Benchmark Report for KNN Parallelization Using MPI

Tom WALTER, Parsa VARES, Luc CARDOSO

November 24, 2024

1 Introduction

This report presents the results of benchmarking a K-Nearest Neighbors (KNN) classifier. For this, we used the provided KNNClassifier program and parallelized it.

2 Parallelization Approach

To parallelise KNN, we first need to understand what can and is worth parallelizing. In KNN, the training part is quite fast, and most of the time is consumed in the predictions. So, the goal is to parallelize the prediction part. For this, we used MPI, given that it was the latest topic in the class.

3 Benchmark Results

The benchmark results were measured in real-time (wall-clock time) and CPU time. To ensure statistical reliability, each run was executed 30 times for both the sequential and parallel versions. The average and standard deviation values for each metric were recorded.

3.1 Sequential Execution

The sequential execution results are as follows:

- **Real Time**: Average = 799.5091 seconds, Standard Deviation = 7.8738 seconds.
- **CPU Time**: Average = 790.9008 seconds, Standard Deviation = 7.7115 seconds.

3.2 Parallel Execution

The parallel execution results are as follows:

- **Real Time**: Average = 14.5060 seconds, Standard Deviation = 0.0328 seconds.
- **CPU Time**: Average = 14.3622 seconds, Standard Deviation = 0.0321 seconds.

4 Speed-Up Calculation

The speed-up gain is calculated as the ratio of the sequential real-time execution average to the parallel real-time execution average:

$$\text{Speed-Up} = \frac{\text{Real Time (Sequential)}}{\text{Real Time (Parallel)}} = \frac{799.5091\,\text{s}}{14.5060\,\text{s}} \approx 55.13$$

These results show that KNN scales well with more cores in this case we ran 64 cores and having a 55x is quite good scaling.

5 How to run

2 Slurm scripts will run the 2 Programs 1 Sequentially 30 runs and one for the Parallel version 30 runs. We also parallelized the sequential run to run the 30 runs simultaneously instead of one after another which would take 6.67 hrs.

- **Sequential**: run: sbatch script.sh out: knn_comparison-seq_parallel.out
- **Parallel**: run: sbatch scriptmpi.sh out: knn_comparison-mpi.out