Parallel Programming HW3 REPORT

109062671 吳竣宇

1 Implementation

a. Blocked Floyd Warshall

b. Phase 2的切法:

依Spec中的pseudo code將pivot row和pivot column分別用不同的kernel計算

並讓gpu中的block一一對應矩陣中的block

Phase 3的切法:

依Spec中的pseudo code將pivot block的左上左下右上右下分別用不同的kernel計算

並讓gpu中的block一一對應矩陣中的block

c. hw3-2和hw3-3的configuration相同:

B locking factor: 64

- 為了盡可能maximum blocking factor並讓Spec的pseudo code中的每個phase都能將data複製到share memory中執行, 已知phase 3所需存取矩陣的blocks數量為3個(pivot column block, pivot row block, target block), 由於GeForce GTX 1080的share memory為49152bytes, 若我們將blocking factor設為64, 則能剛剛好將3blocks複製進share memory(3*64*64*sizeof(int) = 49152bytes)

Number of block:

- 依照Spec的pseudo code中phase 2和phase 3在每個round中需要執行的矩陣block數決定每個kernel所需的gpu blocks的數量

Number of threads: 1024

- 使用block中的threads數量的最大值以達到好的優化結果

d. 使用Unified Memory建立一個Virtual address讓cpu和gpus去存取同一個address space

e. hw3-1:
hw3-2:
hw3-3:

2 Profiling Results(hw3-2)

3 Experiment & Analysis

a. System Spec:

- All the experiments were executed in hades.

b. Blocking Factor:

c. Optimization(hw3-2):

以下為我有用到的優化技巧:

- CPU
- GPU baseline
- Use share memory
- Reduce the warp divergence + unroll the loop
    - Padding to the multiple of blocking factor
- Large blocking factor(B = 64)
- Coalesced memory access + Handle bank conflict
- I/O acceleration
- Reduce the number of calling syncthreads()

d. Weak scalability(hw3-3):

--

e. Time distribution(hw3-2):

–

4 Experience & Conclusion

(a). Experience:

- 要先仔細看實驗spec再來寫作業,否則沒照優化順序寫只能回去重作