Assignment 3

Write a program to cluster moving particles in a 3D space. The particle locations are output of a simulation. The simulation output (per file) corresponds to a time step. You may use any clustering algorithm (e.g. K-means) to cluster the particles. You are required to find the number of clusters, the cluster means and size of each cluster for each time step. The input data is contained in binary files. Each file contains data corresponding to a time step. Number of files is the number of time steps. Each file contains NP lines, where NP is the total number of particles. Each line contains the id and coordinates (X, Y, Z) of particles, i.e. 4 doubles (32 bytes). Note that id is actually an integer, but it is stored as a double in the file. Number of particles (NP) may vary across files.

Your goal is to speed up the given problem using MPI. You need to demonstrate the scalability of your code. When using more than 1 process, particles may be divided among nodes based on any decomposition of your choice. You may read the data files in parallel. You may design any heuristic to read all the files. Run your code on 1, 2, 3, …, P processes; you may decide P (the max. no. of processes) based on the scalability of your code. You may select any number of nodes and any number of processes per node (ppn). However, maintain a constant ppn.

You are required to measure three times (individual and/or cumulative for all time steps):
1. Pre-processing time – Time to read + time to distribute data initially (if required)
2. Processing time – Time to cluster
3. Total time – Measure the total running time of your code (across all time steps)

Pseudo Code

```
For each p in P {

 Start timer for total time

     For each time step do
        Read data files + distribute particles (Measure pre-processing time per time step)
        Apply parallel clustering algorithm (Measure processing time per time step)

 End timer for total time

 Generate output in the given format (see below)

}
```

Output format:

```
------
Number of processes: P
T1: K1, <size1, mean1>, <size2, mean2>, …. <sizeL, meanL>  // say there were L clusters in T1
T2: ....
... till the last time step

Average time to pre-process:
Average time to process:
Total time:
------
```

(Here Ki is the number of clusters for ith time step, size is the number of particles in each cluster, mean is the cluster mean values (positions))

The above output must be present in a file called 'output_M.txt'. [M = {1,2, … P}]

You will be given two data sets. The first data set is the output of a real simulation; the second data set is a synthetic data set. Each contain some number of binary files (each file corresponding to a time step). The format of each file is given below:
<8 byte id1><8 byte x1><8 byte y1><8 byte z1><8 byte id2><8 byte x2><8 byte y2><8 byte z2> ..…. [All doubles]
Size of each file: (8 x 4 x #particles) bytes

For each data set, generate 1 plot file ('plot.png/jpg') with 3 plots (average pre-processing time across time steps, average processing time across time steps, total time) for each process size. Number of processes in x-axis and times in y-axis. The plots should be box plots from 5 runs of each process size.

You must also submit a report (report.pdf) that contains the pseudo code (for your implemented algorithm), brief explanation about the heuristic used, data decomposition method and observations regarding results (e.g., scalability of your code, processing times, pre-processing times etc.). You can add your observations regarding performance/speedup, analysis of your results, code description, code design decisions and any issues that you might have faced.

You are required to execute on CSE cluster and the HPC2010 system. (40 + 40 marks)

General submission instructions

Submission folder (named Assignment3) should necessarily contain the source code ('src.c'), 'Makefile', 'report.pdf', CSE cluster job script ('run.py' or 'run.sh'), HPC2010 job script ('sub.py' or 'sub.sh'), plot script. The job script should run all the configurations, i.e. I should be able to run the respective job scripts to execute your code (all configurations). You may have other auxiliary files, if required. You may not upload the input data files to the repo. The files corresponding to the CSE cluster must be present in a sub-directory ‘cse’, and the files corresponding to the HPC cluster must be present in a sub-directory ‘hpc’. You are given two data sets. Your job script must run both the data sets. Your plot files and output files must be present under sub-sub-directories ‘data1’ and ‘data2’ for each data set.

```
|----cse
        |-----data1
        |-----data2
|----hpc
        |-----data1
        |-----data2
```

General instructions

The entire programming must be done in C. Python may be used only for plotting, job script and hostfile generation.

For CSE cluster, its necessary to have a script that generates hostfile on-the-fly based on the node status so that your jobs never fail. Hostfiles used must NOT be hardcoded and must be from your set S (given at the end). If at any point, you are unable to get the requisite number of nodes for 2 consecutive days, you must inform me and the TAs.

It is recommended to first test your code on CSE cluster and take regular backup on git, before testing on HPC2010.

Other components (20 marks)

- Follow the above instructions carefully (especially naming conventions)
- Use git.cse.iitk.ac.in as a real git repo (i.e. marks will be awarded on progress)
- Neat coding, documented code
- Fully automated execution of all configurations

Due date: 10-11-2019 (There will be NO extensions)

Set S based on your CC usernames (institute email-ids).

A - C

csews1, csews2, csews3, csews4, csews5, csews13, csews17, csews18, csews19, csews20, csews33, csews34, csews35, csews36, csews45, csews47, csews48, csews49, csews50, csews62, csews63, csews64, csews65, csews66, csews79, csews80, csews81, csews82, csews93, csews94, csews95, csews96, csews97, csews109, csews110, csews111, csews112.

D - P

csews6, csews7, csews8, csews9, csews10, csews21, csews22, csews23, csews24, csews25, csews37, csews38, csews39, csews40, csews51, csews52, csews53, csews54, csews55, csews67, csews68, csews69, csews70, csews71, csews83, csews84, csews85, csews86, csews98, csews99, csews100, csews101, csews102, csews113, csews114, csews115, csews116.

Q - Z

csews11, csews12, csews14, csews15, csews16, csews26, csews27, csews28, csews29, csews30, csews41, csews42, csews43, csews44, csews46, csews56, csews58, csews59, csews60, csews72, csews73, csews74, csews75, csews76, csews87, csews88, csews89, csews90, csews103, csews104, csews105, csews106, csews107, csews117, csews118, csews119, csews120.