**Лабораторная работа 1. Установка и настройка распределенной системы. Простейшие операции и знакомство с функциональностью системы.**

**Цель:** ознакомление с процессом установки и настройки распределенных систем, таких как Arenadata Hadoop или Apache Spark. Изучить основные операции и функциональные возможности системы, что позволит понять принципы работы с данными и распределенными вычислениями.

**Задачи:**

1. Установить распределенную систему на базовой версии Ubuntu.
2. Настроить систему для работы в распределенном режиме.
3. Выполнить базовые операции для проверки работоспособности системы.
4. Ознакомиться с основными функциями и возможностями системы.
5. Скачайте датасет "Tech Stocks Daily Prices" с Kaggle:
   https://www.kaggle.com/datasets/dgawlik/nyse
6. Загрузите данные в Spark и выполните следующие операции:
   a) Отфильтруйте данные только для компании Apple (AAPL).
   b) Рассчитайте среднюю цену закрытия акций Apple за каждый месяц.
   c) Найдите максимальную и минимальную цену акций Apple за весь период.
   d) Визуализируйте динамику цены закрытия акций Apple.
7. Напишите краткий отчет о проделанной работе и полученных результатах.

**Ход работы**

Система: Ubuntu 20.04

1. Установка Java Development Kit (JDK).

```
sudo apt update
sudo apt install openjdk-11-jdk -y
java -version
```

2. Загрузка и установка Apache Spark.

```
cd /opt
wget https://downloads.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz
tar xvf spark-3.2.1-bin-hadoop3.2.tgz
sudo mv spark-3.2.1-bin-hadoop3.2 /opt/spark
```

3. Настройка переменных среды: Добавьте следующие строки в файл .bashrc:

```
echo "export SPARK_HOME=/opt/spark" >> ~/.bashrc
echo "export PATH=$PATH:$SPARK_HOME/bin" >> ~/.bashrc
```

4. Примените изменения.

```
source ~/.bashrc
```

5. Скачивание датасета:

```
wget https://www.kaggle.com/datasets/dgawlik/nyse/download/stocks.csv

или
mkdir -p ~/data
cd ~/data
wget
https://query1.finance.yahoo.com/v7/finance/download/AAPL?period1=0&period2=9999999
999&interval=1d&events=history&includeAdjustedClose=true -O AAPL.csv
```

6. Запуск Standalone-кластера Spark.

```
start-master.sh
start-worker.sh spark://<master-hostname>:7077
```

7. Выполнение простой задачи: Создайте файл wordcount.py:

```
from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName("Word Count")
sc = SparkContext(conf=conf)

lines = sc.textFile("input.txt")
counts = lines.flatMap(lambda line: line.split(" ")) \
          .map(lambda word: (word, 1)) \
          .reduceByKey(lambda a, b: a + b)

counts.saveAsTextFile("output")
```

8. Запустите задачу.

```
spark-submit wordcount.py
```

9. Проверка результата.
10. Проверьте, что в каталоге output находятся результаты выполнения задачи.

Или

```
/opt/spark/bin/spark-shell

val data = spark.read.option("header", "true").csv("file:///home/your_username/data/AAPL.csv")
data.show(5)
data.describe().show()
```

11. Запуск PySpark и выполнение операций:

Python

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import month, avg, max, min
import matplotlib.pyplot as plt

# Создание SparkSession
spark = SparkSession.builder.appName("TechStocksAnalysis").getOrCreate()

# Загрузка данных
df = spark.read.csv("stocks.csv", header=True, inferSchema=True)

# a) Фильтрация данных для Apple
apple_df = df.filter(df.symbol == "AAPL")

# b) Расчет средней цены закрытия по месяцам
monthly_avg =
apple_df.groupBy(month("date").alias("month")).agg(avg("close").alias("avg_close"))
monthly_avg.show()
```

```
# c) Нахождение максимальной и минимальной цены
max_price = apple_df.agg(max("high")).collect()[0][0]
min_price = apple_df.agg(min("low")).collect()[0][0]
print(f"Максимальная цена: {max_price}")
print(f"Минимальная цена: {min_price}")
```

**Документация и отчет**

Подготовьте отчет, включающий:

1. Описание установки и настройки окружения.
2. Код, использованный для выполнения операций.
3. Результаты выполнения операций (скриншоты, графики).
4. Анализ и интерпретация полученных данных.

**Варианты заданий**

Вариант выбирается согласно номеру студента в списке группы:

1. Установка Apache Spark на одном узле и выполнение простой задачи на подсчет строк в файле.
Данные: Исторические данные по акциям Сбербанка (SBER) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2020 год, расчет средней цены закрытия, группировка по месяцам.
2. Установка Arenadata Hadoop и выполнение задачи на копирование файлов в HDFS.
Данные: Исторические данные по акциям Газпрома (GAZP) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2019 год, расчет максимальной цены открытия, группировка по кварталам.
3. Установка Apache Spark и выполнение задачи на сортировку данных.
Данные: Исторические данные по акциям Лукойла (LKOH) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за последние 5 лет, расчет минимальной цены закрытия, группировка по годам.
4. Настройка кластерного режима для Apache Spark на 2 узлах.
Данные: Исторические данные по акциям Яндекса (YNDX) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2021 год, расчет средней цены закрытия, тренд анализа.
5. Настройка кластерного режима для Arenadata Hadoop на 2 узлах и проверка работоспособности.
Данные: Исторические данные по акциям Роснефти (ROSN) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за последние 3 года, расчет медианной цены закрытия, группировка по месяцам.
6. Установка Apache Spark на кластере из 3 узлов и выполнение задачи на агрегацию данных.
Данные: Исторические данные по акциям Норильского никеля (GMKN) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2020 год, расчет стандартного отклонения цены закрытия, группировка по кварталам.

7. Установка и настройка Apache Spark для работы с внешним источником данных (например, S3).

Данные: Исторические данные по акциям ВТБ (VTBR) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 10 лет, расчет коэффициента вариации цены закрытия, тренд

8. Установка Arenadata Hadoop и выполнение задачи на объединение файлов в HDFS.

Данные: Исторические данные по акциям Магнита (MGNT) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2018 год, расчет средней цены открытия, группировка по годам.

9. Установка и настройка Apache Spark для работы с Cassandra.

Данные: Исторические данные по акциям Полюса (PLZL) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2019 год, расчет средней цены закрытия, корреляция с объемом торгов.

10. Настройка Apache Spark для работы с SQL-запросами.

Данные: Исторические данные по акциям МТС (MTSS) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 2 года, расчет максимальной цены закрытия, тренд анализа.

11. Установка Arenadata Hadoop и выполнение задачи на создание и удаление каталогов в HDFS.

Данные: Исторические данные по акциям Татнефти (TATN) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 5 лет, расчет минимальной цены закрытия, группировка по месяцам.

12. Установка Apache Spark и выполнение задачи на чтение и запись данных из/в Parquet.

Данные: Исторические данные по акциям Сургутнефтегаз (SNGS) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2020 год, расчет стандартного отклонения цены открытия, тренд анализа.

13. Настройка кластерного режима для Apache Spark на 4 узлах с разными ролями узлов.

Данные: Исторические данные по акциям Мечела (MTLR) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 3 года, расчет медианной цены закрытия, группировка по кварталам.

14. Установка Arenadata Hadoop и выполнение задачи на распределение файлов между узлами.

Данные: Исторические данные по акциям Интер РАО (IRAO) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2018 год, расчет средней цены открытия, корреляция с объемом торгов.

15. Установка Apache Spark и выполнение задачи на анализ текстовых данных.

Данные: Исторические данные по акциям Аэрофлота (AFLT) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 2 года, расчет максимальной цены закрытия, тренд анализа.

16. Настройка кластерного режима для Apache Spark на 3 узлах с использованием Docker.

Данные: Исторические данные по акциям Системы (AFKS) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2019 год, расчет средней цены закрытия, группировка по месяцам.

17. Установка Arenadata Hadoop и выполнение задачи на создание и просмотр логов системы.
Данные: Исторические данные по акциям ФосАгро (PHOR) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за последние 5 лет, расчет минимальной цены закрытия, тренд анализа.

18. Установка Apache Spark и выполнение задачи на фильтрацию данных.
Данные: Исторические данные по акциям Алросы (ALRS) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2020 год, расчет стандартного отклонения цены закрытия, группировка

19. Настройка кластерного режима для Arenadata Hadoop на 4 узлах и выполнение задачи на распределенную обработку данных.
Данные: Исторические данные по акциям Русала (RUAL) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за последние 3 года, расчет медианной цены закрытия, корреляция с объемом торгов

20. Установка Apache Spark и выполнение задачи на работу с JSON-файлами.
Данные: Исторические данные по акциям Мосбиржи (MOEX) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2018 год, расчет средней цены открытия, группировка по годам.

21. Настройка Apache Spark для работы с Hive.
Данные: Исторические данные по акциям РУСГИДРО (HYDR) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за последние 10 лет, расчет коэффициента вариации цены закрытия, тренд анализа.

22. Установка Arenadata Hadoop и выполнение задачи на создание резервной копии данных.
Данные: Исторические данные по акциям Россетей (RSTI) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2021 год, расчет максимальной цены закрытия, корреляция с объемом торгов.

23. Установка Apache Spark и выполнение задачи на вычисление статистических параметров данных.
Данные: Исторические данные по акциям X5 Retail Group (FIVE) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за последние 2 года, расчет средней цены закрытия, группировка по месяцам.

24. Настройка кластерного режима для Apache Spark на 2 узлах с использованием Ansible.
Данные: Исторические данные по акциям ТМК (TRMK) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за 2019 год, расчет минимальной цены закрытия, тренд анализа.

25. Установка Arenadata Hadoop и выполнение задачи на слияние данных из нескольких источников в HDFS.
Данные: Исторические данные по акциям М.Видео (MVID) с сайта Московской биржи (moex.com)
Операции: Фильтрация данных за последние 5 лет, расчет стандартного отклонения цены закрытия, группировка по кварталам.