CPSC 474 Project 2: Programming using MPI

Prof. Doina Bein, CSU Fullerton
dbein@fullerton.edu

Summary

For the project, a parallel implementation of the k-means problem was used. This was implemented using MPI and C++ . The document provides the problem, contributor, pseudocode, how to run the code, and screenshots.

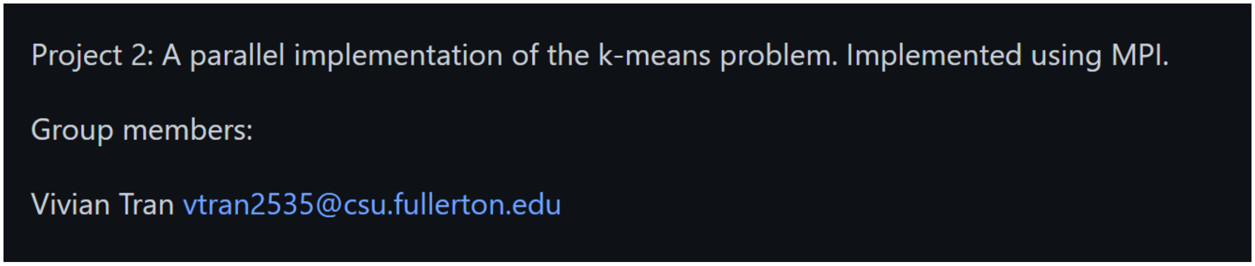
Problem

Given the PokemonCluster.csv dataset, we will separate the data into a number of K clusters and make observations

Contributor

Vivian Tran: vtran2535@csu.fullerton.edu

A full-screen screenshot with your group member names shown clearly. One way to make your names appear in Atom is to simply open your README.md.



Project 2: A parallel implementation of the k-means problem. Implemented using MPI.

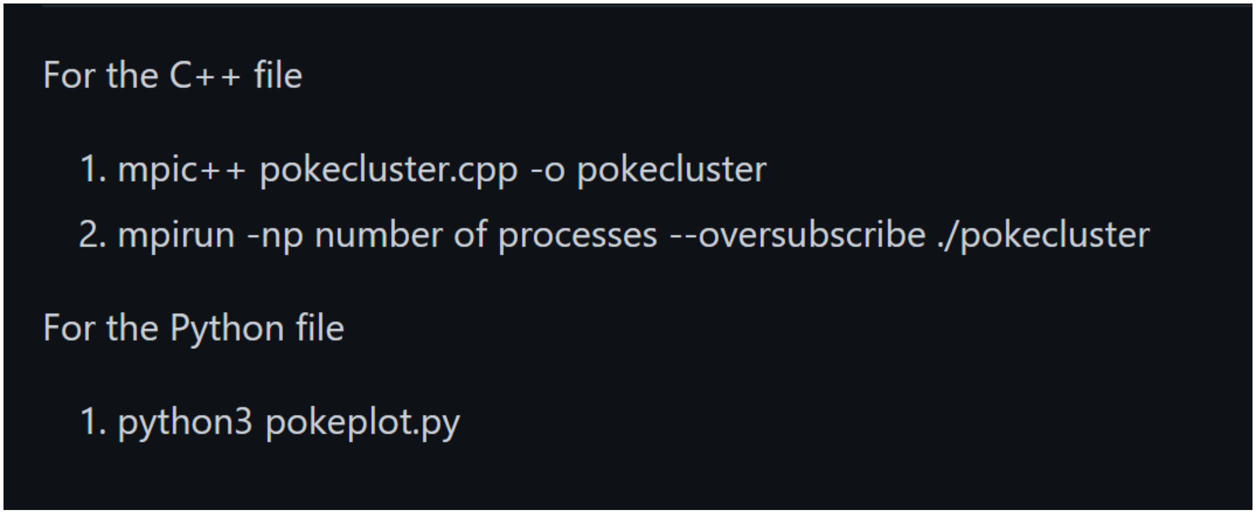
Group members:

Vivian Tran vtran2535@csu.fullerton.edu

Pseudocode
For C++ Code

```
String readFile()
{
        Variables for all the data from the PokedexCluster.csv file and will be stored
          string id;
          string name;
          string totalStat;
          string hpStat;
          string attackStat;
          string defenseStat;
          string spaStat;
          string spdStat;
          string speedStat;
          string genNum;

        Open file

        Use rand() to randomly select a pokemon id to be used as a initial centroid point
        Use a tempId in order to read the data as a string
}

Int main()
{


}

double eucledian_distance(const double **pointA, const double **pointB, int dim)
{
        Variables
        TempDistance//distance between point A and point B
        TotalDistance// distance after solved with power to 2

  for(int i = 0; i < dim; i++)
  {
    tempDistance = pointA[i] - pointB[i];
    totalDistance = pow(tempDistance,2);
  }

  return sqrt(totalDistance); //returns square of total
}
```

Brief description on how to run code



For the C++ file

1. mpic++ pokecluster.cpp -o pokecluster
2. mpirun -np number of processes --oversubscribe ./pokecluster

For the Python file

1. python3 pokeplot.py

Snapshots