Question – 1: Matrix-vector operations on a GPU

The times for ***Vector-Vector multiplication (dot product)*** are shown below in the table for the various GPU.

GPU Name	Dimension (p)	Time in Cuda (s)	Time in CPU (s)	Bandwidth (GB/s)
cuda1.cims.nyu.edu	262144	0.000561	0.001266	33.678
cuda2.cims.nyu.edu	262144	0.000422	0.000724	44.757
cuda3.cims.nyu.edu	262144	0.000690	0.001610	27.348
cuda4.cims.nyu.edu	262144	0.000993	0.001160	19.008
cuda5.cims.nyu.edu	262144	0.001310	0.000729	14.804

The times for ***Matrix-Vector multiplication*** are shown below in the table for the various GPU.

GPU Name	Dimension (m*n)	Time in Cuda (s)	Time in CPU (s)	Bandwidth (GB/s)
cuda1.cims.nyu.edu	1048576	0.001111	0.006215	60.152
cuda2.cims.nyu.edu	1048576	0.001169	0.005614	57.462
cuda3.cims.nyu.edu	1048576	0.001015	0.005947	52.012
cuda4.cims.nyu.edu	1048576	0.002237	0.006469	30.034
cuda5.cims.nyu.edu	1048576	0.002936	0.003289	22.883

Question – 2: 2D Jacobi method on a GPU

The code is available in GitHub repo.

Question – 3: Update on final project

We are tying to parallelize different graph algorithms like BFS, Bellman Ford, PageRank, graph-connectivity, etc. These algorithms will be parallelized by modifying the serial version of the algorithm to a corresponding parallel one and then using OpenMP. The basic framework is ready for testing various graphs especially social media graphs and others. Nodes ranging from 10 to 10,000 has already been tested on the framework, and now we will be testing it for bigger graphs. Additionally, the parallel version of BFS has already been coded, with the testing left.

Though no major issues were accounted, we had issues with the use C++ atomic and also writing a makefile for such a complicated code. Another issue we suspect is the use of GPUs as we will not have much floating-point operations.