

CS451 “Introduction to Parallel and Distributed Computing” Homework 4

Submission:

- Due by 11:59pm of 04/30/2021(Friday)
 - Send chameleon ID to Dr. Lan and CC mduoley1@hawk.iit.edu by 18 April 2021 (Sunday)
 - Late penalty: 20% penalty for each day late
 - Please upload your assignment on the Blackboard with the following name: **CS451_**
LastName_FirstName_HW4
 - Please do NOT email your assignment to the instructor and TA!
-

In this assignment, you will design and code a CUDA-C/C++ version of a Matrix normalization algorithm. Reading the Wikipedia page on Standard Score is encouraged http://en.wikipedia.org/wiki/Standard_score. In Data Analytics and Machine Learning, matrices usually contain data points (rows) in a particular space defined by their attributes (columns). Sometimes, and because attributes may represent things that are very different in nature, normalization by column is required.

Your goal is to take advantage of the power of GPUs to perform this task as fast as possible. Realize that column normalization is composed of three steps per column:

1. Calculating the mean of the column
2. Calculating the standard deviation (which requires the mean)
3. Finally, calculating the normalized value by performing the following calculation (where B is the normalized matrix of A)

$$B[\text{row}][\text{col}] = (A[\text{row}][\text{col}] - \text{mean}) / \text{standard_deviation}$$

It is possible to see that the first two steps can be achieved with a REDUCTION algorithm. In this part, you need to be very careful with your design decisions. Where you put the data, and how you perform the reductions matter. As a hint, you may consider splitting the reductions in two: first, inside the values in each block and, second, reducing the totals for every block. Once the mean and standard deviation are calculated, the third step is straightforward.

The sequential code (*matrixNorm.c*) is provided with this assignment, and can be used as a reference for debugging and performance comparison. **Your code will be graded partially on the efficiency of your algorithm.**

You should write documents explaining your design decisions very clearly and your performance evaluation. Even if your code does not work, or is not efficient enough, you should write the reasons you think that is so. So you should upload the following documents for this assignment:

1. Source Code: .cu file
2. README: how you compile and run the code
3. Design Document
4. Performance Evaluation (comparing CPU & GPU performance and measuring the efficiency of GPU performance - **include graphs**)