Computing at Scale
Assignment 0

Due Thursday, January 16th at 11:59pm Submit on Gradescope

This first assignment is designed to make sure that everyone has a development environment set up and can compile and run a C++ program. For this assignment, you will implement some basic linear algebra operations. And set up the git/github repo for this class.

1. Implement the following functions including any documentation, README files, test cases, etc. that you think are necessary to make your code understandable and usable by others.

    (a) Implement a function that computes the product of a dense matrix with a vector.

    (b) Implement a function that computes the product of two dense matrices.

    (c) Implement a driver program that takes two filenames and the type of computation to perform as an integer (matrix-vector as 0 or matrix-matrix as 1) and prints the result to the standard output in the matrix market format. The input files will contain the matrices in matrix market format `https://math.nist.gov/MatrixMarket/formats.html`.

2. Tooling

    (a) Create a git repository and commit your code to a branch called assignment0 (we will cover this in the 3rd class).

    (b) Create a github repo for the computing at scale class. And push your assignment0 branch to the repo.

3. Please answer the following questions and add them to a markdown file called assignment0.md in your repository. Markdown is a simple markup language that is easy to read and write. You can find a guide to markdown here `https://daringfireball.net/projects/markdown/syntax`.

    (a) How did you test your code? What are the limitations of your testing?

    (b) What are the limitations of the matrix market format for dense matrices?

    (c) If you know the size of the matrices at compile time, how would you change your code to take advantage of this information?

    (d) Was your matrix representation row-major or column-major? How did you decide? Do you think this will have an impact on performance?

    (e) What approach did you take to document your functions and driver program?

    (f) Is your code robust to errors in the input files? How did you test this?

    (g) Are there any limitations in your implementation you are aware of? Do you have any ideas on how to address them? e.g., what is the maximum size of the matrices you can handle?

    (h) If you wanted to optimize the performance of your code, what approach would you take?