

# HW1 Multitasking

0616098 黃秉茂

Python version: Python3.7

```
maomao@pandaman:~$ python --version
Python 3.7.6
```

Packages:

```
import time
import threading
import queue
import random
import string
import hashlib
import requests
import multiprocessing
from lxml.html import fromstring
import asyncio
```

# 1. The effect of the number of threads on performance:

## Task A

1 thread: 540 seconds

2 threads: 582 seconds

4 threads: 654 seconds

The taskbar at the bottom of the screen displays several pinned icons, including File Explorer, OneDrive, Mail, Photos, OneNote, Microsoft Edge, and others. On the right side, there are icons for battery level, signal strength, and volume, along with the date and time (下午 07:59, 2021/1/3).

100 threads: 899 seconds

## Task B

1 thread: 62 seconds



2 threads: 21 seconds

4 threads: 11 seconds

# 100 threads: 9 seconds

```
maomao@pandaman: ~
lsaf.py examples.desktop ML_HW4_SWM_and_ANN.ipynb test.json
a.py hw1.py ML_HW4_SWM_and_ANN.ipynb train.json
b.py input1.txt rr.py y_test.csv
c.py input2.txt test
maomao@pandaman: ~$ python hw1.py < input2.txt
maomao@pandaman: ~$ Tasks: 105 240 thr: 1 running
Load average: 0.00 0.09 0.43
Uptime: 42 days, 08:28:52
[ 100%] [=====] 847M/62.8G Tasks: 105 240 thr: 1 running
Load average: 0.00 0.09 0.43
Uptime: 42 days, 08:28:52
maomao@pandaman: ~
maomao@pandaman: ~ ls
maomao@pandaman: ~ head input2.txt
2
100
100
https://health.udn.com/health/story/6037/4949284
https://health.udn.com/health/story/7426/4932266
https://health.udn.com/disease/sole/63995
https://health.udn.com/health/cat/10694
https://health.udn.com/health/cat/120949
https://health.udn.com/health/calorie
https://health.udn.com/health/story/5967/4946441
maomao@pandaman: ~$ Tasks: 89 194 thr: 1 running
Load average: 0.24 0.18 0.33
Uptime: 42 days, 10:46:00
[ 100%] [=====] 506M/62.8G Tasks: 89 194 thr: 1 running
Load average: 0.24 0.18 0.33
Uptime: 42 days, 10:46:00
maomao@pandaman: ~
maomao@pandaman: ~
maomao@pandaman: ~ ls
maomao@pandaman: ~ head input2.txt
2
100
100
https://health.udn.com/health/story/6037/4949284
https://health.udn.com/health/story/7426/4932266
https://health.udn.com/disease/sole/63995
https://health.udn.com/health/cat/10694
https://health.udn.com/health/cat/120949
https://health.udn.com/health/calorie
https://health.udn.com/health/story/5967/4946441
maomao@pandaman: ~$ Tasks: 89 194 thr: 1 running
Load average: 0.24 0.18 0.33
Uptime: 42 days, 10:46:00
[ 100%] [=====] 506M/62.8G Tasks: 89 194 thr: 1 running
Load average: 0.24 0.18 0.33
Uptime: 42 days, 10:46:00
maomao@pandaman: ~
```

Task / # threads	1	2	4	100
A	540	582	654	899
B	62	21	11	9

(unit: seconds)

可以發現 Task B 的速度都比 Task A 快，因為 hash 找值和計算都要時間

Task A 越多 threads 反而最久，因為 Task A 要花很久的時間執行，而且用再多 thread 使用 cpu 的總量也不會差太多。工作邏輯簡單的其實適合少執行緒，太多執行緒反而會因為一直 context switch 而浪費時間

Task B 是越多 thread 越快，畢竟平行化且每個 thread 執行都很快，單個執行緒的工作邏輯複雜、等待時間較長或者需要消耗大量系統運算資源的像從遠端服務獲得資料並計算適合多執行緒

## 2. The effect of the number of processes on performance:

## Task A

1 processes: 470 seconds

2 processes: 252 seconds

4 processes: 127 seconds

```
maomao@pandaman: ~
13:49:07 [22/01/2021]
maomao@pandaman: ~
[maomao@pandaman ~]$ top
top - 13:49:07 up 42 days, 08:48:16, 1 user, load average: 3.13 2.00 1.24
Tasks: 102, 240 total; 1 running, 849M (62.80) free; 129M (2.000) used by processes
CPU states: 0.0% user, 0.0% nice, 0.0% system, 0.0% idle, 0.0% iowait, 0.0% softirq, 0.0% steal, 0.0% guest, 0.0% guest_nice
CPU usage by task:
  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM    TIME+  Command
 13796 maomao  20  0 4284 20752 1264 S  0.7  0.0  4:06:58 mvtop
 13497 maomao  20  0 34012 5120 3988 R  0.7  0.0  0:04:59 httpd
  4048 maomao  20  0 1260M 1309 3268 S  0.0  0.2  2:25:39 /home/maomao/.vscode-server/bin/eaa3859d4ba21
 13795 maomao  20  0 1260M 1309 3268 S  0.0  0.2  2:25:39 /home/maomao/.vscode-server/bin/eaa3859d4ba21
  933 root     20  0 51568 3916 3268 S  0.0  0.0  0:34:50.76 /usr/bin/dbus-daemon --system --address=system
 1758 root     20  0 1501M 15612 1144 S  0.0  0.0  0:35:52.13 /usr/sbin/cups-browsed
 1551 gdm    20  0 649M 25224 6768 S  0.0  0.0  56:51.9 /usr/lib/gnome-settings-daemon/gsd-color
 4059 maomao  20  0 890M 61065 2648 S  0.0  0.1  0:02:57 /home/maomao/.vscode-server/bin/eaa3859d4ba21
 3994 mailman 20  0 1260M 1309 3268 S  0.0  0.0  0:00:26 /usr/lib/mailman/bin/postfix
 3504 mailman 20  0 47784 924 1448 S  0.0  0.0  23:05:51.06 smbd: mailman@off
 18968 maomao  20  0 33906 5956 2408 S  0.0  0.0  0:00:00.00 smbd: mailman@off: running [pandaman.local]
 29602 root     20  0 3950M 27552 15708 S  0.0  0.0  7:09:34 taux
 4051 maomao  20  0 1260M 1309 3268 S  0.0  0.2  0:02:30 /home/maomao/.vscode-server/bin/eaa3859d4ba21
 1471 gdm    20  0 4610M 6234 36168 S  0.0  0.0  10:57:13 /usr/lib/gnome-shell
 3991 root     20  0 1260M 1309 3268 S  0.0  0.0  0:00:22.36 /usr/sbin/lightdm --foreground
 11386 maomao  20  0 107M 3483 2468 S  0.0  0.0  0:00:00.60 smbd: mailman@7
 4050 maomao  20  0 1260M 1309 3268 S  0.0  0.2  0:02:30 /home/maomao/.vscode-server/bin/eaa3859d4ba21
[maomao@pandaman ~]$ cat input.txt
maomao@pandaman:~$ head input.txt
1
2 4
100
nY_OZ
M19aY
Unmap
[2zL0
SFxO*
-qL>
kAC>
maomao@pandaman:~$
```

100 processes: 59 seconds

Editor: Visual Studio Code | Language: Python | Version: 2021.3 | Date: 2021/3/21 | Time: 08:55 | Status: Running

# Task B

1 processes: 40 seconds

```
maomao@pandaman:~ ls
sat.py examples/test_ML_HW4_SWM_and_ANN3.ipynb test.json
ip.py input1.txt trc.py SYM_and ANN.ipynb train.json
b.py input2.txt test
c.py
maomao@pandaman:~$ python hwl.py < input2.txt
不只有自杀低落想自杀 痴呆專家：花椰菜等食物不明益处也要小心！ 聰明飲食！ 養生
H:\PycharmProjects\pandaman> python hwl.py
[...]
maomao@pandaman:~$ head input2.txt
2.1
100
https://health.udn.com/health/story/6037/4940284
https://health.udn.com/health/story/7424/932266
https://health.udn.com/disease/9395
https://health.udn.com/health/cate/10691
https://health.udn.com/health/calorie
https://health.udn.com/health/story/5967/4946441
maomao@pandaman:~$
```

```
maomao@pandaman:~ ls
sat.py examples/test_ML_HW4_SWM_and_ANN3.ipynb test.json
ip.py input1.txt trc.py SYM_and ANN.ipynb train.json
b.py input2.txt test
c.py
maomao@pandaman:~$ python hwl.py < input2.txt
不只有自杀低落想自杀 痴呆專家：花椰菜等食物不明益处也要小心！ 聰明飲食！ 養生
H:\PycharmProjects\pandaman> python hwl.py
[...]
maomao@pandaman:~$ head input2.txt
2.1
100
https://health.udn.com/health/story/6037/4940284
https://health.udn.com/health/story/7424/932266
https://health.udn.com/disease/9395
https://health.udn.com/health/cate/10691
https://health.udn.com/health/calorie
https://health.udn.com/health/story/5967/4946441
maomao@pandaman:~$
```

2 processes: 17 seconds

```
maomao@pandaman:~
```

```
maomao@pandaman:~ ls
naf.py examples.desktop ML_HW4_SW_and_ANNN3.ipynb test.json
a.py hw1.py ML_HW4_SW_and_ANNN.ipynb train.json
b.py input1.txt rr.py y_test.csv
c.py input2.txt
```

```
maomao@pandaman:~ python hw1.py < input2.txt
```

```
臺灣茶人研究會茶農因疫找不到機心就走了！研究找出可重複機心的方式！潘懷宗！  
名人！元氣網  
不只有茶山頭有茶 蕃薯素中藏 毒物專家：蕃薯葉等食物不明沾染也要小心！  
聯合新聞！養生
```

```
1 [ 0.0%] 7 [ 0.0%] 13 [ 0.7%] 19 [ 0.0%]
2 [ 0.0%] 8 [ 0.0%] 14 [ 0.0%] 20 [ 0.0%]
3 [ 0.0%] 9 [ 0.0%] 15 [ 0.0%] 21 [ 0.0%]
4 [ 0.0%] 10 [ 0.0%] 16 [ 0.0%] 22 [ 5.3%]
5 [ 0.0%] 11 [ 0.0%] 17 [ 0.0%] 23 [ 0.0%]
6 [ 0.0%] 12 [ 0.0%] 18 [ 0.0%] 24 [ 0.0%]
[ 11.24] 884M/62.85G Tasks: 105, 240 thr: 1 running
Mem: [ 11.24] 129M/2.00G Load average: 0.81 3.95 2.64
Swap: [ 11.24] 129M/0.00G Uptime: 42 days, 08:53:07
```

PROGRESS	FPS	RT	YEST	SPS	PERC	TIME	MEM	SWAP
23049 naomao	20	0	175M 33540	1/644	10.6	0.1	0:00:28	python hw1.py
23050 naomao	20	0	175M 33540	10/644	6.6	0.1	0:00:27	python hw1.py
4048 naomao	20	0	1267M 1238	3/2688	8.3	0.2	2:29:49	/home/maomao/.vscode-server/bin/eaa389d4ba2f
13795 naomao	20	0	412M 53924	2/2688	8.3	0.2	2:49:35	http://
13796 naomao	20	0	4482M 2025	2/2688	8.3	0.2	2:49:35	http://
15407 naomao	20	0	3412M 120	9/988	8.7	0.0	0:03:46	9 http://
714 systemd-r	20	0	70788 300	0	0.7	0.0	0:11:23	/lib/systemd/systemd-resolved
23047 naomao	20	0	380M 36224	1/908	8.0	0.0	0:00:09	python hw1.py
1551 gdm	20	0	6439M 23224	7/688	8.0	0.0	56:52:48	/usr/lib/gnome-settings-dammon/gsd-color
939 messagebus	20	0	5168M 23224	7/688	8.0	0.0	34:30:97	/usr/bin/bus-daemon -system --address=system
939 messagebus	20	0	4484M 23224	7/688	8.0	0.0	34:30:97	/usr/bin/bus-daemon -runrand [pandaman.local]
18968 naomao	20	0	33096 5956	408/S 0.0	0.0	0:07:10	10.taux	
11886 naomao	20	0	107M 3488	4/688	8.0	0.0	0:00:65	sshd: maomaopts/7
4051 naomao	20	0	1267M 1238	3/2688	8.0	0.0	0:02:34	/home/maomao/.vscode-server/bin/eaa389d4ba2f
23051 root	20	0	271M 17680	1/688	8.0	0.0	0:00:46	/usr/lib/udev/usbmount
12365 naomao	20	0	5/940 108	0 S 0.0	0.0	1:38:18	/usr/sbin/mount.loops	

4 processes: 9 seconds

100 processes: 3 seconds

The screenshot shows a terminal window with several tabs open. The current tab displays a list of processes and their details, including memory usage, CPU time, and file paths. Other tabs show system logs, disk usage, and network information. The desktop environment includes a dock with various application icons.

Task / # processes	1	2	4	100
A	470	252	127	59
B	40	17	9	3

(unit: seconds)

可以發現 Task B 的速度都比 Task A 快，因為 hash 找值和計算都要時間

Task A 是越多 processes 越快，畢竟平行化

Task B 是越多 processes 越快，畢竟平行化

都是越多 processes 越快，因為越多 process 通常代表用的 cpu 較多，100 processes 為 cpu 全用

### 3. The performance comparison of multithreading, multiprocess, and coroutine

## Task A

single thread: 540 seconds

```
naoao@pandaman:~
```

```
examples.desktop input1.txt FF.PY
```

```
hw1.py hw1.py ME_HW4_SWY_and_ANNN.ipynb test
```

```
naoao@pandaman:~$ python hw1.py < input1.txt
```

```
T[1-N,OZ]
```

```
C[1-OH]
```

```
L[1-L]
```

```
J[1-J]
```

```
I[1-I]
```

```
B[1-B]
```

```
E[1-E]
```

```
R[1-R]
```

```
S[1-S]
```

```
W[1-W]
```

```
F[1-F]
```

```
G[1-G]
```

```
H[1-H]
```

```
D[1-D]
```

```
M[1-M]
```

```
P[1-P]
```

```
Z[1-Z]
```

```
naoao@pandaman:~$
```

```
ps aux | grep -v "grep" | grep -v "python" | grep -v "head"
```

```
naoao@pandaman:~$ head input1.txt
```

100 threads: 899 seconds

```
maonao@panzhanan:~  
macosx@panzhanan:~$ ls  
ai.py examples.desktop ML_HW4_SWM_and_ANNN.ipynb test.json  
hw1.py ML_HW4_SWM_and_ANNN.ipynb train.json  
h.py input1.txt rr.py y_test.csv  
input2.txt  
maonao@panzhanan:~$ python hw1.py < input1.txt  
Key [██████████] 872K/62,80T Tasks: 107, 320 thr: 2 running  
Step [██] 129W/2,00G Load average: 1.42, 1.12, 0.97  
Uptime: 42 days 08:05:46
```

11 | 0 python<sup>4</sup> panduan | 26 | 03 - 21 下午 08:11 2021/1/3 25

```
macao@pandamon ~
```

100 processes: 59 seconds

coroutines: 431 seconds

## Task B

single thread: 62 seconds

100 threads: 9 seconds

100 processes: 3 seconds

The screenshot shows a terminal window with several tabs open. The main tab displays a list of processes with columns for PID, PPID, NI, VIRT, RES, SHR, %CPU, %MEM, TIME+, and Command. A large number of processes are listed, many of which are named 'python hwl.py'. Other visible processes include 'taxd', 'ntpd', 'rsync', and various 'curl' and 'wget' tasks. The bottom of the terminal shows a command history with entries related to file operations like 'head', 'tail', and 'grep'. The desktop environment includes a dock with icons for various applications like a browser, file manager, and terminal.

coroutines: 41 seconds

Task / method	single thread	100 threads	100 processes	coroutines
A	540	899	59	431
B	62	9	3	41

(unit: seconds)

可以發現 Task B 的速度都比 Task A 快，因為 hash 找值和計算都要時間

Task A 是 100 processes > coroutines > single threads > 100 thread，Task A 要花很久的時間執行，工作邏輯簡單的其實適合少執行緒，太多執行緒反而會因為一直 context switch 而浪費時間，而多 process 因為用的 cpu 較多通常會較快。交換 Coroutine 不會牽扯到作業系統，是由使用者程式端在控制。所以交換 Coroutine 比交換 thread 快。

Task B 是 100 processes > 100 threads > coroutines > single thread 越多功越快，畢竟平行化且單個執行緒的工作邏輯複雜、等待時間較長或者需要消耗大量系統運算資源的像從遠端服務獲

得資料並計算適合多功，而多 process 因為用的 cpu 較多通常會較快。交換 Coroutine 不會牽扯到作業系統，是由使用者程式端在控制。所以交換 Coroutine 比交換 thread 快。