НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизації проектування енергетичних процесів і систем

Лабораторна робота № 2

З дисципліни: «Оброблення надвеликих масивів даних»

**Виконав:**

студент 5 курсу

групи ТР-02мп, ТЕФ

Круглий Д.В.

**Перевірила:**

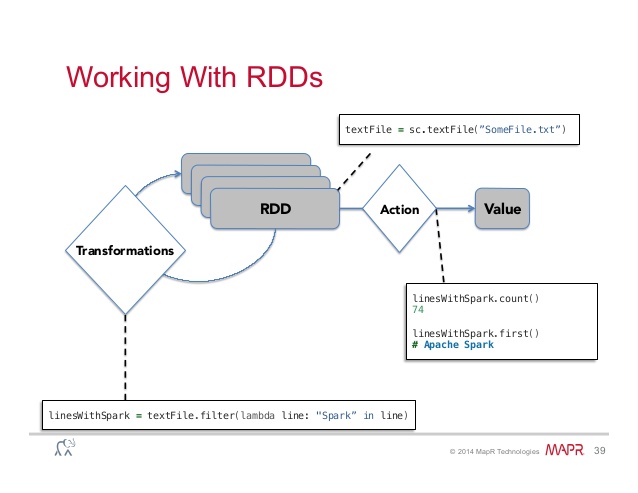
проф. Федорова Н. В.

Київ – 2020

**Тема:** застосування SPARKSQL, робота з даними з використанням Dataframes та Dataset.

**Мета:** навчитися працювати з даними з використанням Dataframes та Dataset, а також застосовувати SPARKSQL на практиці.

**Теоретичні відомості**



Spark підтримує Scala, Python і Java. Основним поняттям в Spark'е є RDD (Resilient Distributed Dataset), який представляє собою Dataset, над яким можна робити перетворення двох типів (і, відповідно, вся робота з цими структурами полягає в послідовності цих двох дій).

Дії застосовуються тоді, коли необхідно матеріалізувати результат - як правило, зберегти дані на диск, або вивести частину даних в консоль. Ось список найпоширеніших дій, які можна застосовувати над RDD:

• **saveAsTextFile (path)** - зберігає дані в текстовий файл (в hdfs, на локальну машину або в будь-яку іншу підтримувану файлову систему - повний список можна подивитися в документації);

• **collect ()** - повертає елементи датасета у вигляді масиву. Як правило, це застосовується у випадках, коли даних в датасета вже мало (застосовані різні фільтри і перетворення) - і необхідна візуалізація, або додатковий аналіз даних, наприклад засобами пакета Pandas;

• **take (n)** - повертає у вигляді масиву перші n елементів датасета

• **count ()** - повертає кількість елементів в датасета;

• **reduce (function)** - знайома операція для тих, хто знайомий з MapReduce.

З механізму цієї операції слід, що функція function (яка приймає на вхід 2 аргументу повертає одне значення) повинна бути обов'язково комутативна і асоціативна.

Дана лабораторна робота була виконана з використанням мови програмування Python.

**Хід роботи**

1. Імпортуємо необхідні пакети для роботи з даними:

import pandas as pd import matplotlib.pyplot as plt plt.style.use('fivethirtyeight')

1. Так як в завданні лабораторної роботи не було вказано, звідки брати дані, то було прийнято рішення взяти їх з інтернету:

|  |
| --- |
| import urllib.request plt.style.use('fivethirtyeight')  urllib.request.urlretrieve("http://kdd.ics.uci.edu/databases/kddcup99/kddc up.data\_10\_percent.gz", "/tmp/kddcup\_data.gz") dbutils.fs.mv("file:/tmp/kddcup\_data.gz", "dbfs:/kdd/kddcup\_data.gz") |

1. Наступним кроком будуємо Resilient Distributed Dataset:

data\_file = "dbfs:/kdd/kddcup\_data.gz" raw\_rdd = sc.textFile(data\_file).cache()

1. Будуємо Spark Dataframe для наших даних:

from pyspark.sql import SQLContext sqlContext = SQLContext(sc)

1. Проводимо розділ CSV даних:

csv\_rdd = raw\_rdd.map(lambda row: row.split(","))

1. Виводимо метадані:

len(csv\_rdd.take(1)[0])

Out[57]: 42

1. Витягуємо колонки з даними:

from pyspark.sql import Row

parsed\_rdd = csv\_rdd.map(lambda r: Row( duration=int(r[0]), protocol\_type=r[1], service=r[2], flag=r[3], src\_bytes=int(r[4]), dst\_bytes=int(r[5]), wrong\_fragment=int(r[7]), urgent=int(r[8]), hot=int(r[9]),

num\_failed\_logins=int(r[10]), num\_compromised=int(r[12]), su\_attempted=r[14], num\_root=int(r[15]), num\_file\_creations=int(r[16]), label=r[-1]

)

)

1. Будуємо Dataframe:

df = sqlContext.createDataFrame(parsed\_rdd)

1. Створюємо тимчасову таблицю:

df.registerTempTable("connections")

1. Виводимо протоколи:

protocols = sqlContext.sql("""

SELECT protocol\_type, count(\*) as freq

FROM connections

GROUP BY protocol\_type

ORDER BY 2 DESC """)

**Висновок:** під час виконання лабораторної роботи було отримано навички роботи з платформою Apache Spark. Розробка програми велась на мові Python та з використанням API Spark. Було здобуто навички роботи з датасетом RDD, а також з інструментом Spark SQL. Було побудовано RDD з використанням різних трансформацій та реалізовано пошук по ньому. Було побудовано тимчасову SQL таблицю та реалізовано пошук в ній.