Course: CS633 Parallel Computing - Assignment 3
Name: Aakrati Jain
Roll no.: 19111001
Programme: M.tech
Department: Computer Science and Engineering

Pseudo Code:

```
For each dataset
{
    For each p in P {2…12 at even intervals}
    {
        Each repeated 5 times
        {
            For each timestamp
            {
                1. Open the file corresponding to timestamp.
                2. Read the entire file at Rank 0.
                3. Calculate the data which will be uniformly distribute.
                4. Initialize the K-clusters randomly at Rank 0.
                5. Distribute the cluster information to all the processes.
                6. Each process finds the most appropriate cluster for its points using the
                Euclidean distance. Count the number of points which are assigned with
                different cluster than before.
                7. Each process calculates the number of its points in each cluster.
                8. Gather the information in previous step at Rank 0.
                9. Gather the total number of points which are assigned with different
                cluster than before. This information is used for converging the algorithm.
                10. Update the clusters according to the points assigned to it at Rank 0.
                11. Repeat Step 1. till 100 iterations or till the algorithm converges.
            }
        }
    }
}
```

Heuristics:

The *K*-means is unsupervised clustering algorithm where each point is assigned to the most appropriate cluster.

For getting the value of *K*, brute-force is applied and graphs corresponding to the various values of *K* were obtained. It was observed that values near 40 and 55 gave good results for the two datasets respectively. Increasing the value further did not give much improvement.

For converging/stopping the algorithm,

a) Number of points with change in value of cluster id were counted and if this value was smaller than 0.5% (c = convergence value) of total number of points
   OR
b) 100 iterations

Whichever is smaller is considered for convergence.

Data Decomposition method:
The entire file is first read at Rank 0 and then the data is uniformly distributed to each process for further computation. Example, if n = no_of_points and p = no_of_process, then roughly *n/p* points are distributed to each process.

Observations:
The code runs perfectly for larger number of processes as well though it is limited to 12 due to the large amount of time required for execution with increase in the number of processes. Through graphs it is observed that, with increase in the number of processes a significant drop is seen in total time (tt) while not much similar pattern is observed for pre-processing (ppt) and processing time (pt). This may indicate that the effect of parallelism is seen in *ppt* and *pt*, hence there decrease now depends on the bandwidth of the link and not much on the number of processes. While the *ttt* includes various other non-parallel elements introduced in the program by the programmer, hence the decrease depends on the number of processes. But this decrease will happen till a certain number of processes (saturation point) after which no more scalability could be obtained. The reason could be the increased amount of communication among processes.

Issues:

a) The execution of the entire program requires much time (approx. 2:30 to 3 hrs on CSE and 1:30 to 2 hrs on HPC and, for *P*= {2…12} at even intervals with ppn = 4, *K*=40 and 55 on dataset1 and dataset2 respectively, *c*=0.5%).
b) For getting the value of *K* algorithmically, elbow method was used but the curve obtained was similar to Independence curve (approx. y=1/x). Hence, no elbow or kink was noticed which otherwise would have given the best value of *K*.

Note:

1. You may encounter 'undefined reference to `sqrt'' during execution but that does not affect the results, so you may ignore.
2. If you change the value of p, the corresponding change in plotscript will be required.
3. Since the program requires much time for execution so in order to verify code change the number of *K* to a smaller value.
4. **Put the dataset files in datain1 and datain2 folders respectively.**
   cse → datain1 → file0 file1 … file16
   → datain2 → file0 file1 … file15
   and
   hpc → datain1 → file0 file1 … file16
   → datain2 → file0 file1 … file15