

Pre-class Assignment 11, due 10/16

Reading

Read Sections 1 and 2 of the [CUDA C Programming Guide](#)

Then, depending on your language of choice, read the following:

C/C++

- <https://devblogs.nvidia.com/easy-introduction-cuda-c-and-c/>
- <https://devblogs.nvidia.com/using-shared-memory-cuda-cc/>
- <https://devblogs.nvidia.com/efficient-matrix-transpose-cuda-cc/>

Fortran

- <https://devblogs.nvidia.com/easy-introduction-cuda-fortran/>
- <https://devblogs.nvidia.com/using-shared-memory-cuda-fortran/>
- <https://devblogs.nvidia.com/efficient-matrix-transpose-cuda-fortran/>

Review

1. Write up a list of the key points from the reading.
2. What questions do you have about the subject material covered in the reading?

Questions

1. List the CUDA *thread* hierarchy from smallest to largest.
2. List the CUDA *memory* hierarchy from smallest to largest.

Exercises

In this exercise you will run a simple CUDA program on the HPCC. Follow the steps below. You will be running the SAXPY code in the "Easy Intro..." article from above.

1. Log onto the HPCC gateway then, from there, log onto `dev-intel18`.
2. Request an interactive job with a GPU using the following command.

```
$ salloc -N 1 -n 1 --gres=gpu:1 --time=00:30:00
```

3. Once your interactive job starts, you should be on a node with name `nv1-00*`. Now, copy and paste the `SAXPY` code from the Intro to a local file on your node. Use a file extension of either `.cu` or `.cuf` depending on whether you are using C/C++ or Fortran.
4. Load the appropriate CUDA module on your interactive node:

```
$ module load CUDA/9.2.88
```

5. Now compile the code with the NVIDIA CUDA compiler:

```
$ nvcc -o saxpy.o saxpy.cu
```

6. Finally, run the code! To utilize the GPU you have requested, you must use `srun`. So,

```
$ srun ./saxpy.o
```

What to turn-in

Commit your write-up, including review and questions, as a plain text, markdown document, or pdf to your assignment repo *before the start of class*.