ECE 4822

Homework 4

Lucas Raab

P01:

In p01, we were tasked with creating a simple vector addition program that is ran off of the GPU. Below is the results of profiling my program using the nvprof command. We can see that, 1. The job is actually being ran off the gpu and 2. Most of the 'lag' comes from allocating our output matrix, instead of the actual addition itself.

```
==802194== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   94.90%  473.12ms         1  473.12ms  473.12ms  473.12ms  vector_add(float*, float*, float*, int)
                    3.04%  15.142ms         1  15.142ms  15.142ms  15.142ms  [CUDA memcpy DtoH]
                    2.07%  10.299ms         2  5.1496ms  4.8791ms  5.4202ms  [CUDA memcpy HtoD]
      API calls:   67.48%  499.58ms         3  166.53ms  4.9947ms  489.09ms  cudaMemcpy
                   31.45%  232.80ms         3  77.601ms  90.637us  232.61ms  cudaMalloc
                    0.45%  3.3325ms       101  32.995us     147ns  1.7921ms  cuDeviceGetAttribute
                    0.35%  2.6272ms         1  2.6272ms  2.6272ms  2.6272ms  cudaLaunchKernel
                    0.27%  1.9709ms         3  656.98us  296.08us  856.53us  cudaFree
                    0.00%  10.494us         1  10.494us  10.494us  10.494us  cuDeviceGetName
                    0.00%  7.0320us         1  7.0320us  7.0320us  7.0320us  cuDeviceGetPCIBusId
                    0.00%  1.6660us         3     555ns     207ns  1.1980us  cuDeviceGetCount
```

P02:

Example 1:

In our first example, we were tasked with paralyzing our previous written code, to use 256 threads. Below is our nvprof results. We can see that

running. One important thing to note, is that despite running on 256 threads, we didn't see a 256 times increase in performance, we actually only saw a 37 times increase, which is much less than what I would

```
==802075== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   39.88%  15.177ms         1  15.177ms  15.177ms  15.177ms  [CUDA memcpy DtoH]
                   32.88%  12.515ms         1  12.515ms  12.515ms  12.515ms  vector_add(float*, float*, float*, int)
                   27.24%  10.369ms         2  5.1843ms  4.9414ms  5.4271ms  [CUDA memcpy HtoD]
      API calls:   83.30%  236.98ms         3  78.993ms  102.02us  236.76ms  cudaMalloc
                   13.69%  38.954ms         3  12.985ms  5.0523ms  28.387ms  cudaMemcpy
                    1.24%  3.5379ms       101  35.028us     135ns  1.9496ms  cuDeviceGetAttribute
                    0.90%  2.5633ms         1  2.5633ms  2.5633ms  2.5633ms  cudaLaunchKernel
                    0.86%  2.4335ms         3  811.17us  307.62us  1.0768ms  cudaFree
                    0.00%  9.6080us         1  9.6080us  9.6080us  9.6080us  cuDeviceGetName
                    0.00%  6.9580us         1  6.9580us  6.9580us  6.9580us  cuDeviceGetPCIBusId
                    0.00%  1.4900us         3     496ns     195ns  1.0570us  cuDeviceGetCount
                    0.00%     718ns         2     359ns     163ns     555ns  cuDeviceGet
                    0.00%     603ns         1     603ns     603ns     603ns  cuDeviceTotalMem
                    0.00%     323ns         1     323ns     323ns     323ns  cuModuleGetLoadingMode
                    0.00%     257ns         1     257ns     257ns     257ns  cuDeviceGetUuid
```

intuitively expect.

Example 2:

```
==802413== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   60.11%  15.926ms         1  15.926ms  15.926ms  15.926ms  [CUDA memcpy DtoH]
                   38.81%  10.282ms         2  5.1410ms  4.8496ms  5.4323ms  [CUDA memcpy HtoD]
                    1.08%  286.74us         1  286.74us  286.74us  286.74us  vector_add(float*, float*, float*, int)
      API calls:   74.37%  231.53ms         3  77.176ms  91.231us  231.33ms  cudaMalloc
                   14.87%  46.280ms         1  46.280ms  46.280ms  46.280ms  cudaLaunchKernel
                    8.85%  27.555ms         3  9.1851ms  4.9639ms  17.076ms  cudaMemcpy
                    1.12%  3.4956ms       101  34.609us     139ns  1.8644ms  cuDeviceGetAttribute
                    0.78%  2.4216ms         3  807.20us  323.95us  1.0699ms  cudaFree
                    0.00%  10.314us         1  10.314us  10.314us  10.314us  cuDeviceGetName
                    0.00%  7.9340us         1  7.9340us  7.9340us  7.9340us  cuDeviceGetPCIBusId
                    0.00%  1.4780us         3     492ns     206ns  1.0360us  cuDeviceGetCount
                    0.00%     885ns         1     885ns     885ns     885ns  cuDeviceTotalMem
                    0.00%     648ns         2     324ns     157ns     491ns  cuDeviceGet
                    0.00%     332ns         1     332ns     332ns     332ns  cuModuleGetLoadingMode
                    0.00%     284ns         1     284ns     284ns     284ns  cuDeviceGetUuid
```