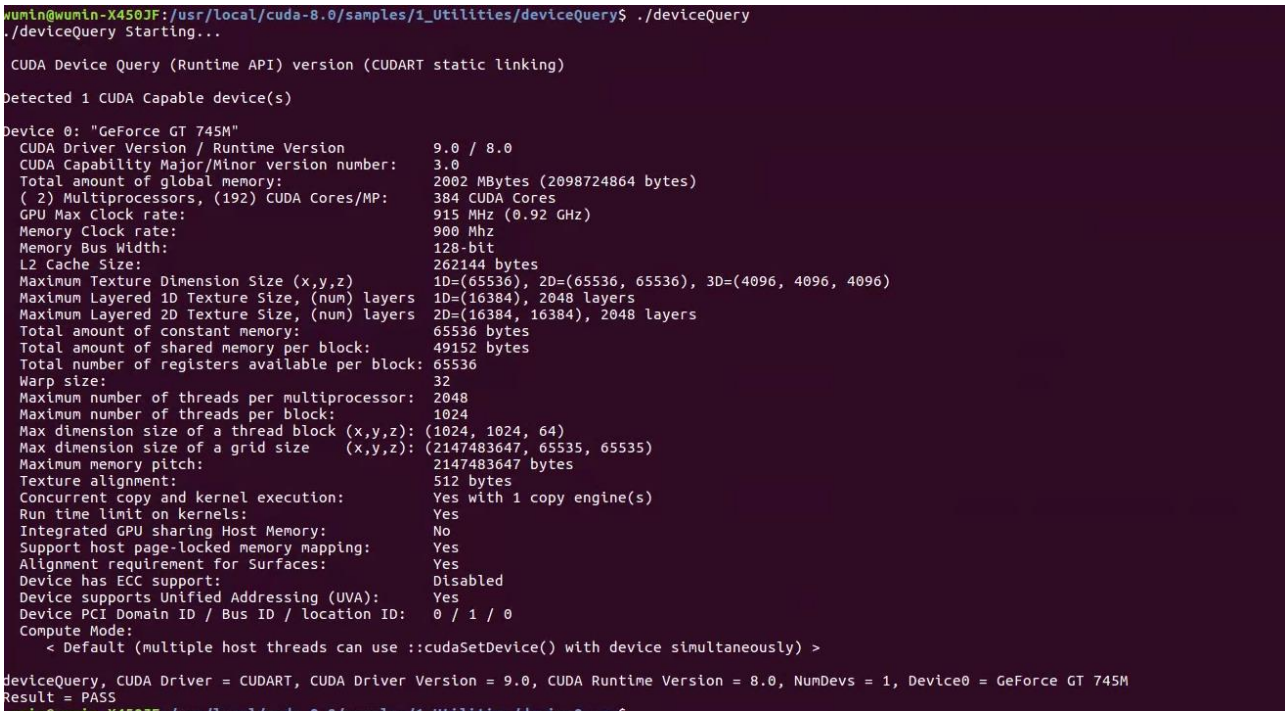
R05921034 吳明憲

我的機器規格:

CPU-i7 4700HQ

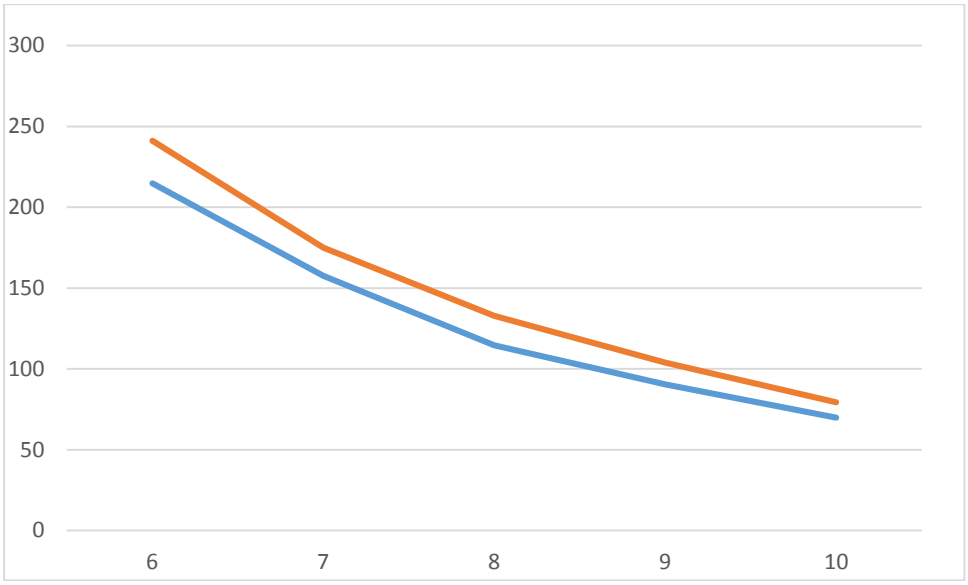
GPU GT745M

PART 1:



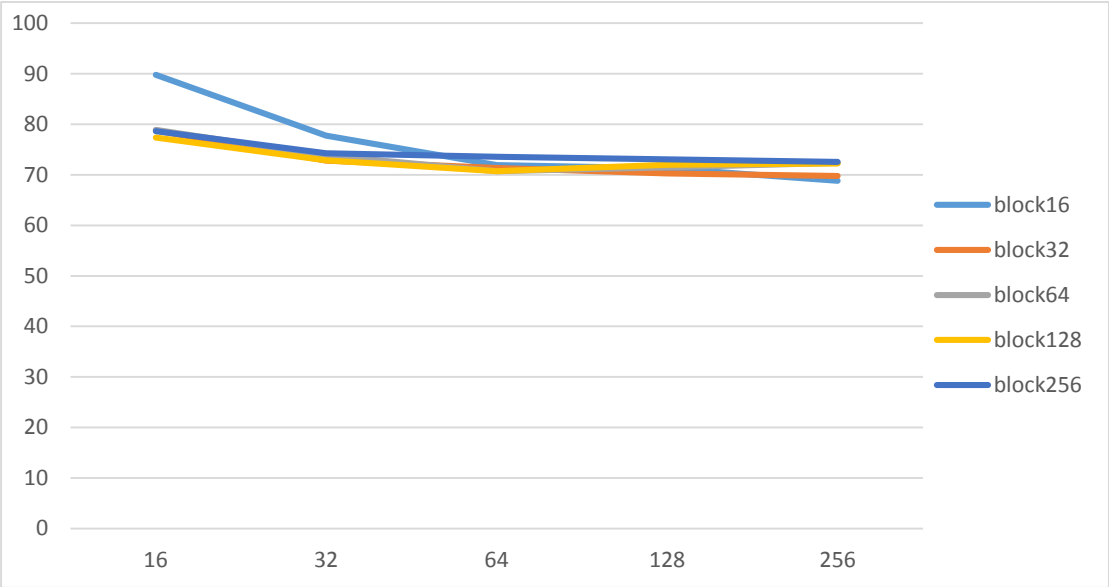
PART2

	gpu	cpu
6	214.717	241.17
7	157.53	175.067
8	114.635	132.89
9	90.334	103.972
10	69.921	79.3228



第一個我比較不同的 min sup，橫軸的 6~10 代表 0.06%~0.1% 可以看出兩個都是有下降的趨勢，不過在 minsup 變大時，GPU 以及 CPU 的時間就開始接近，應該在計算量變小時，平行的效用就比較沒有這麼顯著。

block/thread	16	32	64	128	256
16	89.8025	77.7768	71.9407	71.301	68.809
32	78.7438	72.8286	71.3639	70.2842	69.7747
64	78.8856	73.8229	70.7362	71.4813	72.3689
128	77.3606	72.8861	70.7014	71.9913	72.2337
256	78.6182	74.2916	73.5552	73.0619	72.5363



第二個是我 block 跟 thread 的比較，橫軸是 thread 數，在 thread 的數增加的時候有下降的趨勢，不過逐漸地平緩。應該是我的平行化的方式是對 a[i]&b[i] 的每一個 i 去給一個 thread 計算，由於 array 長度是固定的，造成了再給更多的 thread 也沒有再加快。

GPU DESIGN

那我的平行化的方式主要就是對每個 a[i]&b[i]去平行化，
並且盡量地去少搬動資料進出 GPU。
要是 thread 給得太少，就會造成我的平行化不夠多，不過到達一個數量之後就會飽和，應該是大於 array 的長度之後就會多餘的 thread，造成效能沒有增進，且若是沒有跟 array 的數量 align 的話可能會有稍微的上升。
由於我的 GPU 是非常舊的 GPU，似乎使用更新的 GPU 可以在更加快。