Project 1

Every student should submit the code (`studentID_name.zip`), report (`studentID_Name.pdf`), and ppt (`studentID_name.ppt`) on Blackboard system before 23:59 April 21. Report can be written either in English or Chinese.

Q1 Security Data Analysis

MACCDC 2012 provides operational data of managing and protecting an existing network infrastructure. The data is also available here.

(1) [2 points] Create **two** Spark dataframes (`df_http` `df_dns`) from the files `http.log.gz` `dns.log.gz` in folders `00` to `05` (six folders in total). Convert the `ts` column to Timestamp data type. Create two temp view named `http_log` and `dns_log` .

(2) [2 points] With the http log data, filter the rows where the status code is 200 and method is GET, sort in a descending order according to the accessed count of the `uri` . Use Spark SQL API and Spark dataframe, seperately.

(3) [3 points] Use Spark SQL to join the `http_log` and `dns_log` tables by `uid` , and calculate the percentage of `proto=tcp` for each `uri` group found in taks (2).

(4) [2 points] Use Spark dataframe to calculate the percentage of different `method` in the http log. Also display the pie chart of different `status code` for each `method` .

Q2 Document Analysis

Paul Graham is an English computer scientist, essayist, entrepreneur, investor, and author. He is best known for his work on the programming language Lisp, co-founding the influential startup accelerator and seed capital firm Y Combinator, and Hacker News.

(1) [3 points] Crawl the articles of Paul Graham and store the text of the articles in folder `paul_articles`. Each article is located in a seperate `txt` file. You can use Scrapy or any other tools you like. You can refer to the official guidelines of Scrapy here.

(2) [3 points] Create a dataframe by reading from these txt files with pyspark. Each row only contains the sentences in one paragraph. Select paragraphs that are related to suggestions of career planning. You can read some examples to determine some key phrases or regrex expressions for filtering. Store all the filtered paragraphs in a parquet file `career_suggestions.parquet`. (This is an open task that different answers are allowed.)

(3) [4 points] Extract noun phrases of all articles with Spark user-defined-functions and count their frequencies. You can use the Spacy Package. Plot the word cloud map with wordcloud package for the noun phrases which have the top 40~50 highest frequencies (including both end).

Note: If you fail to crawl the text data, you can manually copy paste some articles and store in different files. In this case, you can still earn the scores for task (2) and task (3).

Presentation

[6 points] Sampled students will present their work and thoughts within 10 minutes.