

## Assignment 7—OpenMP Target Offload and DPC++ Programming

Due: November 11 (Wed), 2020

The purpose of this assignment is to gain hands-on experience in new open standards for programming heterogeneous computers accelerated by graphics processing units (GPUs) and other accelerators. Specifically, you will practice directive-based OpenMP target offload and unified data-parallel programming language, DPC++.

### Prerequisite

---

We will practice both OpenMP target and DPC++ on Intel developer's cloud (DevCloud). To do it, *create your DevCloud account by registering at <https://devcloud.intel.com/oneapi>.*

### Part I: OpenMP Target Offload Computation of $\pi$

---

In this part, you will write a GPU offload program (name it `omp_teams_pi.cu`) to compute the value of  $\pi$  using `omp target`, `teams` and `distribute` constructs.

#### (Assignment)

1. Modify the simple OpenMP target program `omp_target_pi.cu` to its teams-distribute counterpart `omp_teams_pi.cu`, following the lecture note on “OpenMP Target Offload for Heterogeneous Architectures”. *Submit your code.*
2. Compile and run your program on a GPU-accelerated computing node on DevCloud. *Submit your output*, which should look like the following:

```
u49162@login-2:~$ cc -o omp_teams_pi omp_teams_pi.c -fopenmp
u49162@login-2:~$ qsub -I -l nodes=1:gpu:ppn=2
qsub: waiting for job 714173.v-qsvr-1.aidevcloud to start
qsub: job 714173.v-qsvr-1.aidevcloud ready
u49162@s001-n181:~$ ./omp_teams_pi
PI = 3.141593
```

### Part II: DPC++ Computation of $\pi$

---

In this part, you will experience the compilation and running processes for a DPC++ program (`pi.cpp`) to compute the value of  $\pi$ . While programming is not required for this part since C++ is not prerequisite to this class, please use this opportunity to learn the essence of C++ and DPC++ programming by going through the code and understanding why it works following the lecture note on “Data Parallel C++ (DPC++) for Heterogeneous Architectures”.

#### (Assignment)

1. Compile and run `pi.cpp` node. on a GPU-accelerated computing node on DevCloud. *Submit your output*, which should look like the following:

*Submit your output*, which should look like the following:

```
u49162@login-2:~$ dpcpp -o pi pi.cpp
u49162@login-2:~$ qsub -I -l nodes=1:gpu:ppn=2
qsub: waiting for job 714154.v-qsvr-1.aidevcloud to start
qsub: job 714154.v-qsvr-1.aidevcloud ready
u49162@s001-n160:~$ ./pi
Running on: Intel(R) Gen9 HD Graphics NEO
Pi = 3.14159
```