Minh Tran

Assignment 7 CSCI 596: OpenMP Target Offload and DPC++ Programming

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 π 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
```

Solution:

```
#include <omp.h>
#include <stdio.h>
#define NBIN 1000000
#define NTRD 96
#define NTMS 12 // ### number of teams

int main() {
        float step,sum=0.0,pi;
        step = 1.0/(float)NBIN;

        // data privatization among teams ###
        float sum_teams[NTMS];
        for (int j=0; j<NTMS; j++) sum_teams[j] = 0.0;

        // copy variables step and sum
        // #pragma omp target map(step,sum) ###
        // ###
        #pragma omp target teams map(step, sum_teams) num_teams(NTMS)
        {
```

```
                // ###
                #pragma omp distribute // distribute the work between num_teams
                // for each team, need to define index of thread
                for (int j=0; j<NTMS;j++){
                        long long ibgn = NBIN/NTMS*j;
                        long long iend = NBIN/NTMS*(j+1);
                        if (j==NTMS-1) iend = NBIN;
                        // modified for offset and private accummulator
                        // thread reduction of sum; specify number of threads
                        # pragma omp parallel for reduction(+:sum_teams[j])
num_threads(NTRD)
                        for (long long i=ibgn;i<iend; i++) {
                                float x = (i+0.5)*step;
                                sum_teams[j] += 4.0/(1.0+x*x);
                        }
                }
        }

        // ###
        for (int j=0; j<NTMS;j++) sum+= sum_teams[j];
        pi = sum*step;
        printf("PI = %f\n",pi);
        return 0;
}
```

To run on devcloud:

- Open local cygwin terminal to copy local files to cloud
- Run **scp -r csci596-as07 devcloud:home/u51841/project_mt/** -> copy all files to folder in cloud
- Open cygwin and run ssh devcloud
- Then go to the project folder on cloud and compile the program: **cc -o omp_target_pi_teams omp_target_pi_teams.c -fopenmp**
- Run interactive job on a GPU-accelerated computing node: **qsub -I -l nodes=1:gpu:ppn=2**
- Run the program: **./omp_target_pi_teams**

```
u51841@s001-n157:~/project_mt/csci596-as07$ ls
 as07.pdf                   omp_target_pi.c         pi.cpp
 assigment7_minh_tran.docx  omp_target_pi_teams     README
 devcloud_instruction       omp_target_pi_teams.c  '~$sigment7_minh_tran.docx'
u51841@s001-n157:~/project_mt/csci596-as07$ ./omp_target_pi_teams
PI = 3.141593
```

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 π. 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. Compiler 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
```

Solution: