

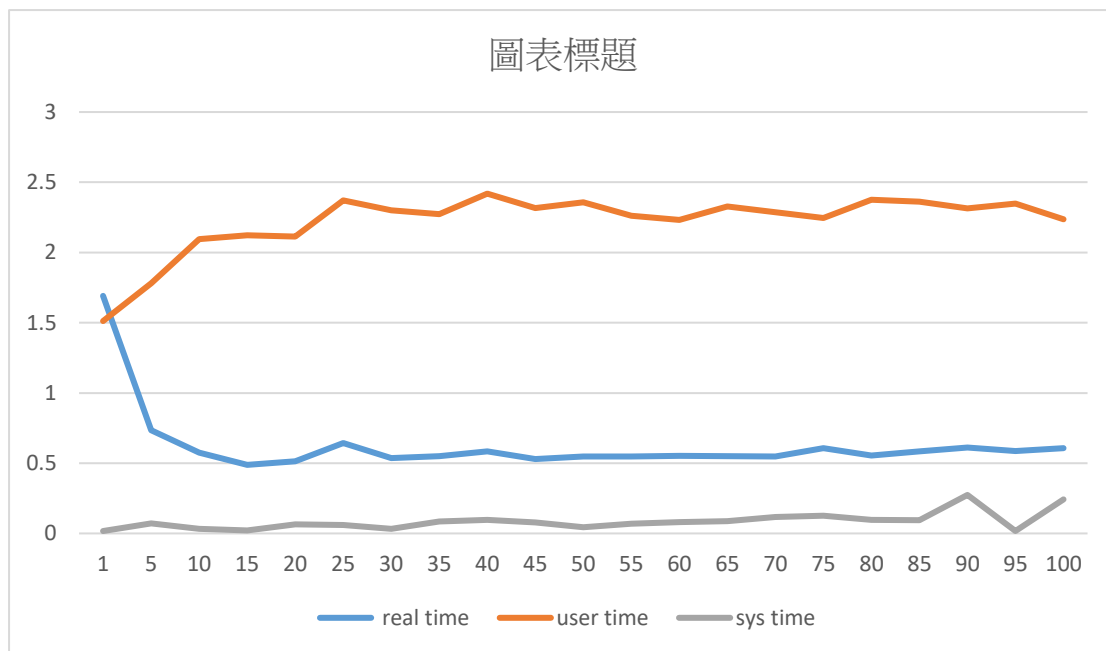
Hw4 report

首先來說一下 **real time**、**user time** 和 **sys time** 的問題。定義部分就是老師上課所講的，但我實測時發現可能會出現 **real time < user time + sys time** 的狀況。根據所查資料，可能因為是現在的電腦可能是多核的，所以會變成多個核同時跑，**user time** 和 **sys time** 同時累積，但 **real time** 只看最終時間差，不看所有 CPU 時間總和，所以才會出現這個情況。

說一下關於我的 **code** 的設計，我要把每個 **cell** 全部 **check** 一次，如果我的 **thread** 是分 19 個 **thread**，那我就每個 **thread** 處理 $\text{row} \times \text{col} / 19$ 個 **cell**，然後我用 **pthread_barrier_t** 來方便處理 **race condition** 的問題。一個設在 **check cell** 周圍 **live cell** 個數後面，等所有 **thread check** 完後，再全部更新 **cell**，這裡也設一個 **pthread_barrier_t**，等所有 **thread** 更新完後，再繼續下一個 **epoch**。

我的 2 個 **thread** 的時間是 **real:0.942s**，**user :1.447s**，**sys:0.043s**，20 個 **thread** 的時間是 **real:0.611s**，**user :2.064s**，**sys:0.112s**。可以看到 **real** 時間減少，可能是因為 **thread** 比較多更好利用多 **core** 的優點，不容易留下空餘的資源，所以時間減少。但 **user** 時間和 **sys** 時間都因為需要呼叫更多 **function** 而增加。

以下是我的折線圖:



根據此圖可看到，**real time** 一開始有大幅下降，大概是因為可以利用多 **core** 同時跑來降低時間，但很快就因為可以所有 **core** 都分到，增加 **thread** 也不會有多少改變。**User time** 也差不多，一開始因為多 **core** 同時跑所以時間倍增，但很快就因為可以所有 **core** 都分到，增加 **thread** 也不會有多少改變。**sys time** 因為比例很少，隨著 **thread** 增加，呼叫的 **system call** 也有些微增加，但不明顯。我的數據是說時間是有一點點上升的，但不太明顯就是了。