Fall 2020 CIS 345/545 Homework 3

(Due Nov. 12)

The purpose of this assignment is to introduce you the CUDA GPU environment (Pascal and later architectures which support Unified Memory). Use the following command on a Linux workstation in our lab to extract the necessary files:

```
tar xvfz ~cis345s/pub/gpu345.tar.gz
```

Part I: deviceQuery

Use `make` to build the executable file `deviceQuery` under the subdirectory `deviceQuery` and then run it. Take a screenshot of the result. How many Streaming Multiprocessors (SM)? How many CUDA cores? How much global memory? What is the GPU clock rate?

Part II: Page Faults

A copy of the sample program `vAddUM.cu` discussed in the class can be found under the subdirectory `vAddUM`. Use the following command to compile it:

```
nvcc vAddUM.cu -o vAddUM
```

Then, use the command below to collect and view profiling data from running `vAddUM`:

```
nvprof ./vAddUM
```

Note that we are interested in the page fault numbers shown in the last few lines of the profiling result. Record the numbers of `Host To Device`, `Device To Host`, and `Total CPU Page faults` for later comparisons.

Assume that in the program `vAddUM.cu`, data arrays are allocated initially in the GPU memory when the function `cudaMallocManaged()` is called. Therefore, the data arrays `x` and `y` will be migrated back and forth when the CPU initializes the arrays and then GPU uses them to calculate. One approach to reducing page migrations is to move initialization from the CPU to the GPU. To implement this, copy the file `vAddUM.cu` to another file `vAddUM_Init.cu`. Next, edit the file `vAddUM_Init.cu` to remove the initialization of the data arrays `x` and `y` in `main()` and then add the following code:

```
__global__ void init(int n, float *a, float *b) {
    int index = threadIdx.x + blockIdx.x * blockDim.x;
    int stride = blockDim.x * gridDim.x;
    for (int i = index; i < n; i += stride) {
      a[i] = 1.0f;
      b[i] = 2.0f;
    }
}
```

You also need to add a kernel launch of the function `init()` before the line `cudaEventRecord(start);`. Compile and run it with `nvprof`. Take a screenshot and record the numbers of `"Host To Device"`, `"Device To Host"`, and `"Total CPU Page faults"`, respectively.

To reduce dynamic page migrations, an alternative way is by using Unified Memory prefetching to move the data arrays to the GPU after initializing them in the CPU. CUDA provides `cudaMemPrefetchAsync()`

for this purpose. Again, copy the file vAddUM.cu to a new file vAddUM_Pfetch.cu and add the following code before the kernel launch:

```
// Prefetch the data to the GPU
int device;
cudaGetDevice(&device);
cudaMemPrefetchAsync(x, N*sizeof(float), device, NULL);
cudaMemPrefetchAsync(y, N*sizeof(float), device, NULL);
```

Also, add the following code to asynchronously prefetch the array z back to CPU before the final for loop:

```
cudaMemPrefetchAsync(z, N*sizeof(float), cudaCpuDeviceId, NULL);
```

Compile and run it with nvprof. Record the numbers of "Host To Device", "Device To Host", and "Total CPU Page faults".

Please note that in addition to using a table to show the recorded numbers, you also have to explain the experimental results in your report.

Part III: Oversubscription Execution

Copy the file vAddUM.cu to another file called vAddUM_Over.cu. Change the value of the variable N from 1<<20 to 1<<28. What is the total memory size (in bytes) of the three arrays x, y, and z? Does it exceed the global device memory? Compile and run the program. Take a screenshot of the result. Explain why or why not the program can be executed.

Turning it in

Each group (two students) has to submit your report and program electronically. You have to put your report file (i.e. hw3_report.pdf) under the NVIDIA_CUDA_345 dir. Before you submit, change the directory to the parent directory of the NVIDIA_CUDA_345 dir and use the following command to submit the whole NVIDIA_CUDA_345 dir:

```
turnin -c cis345s -p hw3 NVIDIA_CUDA_345
```

Your report should include the description of the code, screenshots, experimental results and explanation, etc. The document should be typed. The cover page should contain your photos, names and the login-id the group used to turnin. Start on time and good luck. If you have any questions, send e-mail to sang@cis.csuohio.edu.