uc3m | Universidad Carlos III de Madrid

Master in Big Data

TECHNOLOGICAL FUNDAMENTALS IN THE BIG DATA WORLD

LAB1. PROTEIN MATCHING in PYTHON

Creation of the datasets

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

$>python3 proteins-generator.py numrows

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

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

"id","Sequence"

The Sequence is the description of the protein components expressed as a chain of chars, where you have to look for matches.

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.

Laboratory Description

You are asked to create a program to match a pattern introduced using the keyboard against all the proteins in the dataset using Python.

Part zero

Create a dataset “proteins.csv” with 500,000 lines to be used for final tests in the lab.

Note: for development we advise you to create something smaller (e.g. 50,000 lines).

Part one

Make a serial program in Python that:

1. Reads the pattern to search from the keyboard.
2. Changes the pattern to UPPERCASE
3. Start execution time
4. Reads the protein patterns from the file in pairs (id, sequence)
5. Look for occurrences of the pattern string inside each protein sequence
6. If there are occurrences, register the id of the protein and the number of occurrences in the sequence
7. Show the execution time
8. Print a histogram of occurrences using protein id as X and number of ocurrences as Y, using matplotlib.pyplot. Represent the 10 proteins with more matches.
9. Print the id and number of the protein with max occurrences.

Part two

8.- Write a parallel version of your program using **multiprocessing**. Optimize the program to get the fastest version you can.

9.- Repeat execution, measuring the time before printing the histogram and the protein with max occurrences.
11.- Show the speedup compared with the serial version.

Part three

10.- Write a parallel version of you program using **threads.** Optimize the program to get the fastest version you can.

11.- Measure the time before printing the histogram and optimize the program to get the fastest version you can.
12.- Show the speedup compared with the serial version.

Part four

12.- Write a memory explaining your results (maximum 12 pages). Explain which technique is better.

Laboratory Delivery

Groups of 3 students.

uc3m | Universidad Carlos III de Madrid

Master in Big Data

TECHNOLOGICAL FUNDAMENTALS IN THE BIG DATA WORLD

You have to deliver a compressed file named: "yournia_lab1_2022.zip" including:

- Report with the memory in PDF (include author names)
- Python program with the serial version of the program.
  - Name: "serial-proteins.py".
- Python program with the parallel version of the program with multiprocessing.
  - Name: "mp-proteins.py".
- Python program with the parallel version of the program with threads.
  - Name: "th-proteins.py".

Delivery date: October 18th 2022. 23:30 hours.

DO NOT deliver Jupyter notebooks. Just Python scripts.