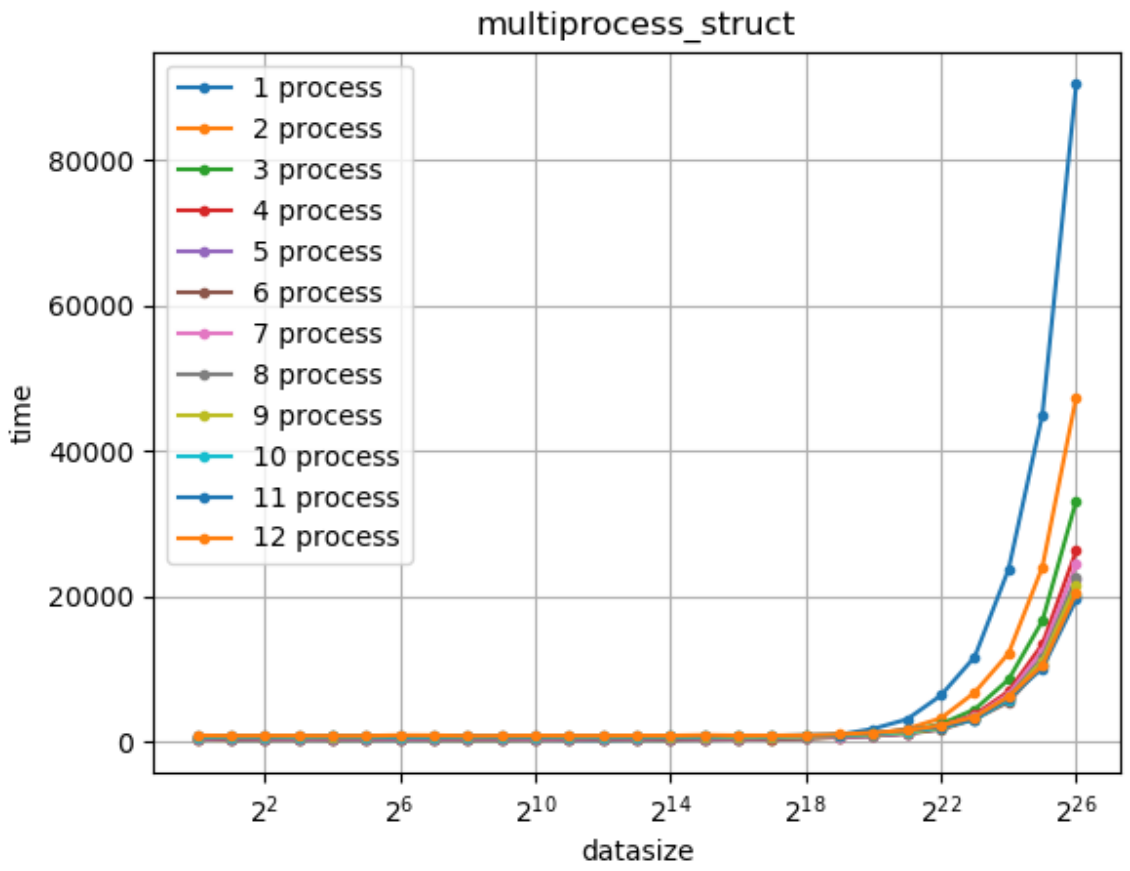
特定 data size

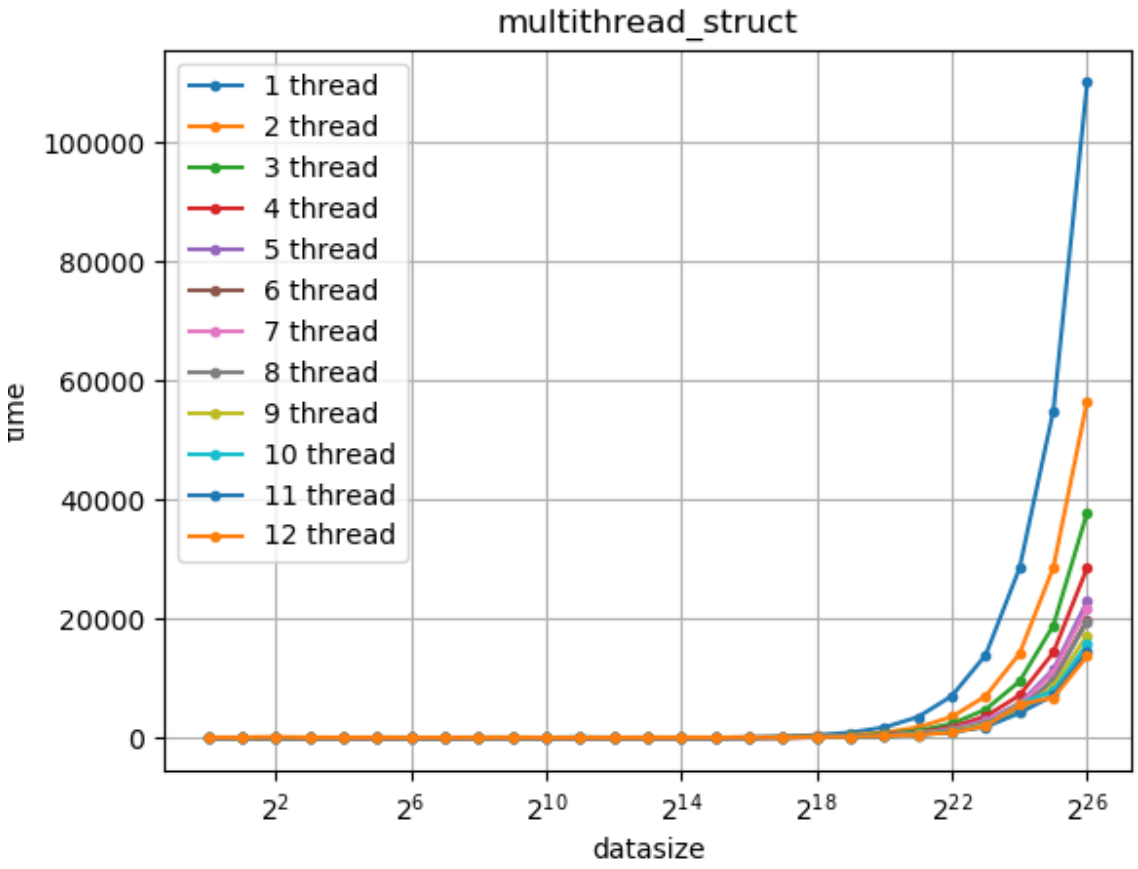

specify_datasize multiprocess_string
time
process
80000
70000
60000
50000
40000
30000
20000
10000
0
2
4
6
8
10
12
datasize=2^15
datasize=2^16
datasize=2^17
datasize=2^18
datasize=2^19
datasize=2^20
datasize=2^21
datasize=2^22
datasize=2^23
datasize=2^24



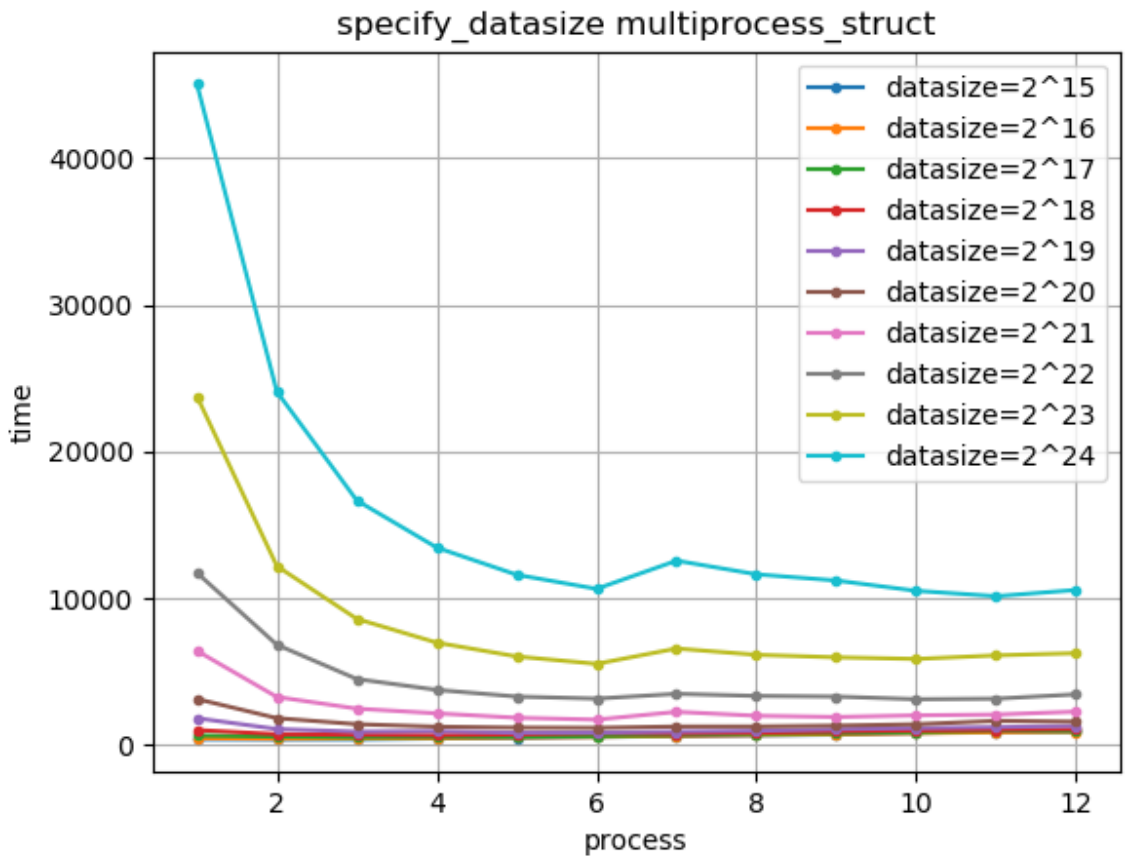
Struct

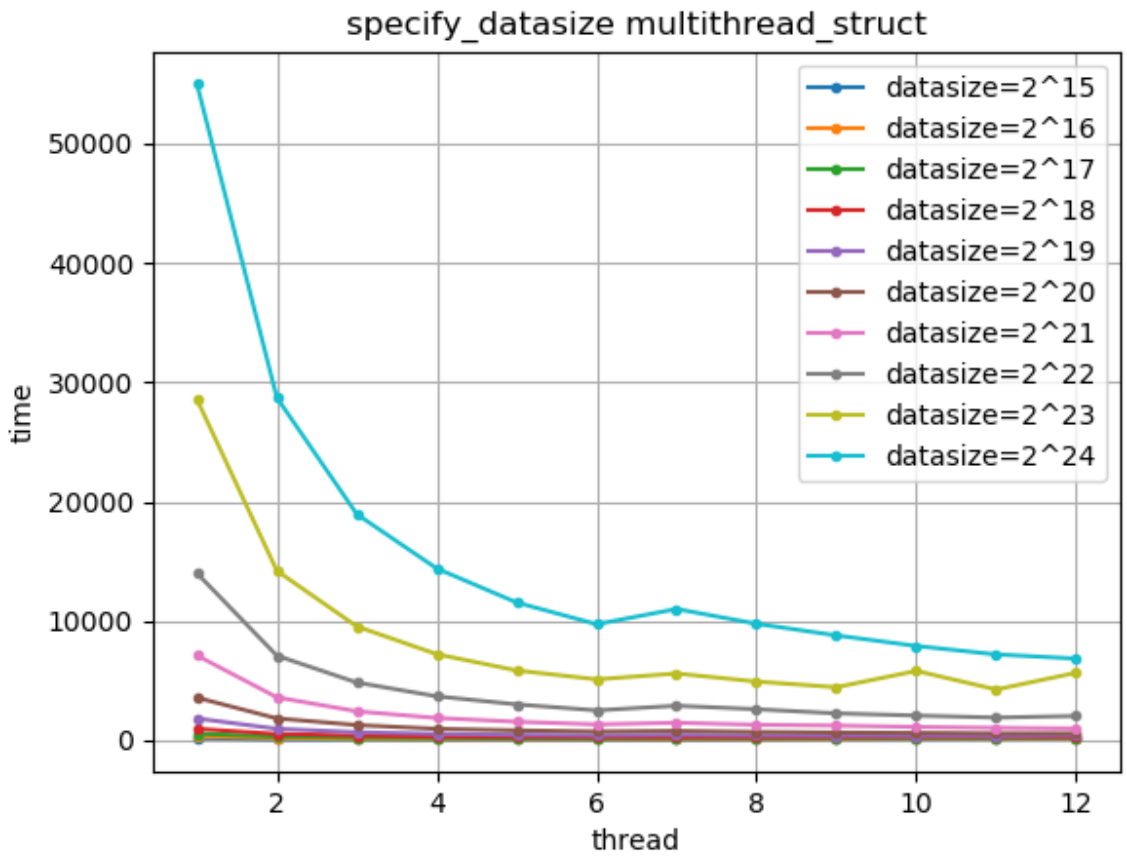
特定 process/thread 數目


multiprocess_struct
1 process
2 process
3 process
4 process
5 process
6 process
7 process
8 process
9 process
10 process
11 process
12 process
time
80000
60000
40000
20000
0
2^2
2^6
2^{10}
2^{14}
2^{18}
2^{22}
2^{26}
datasize


multithread_struct
1 thread
2 thread
3 thread
4 thread
5 thread
6 thread
7 thread
8 thread
9 thread
10 thread
11 thread
12 thread
100000
80000
60000
40000
20000
0
ume
2^2
2^6
2^{10}
2^{14}
2^{18}
2^{22}
2^{26}
datasize

特定 data size


specify_datasize multiprocess_struct
datasize=2^15
datasize=2^16
datasize=2^17
datasize=2^18
datasize=2^19
datasize=2^20
datasize=2^21
datasize=2^22
datasize=2^23
datasize=2^24
time
process
0
10000
20000
30000
40000
2
4
6
8
10
12



Discussion

Mutex 影響執行效能

一個Process底下有數個Thread，而一個Process的Global Variable可以讓它的所有Thread共享，所有Thread都可以存取同一個Process的Global Variable。但是，如果有兩個Thread要存取同一個Global Variable，有可能發生Synchronization問題。解決方法是加入Mutex，將不可以被多個執行緒同時執行的程式碼片段，用互斥鎖包起來，當一個執行緒執行到該處時會先上鎖，避免其他的執行緒進入，若其他的執行緒同時也要執行該處的程式碼時，必須等待先前的執行緒執行完之後，才能接著進入。

比較以下兩段child-thread的寫法和其執行結果

```
void* child(void *arg) {
  u_int64_t *input = (u_int64_t *) arg; // 取得資料
  u_int64_t local_ans = 0;
```

```
  for(u_int64_t j=input[0];j<input[1];j++){
    if (buffer[j]==integer) local_ans++;
  }

  pthread_mutex_lock( &mutex ); // 上鎖
  ans += local_ans;
  pthread_mutex_unlock( &mutex ); // 解鎖
  pthread_exit(NULL);
}
```

```
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./a.out 99999999 12
test setting: datasize=99999999, thread_num=12
time cost is 19551 ms  ,Integer 13 occurs 390577 times in the array
time cost is 29966 ms  ,Integer 13 occurs 390577 times in the array
time cost is 25840 ms  ,Integer 13 occurs 390577 times in the array
time cost is 32522 ms  ,Integer 13 occurs 390577 times in the array
time cost is 28138 ms  ,Integer 13 occurs 390577 times in the array
time cost is 31013 ms  ,Integer 13 occurs 390577 times in the array
time cost is 29806 ms  ,Integer 13 occurs 390577 times in the array
time cost is 30669 ms  ,Integer 13 occurs 390577 times in the array
time cost is 30292 ms  ,Integer 13 occurs 390577 times in the array
time cost is 31753 ms  ,Integer 13 occurs 390577 times in the array
average time cost is 28955 ms
```

```
void* child(void *arg) {
  u_int64_t *input = (u_int64_t *) arg; // 取得資料
  for(u_int64_t j=input[0];j<input[1];j++){
    if (buffer[j]==integer) {
      pthread_mutex_lock( &mutex ); // 上鎖
      ans ++;
      pthread_mutex_unlock( &mutex ); // 解鎖
    }
  }
  pthread_exit(NULL);
}
```

```
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./a.out 99999999 12
test setting: datasize=99999999, thread_num=12
time cost is 66214 ms  ,Integer 13 occurs 390577 times in the array
time cost is 68086 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67213 ms  ,Integer 13 occurs 390577 times in the array
time cost is 66935 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67805 ms  ,Integer 13 occurs 390577 times in the array
time cost is 70076 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67785 ms  ,Integer 13 occurs 390577 times in the array
time cost is 68145 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67129 ms  ,Integer 13 occurs 390577 times in the array
time cost is 68299 ms  ,Integer 13 occurs 390577 times in the array
average time cost is 67768 ms
```

被鎖部份只能以單thread來執行，後者因為上鎖次數較多，執行效率較差。在設計程式時，應盡可能減少被Mutex包住的程式碼，才能讓程式執行效能更好。

執行環境影響

借用實驗室工作站，CPU規格如下

```
lray@argtopWS:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   46 bits physical, 48 bits virtual
CPU(s):                          36
On-line CPU(s) list:             0-35
Thread(s) per core:              2
Core(s) per socket:              18
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           85
Model name:                      Intel(R) Core(TM) i9-10980XE CPU @ 3.00GHz
Stepping:                        7
CPU MHz:                         3000.000
CPU max MHz:                     4800.0000
CPU min MHz:                     1200.0000
BogoMIPS:                        6000.00
Virtualization:                  VT-x
L1d cache:                       576 KiB
L1i cache:                       576 KiB
L2 cache:                        18 MiB
L3 cache:                        24.8 MiB
NUMA node0 CPU(s):               0-35
```

並執行以下實驗，和在自己電腦上的multi-thread實驗結果對照

```
lray@argtopWS:~$ gcc multithread_int.c -lpthread
lray@argtopWS:~$ ./a.out 99999999 12
test setting: datasize=99999999, thread_num=12
time cost is 33584 ms  ,Integer 13 occurs 390577 times in the array
time cost is 33742 ms  ,Integer 13 occurs 390577 times in the array
time cost is 32565 ms  ,Integer 13 occurs 390577 times in the array
time cost is 32358 ms  ,Integer 13 occurs 390577 times in the array
time cost is 40777 ms  ,Integer 13 occurs 390577 times in the array
time cost is 34722 ms  ,Integer 13 occurs 390577 times in the array
time cost is 32925 ms  ,Integer 13 occurs 390577 times in the array
time cost is 36276 ms  ,Integer 13 occurs 390577 times in the array
time cost is 34056 ms  ,Integer 13 occurs 390577 times in the array
time cost is 39812 ms  ,Integer 13 occurs 390577 times in the array
average time cost is 35081 ms
lray@argtopWS:~$ ./a.out 99999999 36
test setting: datasize=99999999, thread_num=36
time cost is 21242 ms  ,Integer 13 occurs 390577 times in the array
time cost is 19385 ms  ,Integer 13 occurs 390577 times in the array
time cost is 12413 ms  ,Integer 13 occurs 390577 times in the array
time cost is 10277 ms  ,Integer 13 occurs 390577 times in the array
time cost is 8372 ms  ,Integer 13 occurs 390577 times in the array
time cost is 7597 ms  ,Integer 13 occurs 390577 times in the array
time cost is 8469 ms  ,Integer 13 occurs 390577 times in the array
time cost is 7628 ms  ,Integer 13 occurs 390577 times in the array
time cost is 7607 ms  ,Integer 13 occurs 390577 times in the array
time cost is 8814 ms  ,Integer 13 occurs 390577 times in the array
average time cost is 11180 ms
```

```
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ gcc multithread_int.c -lpthread
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./a.out 99999999 12
test setting: datasize=99999999, thread_num=12
time cost is 66214 ms  ,Integer 13 occurs 390577 times in the array
time cost is 68086 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67213 ms  ,Integer 13 occurs 390577 times in the array
time cost is 66935 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67805 ms  ,Integer 13 occurs 390577 times in the array
time cost is 70076 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67785 ms  ,Integer 13 occurs 390577 times in the array
time cost is 68145 ms  ,Integer 13 occurs 390577 times in the array
time cost is 67129 ms  ,Integer 13 occurs 390577 times in the array
time cost is 68299 ms  ,Integer 13 occurs 390577 times in the array
average time cost is 67768 ms
```

和在自己電腦上的multi-process實驗結果對照

```
lray@argtopWS:~$ gcc multiprocess_int.c
lray@argtopWS:~$ ./a.out 99999999 12
test setting: datasize=99999999, process_num=12
time cost is 37688 ms  ,Integer 13 occurs 566 times in the array
time cost is 41918 ms  ,Integer 13 occurs 668 times in the array
time cost is 40350 ms  ,Integer 13 occurs 867 times in the array
time cost is 47371 ms  ,Integer 13 occurs 993 times in the array
time cost is 43458 ms  ,Integer 13 occurs 573 times in the array
time cost is 43542 ms  ,Integer 13 occurs 796 times in the array
time cost is 44293 ms  ,Integer 13 occurs 503 times in the array
time cost is 41967 ms  ,Integer 13 occurs 804 times in the array
time cost is 37835 ms  ,Integer 13 occurs 929 times in the array
time cost is 41400 ms  ,Integer 13 occurs 914 times in the array
average time cost is 41982 ms
lray@argtopWS:~$ ./a.out 99999999 36
test setting: datasize=99999999, process_num=36
time cost is 31654 ms  ,Integer 13 occurs 2425 times in the array
time cost is 32764 ms  ,Integer 13 occurs 1970 times in the array
time cost is 34072 ms  ,Integer 13 occurs 2067 times in the array
time cost is 34800 ms  ,Integer 13 occurs 2125 times in the array
time cost is 45331 ms  ,Integer 13 occurs 2116 times in the array
time cost is 33082 ms  ,Integer 13 occurs 2421 times in the array
time cost is 32460 ms  ,Integer 13 occurs 2333 times in the array
time cost is 45453 ms  ,Integer 13 occurs 2112 times in the array
time cost is 33522 ms  ,Integer 13 occurs 2333 times in the array
time cost is 40972 ms  ,Integer 13 occurs 2291 times in the array
average time cost is 36411 ms
```

```
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ gcc multiprocess_int.c
yellow@yellow-arg7:~/nctu-operating-system-course/hw1$ ./a.out 99999999 12
test setting: datasize=99999999, process_num=12
time cost is 25571 ms  ,Integer 13 occurs 882 times in the array
time cost is 35447 ms  ,Integer 13 occurs 580 times in the array
time cost is 32055 ms  ,Integer 13 occurs 963 times in the array
time cost is 33039 ms  ,Integer 13 occurs 577 times in the array
time cost is 31437 ms  ,Integer 13 occurs 671 times in the array
time cost is 27434 ms  ,Integer 13 occurs 532 times in the array
time cost is 34655 ms  ,Integer 13 occurs 716 times in the array
time cost is 32492 ms  ,Integer 13 occurs 786 times in the array
time cost is 32432 ms  ,Integer 13 occurs 531 times in the array
time cost is 30505 ms  ,Integer 13 occurs 763 times in the array
average time cost is 31506 ms
```

執行multi-process時，一個CPU一次只能做一件事情，但process數大於CPU的數量，因此每個process使用的時間需要被排程。對此擁有18核的電腦並沒有比較好的性能表現。

但在執行multi-thread時，可以在同一個process底下，開啟更多thread同時工作，對此擁有18核36threads的電腦性能上就有顯著的優勢。而process又會根據thread的優先權以及

已經用掉的CPU時間，在不同的thread作切換，以讓各個thread都有機會執行，因此，同樣開12multi-thread時，仍然是18核電腦表現較佳。

multi-process 和 multi-thread比較

由實驗結果的圖表來做比較，特定 process/thread 數目時，分析圖表的最右邊，藍線(1 process/thread)看來 multi-process需要的時間較短。但橘線(12 process/thread)看來結果相反。

而特定 datasize 時，往圖表的最左上角看來multi-process較佳，但往圖表的右下角看來，則multi-thread較佳。

multi-process增加了process數目，但當CPU數目遠不足，開啟多個process依然只能放著等待，而multi-thread能夠利用這些閒置時間，降低 CPU 閒置的機會，故能提高計算效能。

但如果CPU處於都不需要等待的狀態，使用multi-thread反而額外增添資源同步的協調動作，導致效能降低。 反觀multi-process，由於不需額外的資源同步協調動作，故CPU反而能全速運轉，使其效能隨計算核心數量提高。