Parallel k-means clustering of stars from the SLOAN digital sky survey.

You need to:

Download a set of star files from MilkyWay@Home:

http://milkyway.cs.rpi.edu/milkyway/download/stars.zip

The format of the files is:

the first line will contain the number of stars in the file.

the remainer of the file are star points (three values per star per line). These are using galactic coordinate system, with the sun as the center (http://en.wikipedia.org/wiki/Galactic_coordinate_system). You'll want to convert these to cartesian coordinates (X Y Z), which is fairly straightforward as they're essentially spherical coordinates: http://en.wikipedia.org/wiki/Spherical_coordinate_system . Note that the L and B are in degrees, and they'll need to be convert to radians as well.

You need to:

1. (10%) write a program called kmeans which takes for the first argument the number of clusters, and then all the remaining arguments will be star files (your program should be able to read in multiple star files, as this will help in testing), eg:

./kmeans 5 stars-9.txt stars-11.txt stars-82.txt

./kmeans 10 stars*.txt

2. (20%) This program should read in all the stars from all the files into a single dimensional array (this will make your life easier later) of doubles, eg:

double* stars = new double[number_of_stars * 3];

where, stars[0], stars[1], stars[2] will be the x y z of the first star; stars[303], stars[304], stars[305] will be the x y z of the 101st star, etc.

All the stars should go into this array.

3. (30%) Implement non-parallel k-means clustering using Lloyd's algorithm: http://en.wikipedia.org/wiki/K-means_clustering

You should use this to verify the results of your parallel implementation. You should also be able to test it easily using the star files. Each star file comes from a different section of the sky, so if you run ./kmeans 3 stars-9.txt stars-15.txt stars-82.txt, it should put all the stars in the first file in one cluster, all the stars in the second file in another cluster, etc. You should also generate simple test files to test your program on as well.

4. (50%) Implement a parallel k-means clustering with MPI based on Lloyd's algorithm. You should;

NAVIGATION

Home
My home
Site pages
My profile
Current course
CS532
Participants
General
22 August - 28 August
29 August - 4 September
5 September - 11 September
MPI Programming Assignment 1
12 September - 18 September
19 September - 25 September
26 September - 2 October
3 October - 9 October
10 October - 16 October
17 October - 23 October
24 October - 30 October
31 October - 6 November
7 November - 13 November
14 November - 20 November
21 November - 27 November
28 November - 4 December
5 December - 11 December
12 December - 18 December
19 December - 25 December
My courses

SETTINGS

Course administration
My profile settings

a. (10%) start with k random centroids (you get k from the first command line argument); generate their x y z randomly within the bounds of the stars (you should be able to calculate the min and max values of x/y/z easily).

Only the master should calculate the centroids and it should use the MPI_Broadcast function to broadcast them to the other processes.

b. (10%) have each MPI process work on a subset of stars; eg., if there are N stars, and 4 processes, process 1 will work on stars 0 ... (N/4)-1, process 2 will work on stars n/4...(2N/4)-1, process 3 will work on stars (2N/4)...(3N/4)-1, and process 4 will work on stars (3N/4)...N. You need to be able to handle having a number of stars that doesn't divide evenly over the number of processes (some processes might need to have an extra star).

Only the master process should read the file, and it should send the other processes their stars using the MPI_Scatterv function.

c. (10%) each of these processes should have an array called assignments which the processes will store the cluster assignments for their subset of stars; eg., if star 32 is assigned to centroid 5, assignments[32] will be 5. After each process calculates the new centroid for the stars in their subset, all these stars should be gathered to a single array on the master process using the MPI_Gatherv function.

d. (10%) the master process should calculate a new set of centroids from the new assignments; and then broadcast these back out to all the other processes; then repeat from b. This process should repeat until the newly calculated centroids are the same as the previous centroids.

Your program should output (at each iteration) what the centroids are, and then the final set of centroids.

BONUS: (20%) Have the fastest k-means clusterer in the class.

BONUS: (10%) Have the 2nd fastest k-means clusterer in the class.

(The bonus will most likely include parallelizing the calculation of the centroids).

You should upload all your code (with a readme of instructions on how to compile and run it) to moodle as a zip file named <your_last_name>_kmeans.zip (a tarball is fine as well).

You can use the following program to visualize and validate your results:

https://github.com/travisdesell/star_visualizer/

In a terminal (assuming git is installed) you can do:

git clone https://github.com/travisdesell/star_visualizer.git

which will download all the star_visualizer code into a directory called 'star_visualizer'

then you can follow the compilation instructions.

You'll to have installed OpenGL and git:

sudo apt-get install git

sudo apt-get install opengl

On the lab machines you can run:

$git clone https://github.com/travisdesell/star_visualizer/

This will download the star visualizer code into your current working directory in a sub directory called "star_visualizer"

$cd star_visualizer

$mkdir build

$cd build

$g++ ../star_visualizer.cxx -o star_visualizer -lGL -lglut -lGLU

The above compiles the star visualizer code in the lab.

You can the ndownload the star files with:

$cd ..

$wget http://milkyway.cs.rpi.edu/milkyway/download/stars.zip

Which will download the zip file. You can then unzip it:

$unzip stars.zip

Which will put all the star files in the stars subdirectory.

You can then run the star visualizer with the following:

$cd build

$.star_visualizer --window_size 500 500 --star_files ../stars/*.txt --modulo 10

Available from:	Thursday, 28 January 2016, 8:00 AM
Due date:	Thursday, 21 September 2017, 11:55 PM

Upload a file