CPrE/SE 419: SOFTWARE TOOLS FOR LARGE-SCALE DATA ANALYSIS, SPRING 2019

LAB 7: SPARK #2

Purpose

The main objective of this lab is to solidify the experience with Spark – and you are required to write programs that will have to pipeline the activities/jobs. Specifically, you will write a code for jobs that will:

- Analyze GitHub data
- Analyze Graph data

(Note: This lab should still be run on your local/lab machine)

Submission

Create a single zip archive with the following and hand it in through Canvas:

- The output file for each task generated by your program.
- Commented Code for your program. Include all source files needed for compilation.

Experiment 1 (40 points)

Our data is "github.csv" on the HDFS at: /cpre419/github.csv, where you can copy from to your local

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	repository	language	architecture	community	continuous_i	document	history	license	managem	size	unit_test	state	stars
2	matplotlib/matplotlib.github.com	Python	0.770463	2	0	0.014931	2.297872	0	0.212766	1575488	0.013242	active	5
3	NCIP/c3pr-docs	Java	0.997449	3	0	0.087444	1.434211	0	0	765164	0	dormant	0
4	AnXgotta/Sur	C++	0.714286	1	0	0.123698	0	0	0	2155	0	dormant	0
5	bigloupe/SoS-JobScheduler	Java	0.957573	3	1	0.315557	11.42857	1	0	657960	0.007257	None	1
6	barons/zf_shop	Ruby	0.381323	3	0	0.327179	0	1	0	472610	0.055335	None	0
7	uzleo/hiwi	C++	0.865123	2	0	0.218128	15.8	1	0	170144	0.011772	None	0
8	berlinonline/banned_books	PHP	0.44	4	0	0.017882	5	1	0	399320	0	None	0
9	pszabolcs/canvasandroid	Java	0.988235	4	0	0.136708	32.66667	0	0	119414	0	None	0
10	mk12/mycraft	Java	0.662614	1	0	0.326084	3.583333	1	0	134913	0.117074	None	7
11	BulldogDrummond/etmod	C	0.820513	1	0	0.085501	0	0	0	220996	0.00994	None	0
12	ryseto/stodyn	C++	0.943548	1	0	0.14322	0	1	0	228026	0	None	0
13	UfSoft/ILog	Python	0.666667	1	0	0.186233	14.16667	0	0	6004	0	None	0
14	nix858/osu	C++	1	1	0	0.18365	0	0	0	3648	0	None	0
15	WilbertHo/foobar	Python	0.705882	2	0	0.320261	4.666667	0	0	210	0.251282	None	0
16	kaludis/epoll-echo-server	C	1	1	0	0.36478	0	0	0	303	0.090592	None	0
17	Jarcionek/MTG-Deck-Builder	Java	0.983051	1	0	0.014853	0	0	0	3051	0.514811	None	0

For each language, find out how many repositories using it, one repository that has the highest stars number.

In this experiment, the job is to generate a list with the following format:

<language> <num_of_repo> <name_of_repo_highest_star> <num_stars>

num_of_repo	total number of projects in GitHub using a specific language.
name_of_repo_highest_star	name of a repository that has highest stars number.
num_starts	number of stars of the repository that has highest starts number.

This list should be sorted by the num_of_repo in descending order.

Experiment 2 (60 points)

This is somewhat of a "Déjà vu" – but now you will think of the solution with a specific pipelining and RDDs on mind… A graph G= (V,E) consists of a set of vertices V, and a set of edges E such that each element e in E is an pair (u,v), denoting an edge between u and v. In a undirected graph, a cycle of length three is a triple of vertices (x,y,z) such that eaches of (x,y) (y,z) and (z,x) exist in E.

Write a program that calculate number of all undirected cycles of length 3 in a graph.

We use the dataset "patents": /cpre419/patents.txt, where you can copy from, to your local.
The graph is in the form of an edge list. Every line of the file has information about a single edge. A line contains information in the format <vertex id 1> <vertex id 2>, which means that it's an edge between those 2 vertices.