EE217: GPU Architecture and Parallel Programming
Lab #2

Akber Raza
araza008@ucr.edu

February 6, 2019

Questions

1

```
Setting up the problem...0.001795 s
    A: 256 x 256
    B: 256 x 256
    C: 256 x 256
Allocating device variables...0.465119 s
Copying data from host to device...0.000272 s
Launching kernel...0.000173 s
Copying data from device to host...0.000297 s

Setting up the problem...0.001741 s
    A: 1024 x 64
    B: 64 x 1024
    C: 1024 x 1024
Allocating device variables...0.456570 s
Copying data from host to device...0.000281 s
Launching kernel...0.000534 s
Copying data from device to host...0.003433 s
```

$\Rightarrow 256 \times 256$ square matrix multiplication is 68% faster than 1024×64 and 64×1024 rectangular matrix multiplication.
Square matrix multiplication is faster because it needs to compute 65536 threads compared to 1048576 thread computations needed for rectangular matrix multiplication.

2 and 3

Number of element in output matrix: $64 \times 64 = 4096$
Number of global memory accesses required for each element: $64 + 64 = 128$
Total number of global memory accesses for tiled matrix multiplication: $64 \times 64 \times 128/16 = 32768$
Total number of global memory accesses for non-tiled matrix multiplication: $64 \times 64 \times 128 = 524288$

4

Title	8	16	32	Note
gpu_tot_sim_cycle	40252	26447	57554	Total cycles
gpu_tot_ipc	416.3975	455.3348	389.1463	Instruction per cycle
gpgpu_n_load_insn	524288	262144	131072	Total loads to global memory
gpgpu_n_store_insn	16384	16384	16384	Total stores to global memory
gpgpu_n_shmem_insn	4718592	4456448	4325376	Total accesses to shared memory

Table 1: Effects of tile size on GPGPU performance

5

Tile size of 32 resulted in least number of accesses to global memory while tile size of 8 resulted in most number of accesses to global memory. Within a given tile, each element of the input matrices is loaded once. Therefore, larger the tile size, fewer the accesses to global memory.

6

Tile size of 16 performed the fastest. Larger tile size does reduce global memory accesses; however, large block sizes also mean that multiple thread blocks may not be scheduled to the same SM, or induce boundary effects/thread divergence (in this case, there were no boundary effects). Therefore, this trade-off makes tile size of 16 the fastest.