# Homework 2 - Histogram

Philip Nelson

4th October 2021

## Histogram

The project includes a CMakeLists.txt with three projects: histogram, histogram_benchmark, and tests. The project histogram requires command line arguments for number of threads, bin count, min measurement, max measurement, data count, and optionally a seed for the pseudo random number generator (default 100). It prints the bin maxes and bin counts to stdout. The pseudo random number generator, `std::rand`, was seeded with `100` for each trial to ensure the generated data was identical. This kept the results the same from different executions and different number of threads.

The project histogram_benchmark runs the histogram program with the same command line arguments but performs timing measurements and averages them over a number of trials. It prints the number of threads used and the time in milliseconds to stdout.

The final project is tests which includes unit tests for various components internal to the histogram computation.

The following commands can be used to build and run the project.

```
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release ..
make
./histogram/histogram 4 10 0.0 5.0 100000

# bin_maxes: [ 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 ]
# bin_counts: [ 9873, 9980, 9920, 9932, 9997, 10136, 9909, 10030, 10187, 10036 ]
```

## Results

The timing trials were performed on a vector of $500,000,000$ elements with a 50 run average for each thread count. The trials were run on a 6 core / 12 thread Intel Xeon CPU. The results of the trials can be seen in Figure 1 and Table 1 below.

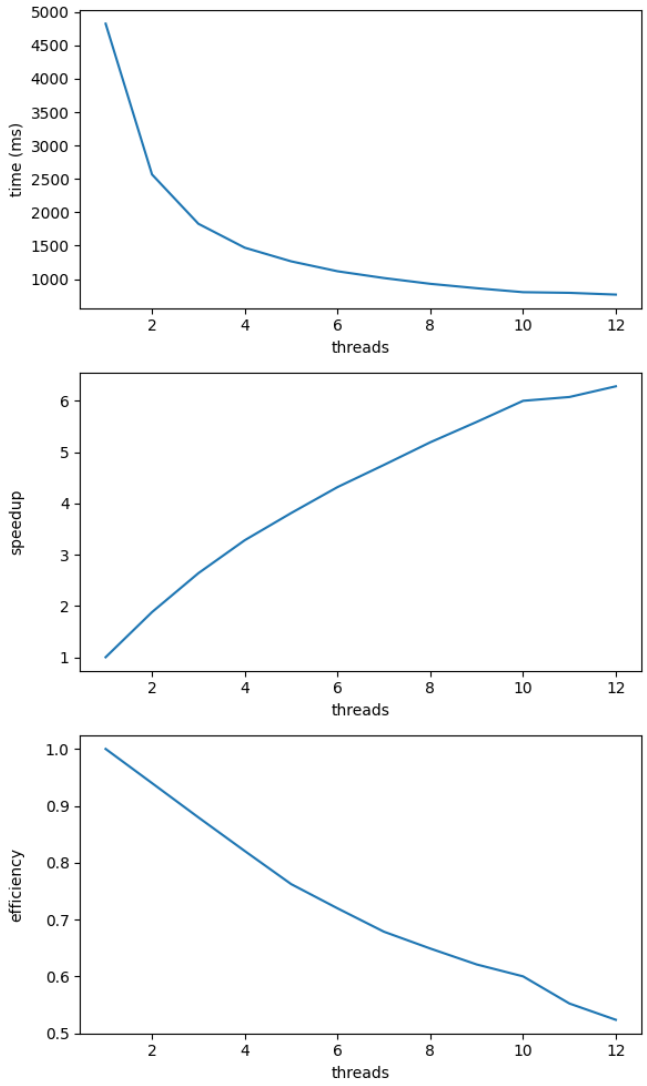


Figure 1: Timing, Speedup, and Efficiency

Table 1: Timing, Speedup, and Efficiency

| threads | time_ms | speedup | efficiency |
|---|---|---|---|
| 1 | 4825.110 | 1.000 | 1.000 |
| 2 | 2566.870 | 1.880 | 0.940 |
| 3 | 1828.590 | 2.639 | 0.880 |
| 4 | 1470.290 | 3.282 | 0.820 |
| 5 | 1266.030 | 3.811 | 0.762 |
| 6 | 1117.530 | 4.318 | 0.720 |
| 7 | 1015.900 | 4.750 | 0.679 |
| 8 | 929.464 | 5.191 | 0.649 |
| 9 | 863.428 | 5.588 | 0.621 |
| 10 | 804.230 | 6.000 | 0.600 |
| 11 | 794.370 | 6.074 | 0.552 |
| 12 | 768.030 | 6.282 | 0.524 |