Uriel Escobar
861219219
uesco001
EE147
Lab02

Lab 2: Tiled Matrix Multiplication

```
./sgemm-tiled
Setting up the problem...0.026458 s
   A: 1000 x 1000
   B: 1000 x 1000
   C: 1000 x 1000
Allocating device variables...0.549205 s
Copying data from host to device...0.002144 s
Launching kernel...0.161074 s
Copying data from device to host...0.003330 s
Verifying results...TEST PASSED
```

1. In your kernel implementation, how many threads can be simultaneously executing? Assume a GPU which has 30 streaming multiprocessors.

7680 threads are simultaneously executing because we have 30 SM handling 16*16 blocks, which have 256 threads each.

2. **Experiment with the Nvidia visual profiler, which is part of the CUDA toolkit, and use it to further understand the resource usage. In particular, report your branch divergence behavior and whether your memory accesses are coalesced.**

```
nvcc --ptxas-options="-v" main.cu
ptxas info    : 0 bytes gmem
ptxas info    : Compiling entry function '_Z7mysgemmiiiPKfS0_Pf' for 'sm_30'
ptxas info    : Function properties for _Z7mysgemmiiiPKfS0_Pf       v
   0 bytes stack frame, 0 bytes spill stores, 0 bytes spill loads
ptxas info    : Used 25 registers, 2048 bytes smem, 360 bytes    cmem[0]
/tmp/tmpxft_00001b90_00000000-10_main.o: In function `main':
tmpxft_00001b90_00000000-5_main.cudafe1.cpp:(.text+0x21b): undefined reference to `startTime'
tmpxft_00001b90_00000000-5_main.cudafe1.cpp:(.text+0x48e): undefined reference to `stopTime'
tmpxft_00001b90_00000000-5_main.cudafe1.cpp:(.text+0x4c2): undefined reference to `elapsedTime'
tmpxft_00001b90_00000000-5_main.cudafe1.cpp:(.text+0x8b2): undefined reference to `verify'
collect2: error: ld returned 1 exit status
```

From the output above we can conclude that for each sm we use 25 registers and use 2048 bytes for our smem which is our shared memory. and 360 bytes for constant memory

```
 nvcc --ptxas-options="-v" kernel.cu
ptxas info    : 0 bytes gmem
ptxas info    : Compiling entry function '_Z7mysgemmiiiPKfS0_Pf' for 'sm_30'
ptxas info    : Function properties for _Z7mysgemmiiiPKfS0_Pf
    0 bytes stack frame, 0 bytes spill stores, 0 bytes spill loads
ptxas info    : Used 25 registers, 2048 bytes smem, 360 bytes cmem[0]
/usr/lib/gcc/x86_64-redhat-linux/4.8.5/../../../../lib64/crt1.o: In function `_start':
(.text+0x20): undefined reference to `main'
collect2: error: ld returned 1 exit status
```

From the output above we can conclude that for each sm we use 25 registers and use 2048 bytes for our smem which is our shared memory, and 360 bytes for constant memory.

Both of them had the same results, so There is no branch divergence. All the threads execute a load instruction when we tile it. Because they are all in the same burst section, only one request is made and the threads are executed all at once; the memory access is coalesced.

3. How many times is each element of the input matrices loaded during the execution of the kernel?

Each element of the input is loaded (n/16+1) times for matrix A (n is the greatest column value).
Each element of the input is loaded (m/16+1) times for matrix B (m is the greatest row value).