МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Лабораторная работа №\_\_3\_\_**

по дисциплине«Технология обработки больших данных»

Тема: «Работа с большими данными с использованием Apache Spark и Yandex.Cloud»

ИСПОЛНИТЕЛЬ: \_\_Журавлев Н. В.\_\_

ФИО

группа ИУ5-44М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"23"\_\_апреля\_\_\_2025 г.

ПРЕПОДАВАТЕЛЬ: \_\_\_\_Сухобоков А.В.\_\_\_\_\_

ФИО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"23"\_апреля\_\_\_2025 г.

Москва - 2025

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Описание задачи и алгоритма

Этап 1. Изучение теоретических основ Apache Spark и Yandex.Cloud

1. Изучить:

- Архитектуру и основные концепции Apache Spark.

- Понятие RDD и DataFrame, отличия и преимущества.

- Ключевые компоненты Spark: Spark Core, Spark SQL.

- Основные функции облачной платформы Yandex.Cloud, в частности Managed Service for Apache Spark.

Этап 2. Создание Apache Spark-кластера в Yandex.Cloud

1. Зарегистрируйтесь на платформе [Yandex.Cloud](https://cloud.yandex.ru/) и получите тестовый аккаунт с бесплатным периодом.

2. Создайте Managed Apache Spark-кластер с базовыми параметрами (кластер небольшой, 2-3 worker-узла).

Этап 3. Загрузка и подготовка данных

1. Создайте и загрузите в Yandex Object Storage (AWS S3-совместимый сервис в составе Yandex.Cloud) большой файл данных (~1–2 Гб).

2. Создайте или загрузите подготовленный CSV-файл. Рекомендуемая структура:

| user\_id | timestamp | message\_length | message |

|---------|-----------|----------------|---------|

| 1234 | 1705227305 | 24 | текст сообщения |

Этап 4. Реализация задачи анализа больших данных на Apache Spark

Студенты должны реализовать (на выбор):

\*\*Вариант A:\*\* Выявление наиболее активных пользователей

- Загрузить данные из Object Storage в Spark DataFrame.

- Посчитать количество сообщений от каждого пользователя.

- Найти ТОП-10 самых активных пользователей за указанный период времени.

\*\*Вариант B:\*\* Проведение анализа средней длины сообщений по времени

- Вычислить среднюю длину сообщения по каждому дню недели.

- Определить самый «болтливый» день недели.

\*\*Вариант C:\*\* Анализ популярных слов в сообщениях

- Выполнить преобразование текста сообщений, удалить стоп-слова.

- Выявить 20 самых популярных слов в сообщениях пользователей.

Этап 5. Выполнение задач и анализ производительности

1. Выполните запуск написанного приложения в кластере.

2. Измерьте производительность (время выполнения задач, использование памяти и CPU кластера).

3. Проанализируйте различные настройки и их влияние (например, количество узлов, объем памяти).

# Введение

В начале была выполнена установка APACHE SPARK. Для этого необходимо скачать его с официального сайта.

После запустить командой:

./start-all.sh

Далее выполнить написанный скрипт командой:

spark-submit /home/alisk/data/tobd\_lab3.py

# Реализация алгоритма

Необходимо реализовать подсчёт частоты появления каждого отдельного слова. Затем найти 20 самых популярных слов в наборе сообщений (игнорируя стоп-слова). Реализация представлена на листинг 1.

Листинг 1. Алгоритм подсчёта частоты появления каждого слова

|  |
| --- |
| from pyspark.sql import SparkSession  from pyspark.sql.functions import col, explode, split, regexp\_replace, length  spark = SparkSession.builder \  .appName("Message Analysis").getOrCreate()  df = spark.read.csv("file:///home/name/data/data.csv")  STOP\_WORDS = ["a", "no", "of", "A", "in", "at", "on", "or", "it", "me",  "he", "be", "to", "her", "us", "if", "for", "so"]  words\_df = df.withColumn("word",  explode(split(  regexp\_replace(col("\_c3"), "[^a-zA-Z\\s]",  "\\s+"  ))  ).filter(  (col("word") != "")  )  filtered\_words = words\_df.filter(~col("word").isin(STOP\_WORDS))  word\_counts = filtered\_words.groupBy("word").count()  top\_20\_words = word\_counts.orderBy(col("count").desc()).limit(20)  print("Топ 20 самых популярных слов:")  top\_20\_words.show(truncate=False)  spark.stop() |

# Полученные результаты

В результате выполнения алгоритма на 3 видах сгенерированных файлов получились следующие результаты.

Таблица 1. Результат выполнения алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| Размер | 109 МБ | 545 МБ | 1038 МБ |
| Spark | | | |
| Время выполнения | 12 с | 23 с | 35 с |
| Пиковое использование памяти | 366.3 МБ | 366.3 МБ | 366.3 МБ |
| MapReduce | | | |
| Время выполнения | 3.16 с | 20.30 с | 60.00 с |
| Пиковое использование памяти | 196.82 МБ | 797.08 МБ | 1534.33 МБ |

# Вывод

Были освоены практические навыки работы с Apache Spark в облаке Yandex.Cloud, закреплены понятия обработки и анализа больших данных, а также получены навыки настройки и оптимизации решений в облачной среде.