

HW5 pthread 的建立和等待

系級：資工二 學號：409410114 姓名：周述君

1. 使用 `time` 函數得到的「運算時間各為多少」，例如：`real`、`user`、`sys` 各為多少。並說明 `real`、`user`、`sys` 的意義

```
sujean@vm:~/oshw/hw5$ time ./pi 7
num_CPU 2
pi = 3.1415927

real    0m1.693s
user    0m3.162s
sys     0m0.039s
sujean@vm:~/oshw/hw5$ time ./pi 6
num_CPU 2
pi = 3.141593

real    0m1.745s
user    0m3.290s
sys     0m0.017s
sujean@vm:~/oshw/hw5$ time ./pi 5
num_CPU 2
pi = 3.14159

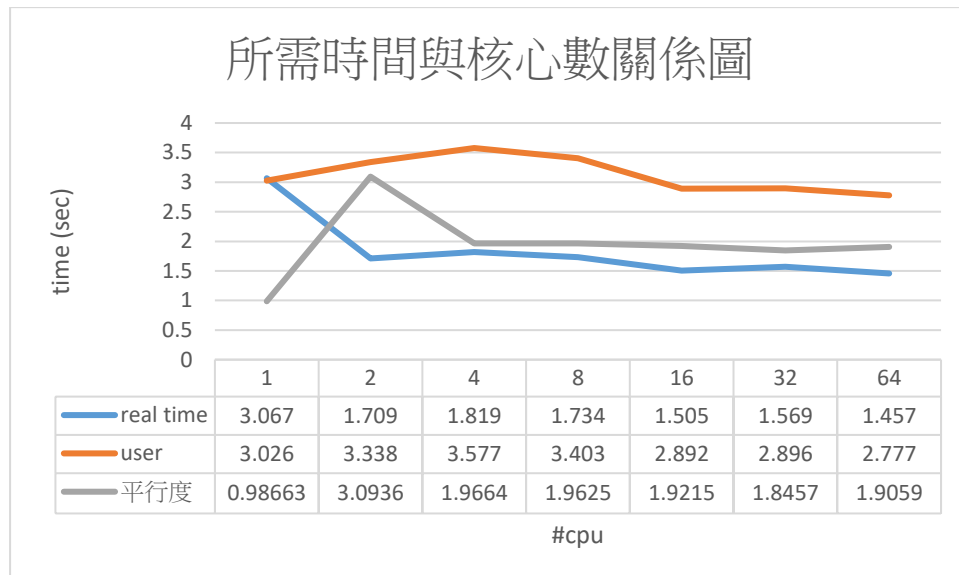
real    0m1.616s
user    0m3.057s
sys     0m0.012s
sujean@vm:~/oshw/hw5$ time ./pi 4
num_CPU 2
pi = 3.1416

real    0m2.112s
user    0m3.877s
sys     0m0.028s
```

- `real time`：程式執行實際所花費的時間加總。
- `user time`：在 `user mode` 下 CPU time 總和，多核 CPU 計算時就要把每顆 CPU 所花費的時間加起來。
- `sys time`：在 `kernel space` 下 CPU time 總和。

上圖是在不同 CPU 數量下執行程式所得到的運算時間，`real time` ≈ 1.7 ，`user time` ≈ 3.4 ，`sys time` ≈ 0.025 。

2. 如果你的程式可以指定不同的核心數量，請說明在同樣的精準度下，你的程式是否可以得到線性的加速。例如：畫圖，橫軸為 **core** 數量，縱軸為所需時間（這一題的目的是讓大家了解怎麼樣說明所撰寫的程式的「平行化」是否良好）



上圖是在精確度 7 下指定不同核心數所得的時間，可看到 **real time** 的執行時間隨著核心增加而減少，平行化效能只在剛增加核心數時有顯著增加，但過 4 核之後就趨緩了。

3. 請說明你是否使用特別的方法加速你的運算？例如：每次運算可以基於已知的結果繼續往下算。

在計算 **lower bound** 時利用 **upper bound** 往右平移，再加上一小塊得出 **lower bound**，這樣就不用用計算 **upper bound** 的方式在算一次了。

```
up = ((long double)total)/(loopCount);  
low = up - ((long double)1.0/loopCount) + ((long double)1.0/loopCount)*z;
```