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

**Unit 7: CUDA**

**Lesson 7: Parallel Reduce in CUDA**

**Exercise Instructions for Students**

*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)

1. Exercise 1: Review the given source code for parallel reduction using sum operator for 2^20 elements

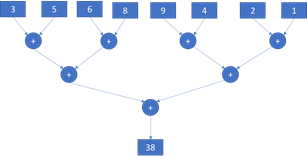
To run the program:

nvcc <filename>

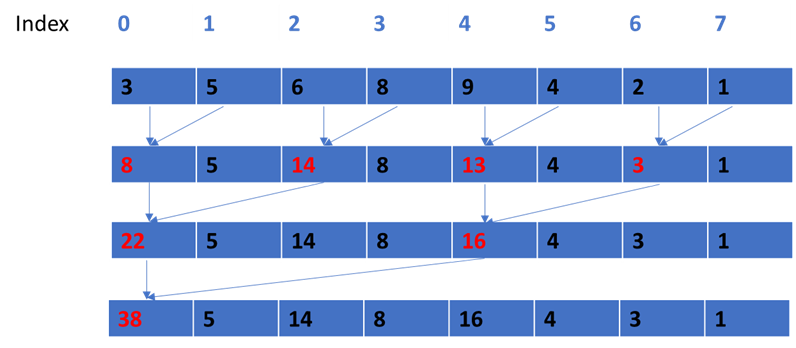
a //(in windows)

or ./a //(in linux)

1. Exercise 2: Write a kernel that performs reduction as shown in figure using shared memory for 2^20 elements.



Here is how it can be designed



Elements in red represent the reduced values.

Note: Since each block will compute a single result, you will need to add all the block results at the end to get the final reduced result.

1. Exercise 3: Write a program for parallel reduction using max operation to find the max value among 2^20 elements
   1. First define host variables and a serial method to find max
   2. Define device variables and initialize them
   3. Create a kernel to find max similar to exercise 1 using global memory
   4. Create a kernel to find max similar to exercise 1 using shared memory