CMSC 603 – High Performance Distributed Systems

Assignment 1: Threading — Virginia Commonwealth University, Fall 2018

Due date: September 23, 2018

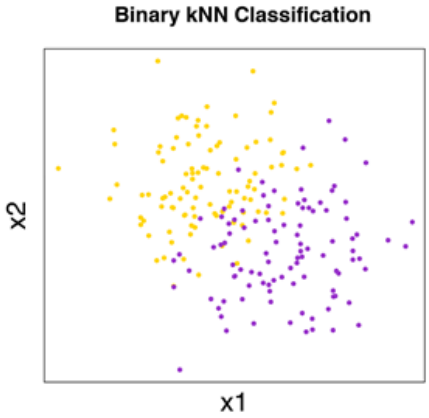
Big data mining involves the use of datasets with millions of instances. The computational complexity of machine learning methods limits their scalability to larger datasets. The simplest classifier is the nearest-neighbor classifier (NN) but its computational complexity is O(n^2), where n is the number of instances.

The KNN algorithm computes for each data instance the distances to the other instances and predicts the data class by majority voting from the k-nearest neighbors. The accuracy of the classifier is measured as the relation between the number of successful predictions and the number of instances. You may run the code using the datasets provided and analyze how the runtime increases with the size of the dataset.

The assignment consists on implementing the sequential and parallel the code using threads (C/C++/Java) and conduct all the code optimizations you consider relevant to speed up its execution, as long as the output is correct (accuracy of sequential and parallel must be the same). A skeleton is provided for the C version to facilitate the reading of the dataset file and accuracy calculation. The number of k should be defined by a parameter in the function. Should you consider critical sections in your code, be aware of data races and employ the tools to guarantee the mutual exclusion.

Execute both the sequential and parallel codes in your host machine and in maple.cs.vcu.edu to:

1. Compare the runtime of the sequential and parallel versions considering the datasets provided having different sizes. Calculate the speedup using 1, 2, 4, and 8 threads.
2. Estimate the proportion of parallel code by relating the speedup you get and the number of threads employed. What's the maximum speedup you would be able to obtain using an infinite number of threads and cores?
3. Force the code to run using 2048 threads. Recalculate the speedup. What's your conclusion?
4. Deliverables: 1) source code of the serial and sequential implementation.
   2) small documentation (1 page max) reporting the results obtained.



Remember: no code from previous years / internet may be employed here, your code must be your original contribution. Plagiarism will be reported.