LAB1. K-MEANS PARALLELIZATION in PYTHON

Creation of the datasets

In the lab material you would find a file named “proteins-generator.py”. You have to use it to generate proteins datasets for the lab. To generate a data set, execute the command:

$>python proteins-generator.py numrows

Being “numrows” a parameter specifying the number of protein chains in the dataset.

For development:
$>python proteins-generator.py 50000

To test performance of the solution to deliver a least 2,000,000 rows, but you can test as many as you want:

$>python proteins-generator.py 2000000

The file "proteins.csv", created include a data set about a list of proteins, including the following information per protein:

"protid”,”enzyme”,”hydrofob”,”sequence"

IMPORTANT: Do not modify, touch the file or create transformed fields. For the lab delivery extra files will not be accepted. We will use the same command to generate the dataset.

Notice: In the data you have 1 field that is not numerical: *sequence*. For working with numbers, when needed you can use sequence length to normalize data.

Laboratory Description

You are asked to extract useful information the proteins data set implementing a program using the k-means algorithm in Python.

uc3m | Universidad Carlos III de Madrid

Master in Big Data

TECHNOLOGICAL FUNDAMENTALS IN THE BIG DATA WORLD

Use the path "*proteins.csv*". for the file. Do not include the full path in your computer (e.g things like C:\mifolder\proteins.csv, or ../tmp/proteins.csv are forbidden). You have to execute the python program in the directory where the file is.

Part one – Python serial

1.- Implement the k-means algorithm for clustering using Enzyme and Hydrofob fields.

- Use euclidean distance

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

- Random centroids at the beginning

2.- Using that implementation, write a program that:

- Register start time
- Read the csv file
- Construct the elbow graph and find the optimal clusters number (k).
- Cluster the data using the optimum value using k-means.
- Find the cluster with the highest sequence length and compute its average sequence length.
- Measure end time and print execution time (end-start)
- Plot results of the execution:
  - Elbow graph.
  - Clusters with centroids.
  - Heat map using the values of the clusters' centroids.
  - For the cluster with highest sequence length, print id, highest value, average sequence length previously computed.

`Important: Figures of the plot must be created without stopping the execution phase and printed at the end.`

Part two – Python parallel, multiprocessing

1.- Write a parallel version of you program using multiprocessing.

2.- Measure the time and optimize the program to get the fastest version you can.

3.-Measure time and print it

4.- Plot the results as in the serial version.

Part three – Python parallel, threading

1.- Write a parallel version of you program using threads

2.- Measure the time and optimize the program to get the fastest version you can.

3.-Measure time and print it

4.- Plot the results as in the serial version.

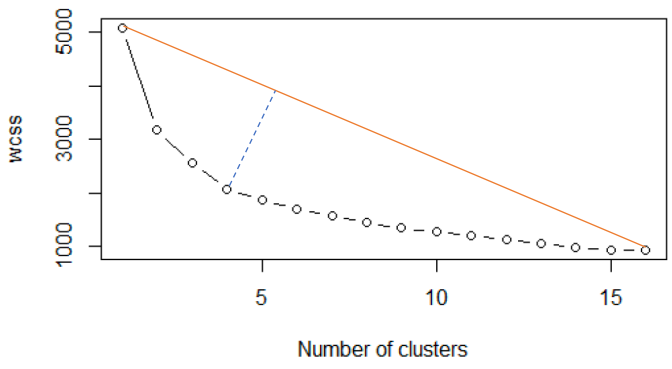
Part four

1. - Write a memory explaining your results (maximum 12 pages).
- Important: show measures with speedup of the parallel versions against serial.
- Show possible speedup using Amdahl Law.

Important

- To correctly measure time, you cannot stop the program to see the graphs (e.g. k-means). You have to keep all the graphs and print them at the end, so as the messages in the screen.
- Computing the Optimum K can be done using the elbow algorithm and computing the interval (k1,K2) using the slope of the curve or the maximum distant from points to a straight line between extreme points. For example in the picture a possible interval would be [3,6].

Lab1



uc3m | Universidad Carlos III de Madrid

Master in Big Data

TECHNOLOGICAL FUNDAMENTALS IN THE BIG DATA WORLD

Laboratory Delivery

Maximum group: 4 people.

Do not use deliver Jupyter notebooks. Deliver .py files.

You have to deliver a compressed PER GROUP file named:
"StudentNIA_proteins_2024.zip" (e.g. 100023456_proteins_2024.zip) including:

- A PDF report with the memory (include author names)
- "authors.txt" file, including a line per author (NIA, SURNAMES, NAME)
- Three Python programs with serial and parallel versions of the program with multiprocessing. Names:
  - lab1-proteins-serial.py
  - lab1-proteins-mp.py
  - lab1-Proteins-th.py

Delivery date: October 6th 2025. 23:30 hours.