

Project 3 (Part 1): Easy as Pi

Due 10/21/2015 @ 12pm (noon)

TOPICS:

- ↪ CUDA Memory Structure
- ↪ cuRand library
- ↪ Monte Carlo Method
- ↪ Atomic Operations
- ↪ File I/O



BACKGROUND:

During class, we discussed how the Monte Carlo Method can be used to compute the value of pi.

DIRECTIONS:

Design and implement C and CUDA programs that called MCPi.c and MCPi.cu, respectively.

The MCPi programs will use one command line argument that corresponds to the number of random points used to evaluate the value of pi. The C program must implement a serial code via a function call, and the CUDA program must implement a parallel code via a kernel call. In addition, the C program must use a host-based pseudorandom number generator (e.g., srand48/drnd48) and the CUDA program must use a device-based pseudorandom number generator (e.g., cuRAND). **It is completely up to you to implement the serial and parallel algorithms, including any parameters associated the implementation.**

You must check for 1) the appropriate number of command line arguments and 2) whether the command line argument corresponds to a positive number (you may round floating point inputs). Appropriate error messages must be issued, followed by a graceful exit.

There will be two sets of outputs: 1) the value of pi outputted to the screen and 2) a histogram of the normalized frequencies of the random numbers outputted to a file name freq.dat where the first column is the numbers 0-9 and the second column is the normalized frequency of that number.

You will then construct using your favorite graphing program a histogram that reflects the frequencies of the random numbers. Submit only an image of your graph.

IMPLEMENTATION NOTES:

Any program that does not compile or does not have correctly constructed Makefiles will not be graded. The C program must compile using g++ and the CUDA program must compile using nvcc.

COMMENTS AND STYLE:

Although there will be no formal policy on commenting and style, the reader should be able to easily follow the main purpose of the code. Each set of code that does something significant must be commented. The variable names should be easily recognizable and acronyms should be avoided if possible.

Do not be surprised if help is not forthcoming if your code is poorly commented and/or difficult to follow. You have been warned.

PROJECT SUBMISSION:

The programs should be in appropriate directories named "MCPi-cpu" and "MCPi-gpu". Both programs must have corresponding Makefiles. The contents of the directories must be archived in a tarball that is gzipped called Proj3-1.tar.gz.

Place the gzipped tarball in your Drop Box on Sakai before it is due.

PLEDGED WORK POLICY:

Assignments in Computer Science courses may be specified as "pledged work" assignments by the professor of the course. When an assignment is specified as "pledged work" the only aid that the student may seek is from either the course professor or an assistant that the professor has explicitly specified. On "pledged work" assignments the student may not use the services of a tutor.

For this project, you may discuss only basic C syntax with others. Any other discussions of the project are strictly prohibited except with the professor of the course. Your code and your implementation of the project must be the product of your own work.