**LAB 3**

***Xử lý dữ liệu văn bản với PySpark***

## **Khởi tạo ứng dụng PySpark**

Mỗi ứng dụng Spark được quản lý bởi một trình điều khiển (driver). Trình điều khiển chứa hàm main() của trình ứng dụng, tạo dữ liệu phân tán trên cụm máy tính rồi thực hiện các xử lý trên chúng. Trình điều khiển kết nối với Spark thông qua một đối tượng của lớp SparkContext.

***# Khai báo thư viện***

***import os, shutil***

***from pyspark import SparkContext***

***# tạo đối tượng Spark context***

***sc = SparkContext("local", "Text processing with PySpark")***

## **Tạo RDD**

* Spark thực hiện các xử lý trên một cấu trúc gọi là resilient distributed datasets (RDDs). Mỗi RDD được chia thành các phân vùng (partitions), mỗi phân vùng có thể được xử lý trên các trạm khác nhau trên cụm máy tính. RDD có thể chứa kiểu dữ liệu của Python, Java, Scala, hoặc đối tượng của lớp tự định nghĩa. Các xử lý của Spark đều xoay quanh RDD: tạo mới, biến đổi RDD, tính toán kết quả từ RDD.
* Sau khi tạo đối tượng điều khiển SparkContext, cần tạo RDD để tiến hành các xử lý. Một cách thông dụng là tạo RDD từ tập dữ liệu sẵn có trên bộ nhớ ngoài. Ví dụ, lệnh sau đây tạo RDD từ dữ liệu văn bản trên bộ nhớ ngoài:

***# Đọc dữ liệu vào RDD từ tệp văn bản***

***lines = sc.textFile("input.txt")***

* Nếu thực hiện thành công, biến lines là một đối tượng RDD chứa tất cả các dòng văn bản của tập dữ liệu. Để tách mỗi dòng thành các từ riêng biệt, cần gọi phương thức biến đổi (transformation) có tên là flatMap(). Phương thức flatMap() gọi hàm do người dùng định nghĩa lên các phần tử của RDD và trả về một RDD chứa các phần tử kết quả của lời gọi hàm. Ví dụ: để tách mỗi dòng văn bản thành các từ riêng biệt, có thể định nghĩa hàm Python như sau:

***# split the lines into separated words***

***words = lines.flatMap(lambda line: line.split(" "))***

## **Xử lý trên RDD**

* Các xử lý trên RDDs được chia làm hai loại: biến đổi (transformations) và hành động (actions). Thao tác biến đổi thực hiện xử lý trên RDDs và trả về một RDD mới, chẳng hạn như map(), filter(). Thao tác hành động xử lý RDD và trả về kết quả cho trình điều khiển hoặc lưu lên bộ nhớ ngoài, chẳng hạn count() hay take().
* Ví dụ: để đếm tần số của mỗi từ trên tập dữ liệu, trước hết sử dụng phương thức map() để biến đổi mỗi từ thành một cặp <key, value> = <word, 1>. Sau đó gọi phương thức reduceByKey() để thực hiện gộp kết quả theo qui tắc được định nghĩa trong hàm truyền cho nó.
* Ở đây hàm

***lambda a, b: a + b***

nhận vào hai đối số và trả về tổng của chúng. Trong trường hợp này, phương thức reduceByKey() sẽ gộp theo từ và trả về tổng số đếm của mỗi cặp <word, 1>.

***# count the occurrence of each word***

***wordFrequencies = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b: a + b)***

* Trong trường hợp muốn đếm tổng số từ trong tập dữ liệu, có thể thực hiện như sau:

***# count total number of words in the dataset***

***totalWordNumber = words.map(lambda word: 1).reduce(lambda a,b: a + b)***

***# display results***

***# print the number of lines***

***print('\nTotal number of lines:', lines.count())***

***# print total number of words in the dataset***

***print('\nTotal number of words:', totalWordNumber)***

***# lấy ra 10 từ***

***someResults = wordFrequencies.take(10)***

***print("\nSome results:")***

***print(someResults)***

***# In ra top 10 từ xuất hiện nhiều nhất:***

***topFrequentWords = wordFrequencies.takeOrdered(10, key = lambda x: -x[1])***

***print("\nTop frequent words:")***

***print(topFrequentWords)***

***Lưu kết quả vào một tệp văn bản***

***# save the set of <word, frequency> to disk***

***savingPath = “file:///C:/spark/spark-3.5.1-bin-hadoop3/output"***

***if os.path.isdir(savingPath):***

***shutil.rmtree(savingPath, ignore\_errors=True)***

***wordFrequencies.saveAsTextFile(savingPath)***

***Đóng SparkContext***

***sc.stop()***

## **Bài tập:** *X*ử lý dữ liệu Twitter

Cho file văn bản elonmusk\_tweets.csv chứa các dòng tweets của Elon Musk từ 2011-2017. Dữ liệu được chia sẻ bởi Adam Helsinger. Từ file dữ liệu trên, hãy thực hiện Viết chương trình lab3.py thực hiện các xử lý sau với PySpark:

* Liệt kê 20 từ được nhắc đến nhiều nhất.
* Liệt kê 10 tài khoản được nhắc đến nhiều nhất.

Sau đó thực thi các chương trình bằng spark-submit như hướng dẫn ở lab 2.