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

**Unit 4: OpenMP**

**Lesson 12: OpenMP Tasks**

**Sample Assessment**

*Developed by Cameron Foss 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)

Students can be provided with the serial version of either the compute pi or fibonacci codes and then be asked to parallelize them with openMP, particularly to apply OpenMP tasking to the recursive parts. They can then be asked to run the codes for different input parameters and number of OpenMP threads.

An example assignment is below:

1. Parallelize the serial version of the compute pi code with openMP
2. Use openMP tasks on the recursive function pi\_r
3. Once the code has been parallelized, perform the following:
   1. Compute pi with [100,500,1000,5000,10000,1000000,100000000] numbers of rectangles for both the serial and parallelized code with 2 and 4 threads. Note the runtime and accuracy of the approximation in each.
   2. Comment on trends you see with the runtime when increasing the number of rectangles and the number of threads.