**ECE565 - Fall 2019**

**Assignment #4**

**Parallel Programming with OpenMP**

Ni Nan (nn75), Jing Liming ()

**Part 1: Histogram**

1. Use an array of locks.

(1) Describe your OpenMP directives and any code changes you make

(code snippet)

(2) Record the execution time reported by the program when running with 2, 4, and 8 threads.

(picture)

2. Use atomic operation

(1) Describe your OpenMP directives and any code changes you make

(code snippet)

(2) Record the execution time reported by the program when running with 2, 4, and 8 threads

(picture)

3. Creative solution

(1) Describe your OpenMP directives and any code changes you make

(code snippet)

(2) Record the execution time reported by the program when running with 2, 4, and 8 threads

(picture)

4. Analysis

(1) Observations about which version performs best and discuss why you think that is the case.

(picture of comparison of three approaches and against the sequential version)

(2) Describe how your own creative solution removes the need for using locks and atomic operations while still producing correct results.

**Part 2: AMG**

1. Analyze profile

2. Code changes

(1) Describe your OpenMP directives and any code changes you make

(code snippet)

(2) Summarize sequential vs. optimized parallel performance across 1, 2, 4, and 8 threads

(picture)

(profile)