**Blue Waters Petascale Semester Curriculum v1.0**

**Unit 7: CUDA**

**Lesson 10: Numba for CUDA GPUs**

**Instructor Guide**

*Developed by Sanish Rai for the Shodor Education Foundation, Inc.*



*Except where otherwise noted, this work by The Shodor Education Foundation, Inc. is licensed under CC BY-SA 4.0. To view a copy of this license, visit*[*https://creativecommons.org/licenses/by-sa/4.0*](https://creativecommons.org/licenses/by-sa/4.0)

*Browse and search the full curriculum at*[*http://shodor.org/petascale/materials/semester-curriculum*](http://shodor.org/petascale/materials/semester-curriculum)

*We welcome your improvements! You can submit your proposed changes to this material and the rest of the curriculum in our GitHub repository at*[*https://github.com/shodor-education/petascale-semester-curriculum*](https://github.com/shodor-education/petascale-semester-curriculum)

*We want to hear from you! Please let us know your experiences using this material by sending email to* [*petascale@shodor.org*](mailto:petascale@shodor.org)

Notes:

Make sure that CUDA is installed and working. For more information check: Lesson 7.10 Installing and running CUDA

Make sure that python is installed. To install python using Anaconda

<https://docs.anaconda.com/anaconda/install/windows/>

After installing Anaconda, check that Jupyter notebook is working properly.

Make sure that Numba is installed. To install using Conda, type:

conda install cudatoolkit

If you are not using Conda or if you want to use a different version of CUDA toolkit, check here: https://numba.pydata.org/numba-doc/latest/cuda/overview.html

Exercises

1. Review the code Numa\_example.ipynb in Jupyter notebook
2. Similar to the given example, write a python code with Numba to add two 1D lists

Since it is a 1D array, you need to use single dimension instead of 2 Dimension

Solution: Numba\_exercise.ipynb

* In this work, the 2D needs to be changed to 1D.
* Lists should be 1D
* Memory allocation should be for 1D
* For blockspergrid, blockspergrid\_y is not required
* In the kernel, cud.grid(1) should be used to get thread id
* Add only the x dimension

**Common Pitfalls for Students and Instructors**

* Students will need at least some basic knowledge of CUDA concepts and GPU, working with Numba without these prerequisites can make the concept confusing and implementation difficult
* Make sure that CUDA, Python and Numba are installed and set correctly in path if using from command line