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

**Unit 7: CUDA**

**Lesson 6: CUDA Atomic Functions**

**Sample Assessment**

*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)

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

•Exercise 1: Run the given program for atomic addition and analyze the output of with and without atomic function

The program source code is given in atomic\_addition.cu

•Exercise 2:

This program is very similar to given example, students just need to use atomicMax method.

The syntax is atomicMax(max,a[thread\_id]);

Where max is the global variable to store the max element by all the threads and a is the global array containing the data.

The solution for the program is given in atomic\_max.cu

•Write a CUDA program to find the max value in a list using atomicMax

•Create a list of size 900,000 and set the values from 1 to 900,000

•Write a CUDA kernel to find max from the list using simple comparison

•Eg. Let max = 0, if val> max, max = val

•Write a CUDA kernel to find max from the list using atomicMax

•Eg. Let max = 0, atomicMax(max,val)

•Check your output to ensure that atomicMax finds the correct max, that is 900,000

Common Pitfalls for Students and Instructors

* Mistakes in proper allocation of memory in CPU and GPU
* Mistakes in writing kernel and using atomic functions
* Students might be confused with using proper number of threads and blocks
* Students might make mistake in copying data from CPU to GPU or vice versa