

Parallel and Distributed Computing

Assignment-3 (OpenCL)

When to submit: Tuesday 21st March 08:59 PM

Late Policy: 30% submission marks deduction for late up to 03 hours (21st March 11:59PM)

Zero marks for submissions which are late more than 03 Hours

What to Submit: One Zip folder (no .rar file please) containing all source code implementations (any folder containing binary files will be marked zero).

Total marks: 100

Write two OpenCL programs (for CPU device) to implement the following scenarios. For each scenario you have to implement a separate OpenCL program (both host and kernel application). Please use one work-item (OpenCL thread) to mimic functionality of 1 MPI process. You are free to design solution as per your logic, however one work-item should be functionally equal to 1 MPI process related concept. Communications will be via a common memory, therefore do consider proper synchronizations.

Concept-1: Scatterv [50 Marks]

There are 10 processes that are participating in the communication equivalent to MPI “Scatterv”. Out of these 10 processes one process should be randomly selected as coordinating process (who’s buffer will be scattered). The data-buffer to be scattered should be dynamically created [1000 min-elements to 10000 max-elements] and initialized by each process. The *sendcount* and *displacement* data-structure should also be randomly generated (for 10 processes) in coordinating process. After scattering data, every process should calculate sum of the received data values. All the processes should communicate their local sums to main program (host program in this case) and the main program then should find a global sum and display the result on the screen.

Concept -2: Gatherv [50 Marks]

There are 10 processes that are participating in the communication equivalent to MPI “Gatherv”. Out of these 10 processes one process should be randomly selected as coordinating process (who’s buffer will be prepared by gathering data sent by others). The data-size to be gathered should be dynamically created by each process [20 min-elements to 200 max-elements] and initialized. The *sendcount* and *displacement* data-structure should also be dynamically/randomly prepared (for 10 processes) and used by the coordinating process. After gathering data, the coordinating process should calculate a dot product using another randomly created and prepared sparse data-array (sparse because the gathered data have been prepared using displacements thus could result in vacant data values). The coordinating process should convey the dot product data to main program (i.e., host). The host program will sum all values and display the result.

Best of luck!