Экосистема Hadoop, Spark, Hive

Экосистема Hadoop, Spark, Hive

Длительность курса: 106 академических часов

1 Scala

1	**Основы Scala**	читать код Scala; писать простые приложения на языке Scala.
2	**Сборка проектов на Scala**	собирать готовые для использования приложения на Scala; использовать библиотеки в вашем проекте. Домашние задания 1 Первое приложение на Scala для работы с файлами в формате JSON Цель: Данное задание позволит студенту создать свой первый Scala-проект и собрать его с помощью SBT, а также познакомиться с базовыми инструментами обработки данных нативными средствами языка Scala Загрузите файл с географическими данными различных стран (https://raw.githubusercontent.com/mledoze/countries/master/countries.json) Среди стран выберите 10 стран Африки с наибольшей площадью Запишите данные о выбранных странах в виде JSON-массива объектов следующей структуры: [{ “name”: <Официальное название страны на английском языке, строка>, “capital”: <Название столицы, строка>(если столиц перечисленно несколько, выберите первую), “area”: <Площадь страны в квадратных километрах, число>,

}]
Обеспечьте проект инструкциями для сборки JAR-файла, принимающего на вход имя выходного файла и осуществляющего запись в него

2 Hadoop

1	**Hadoop**	объяснить назначение Hadoop и его роль в системах обработки данных; назвать основные дистрибутивы Hadoop.
2	**HDFS**	использовать Hadoop для хранения данных; описывать архитектуру и роль Zookeeper в распределенных системах. Домашние задания 1 Работа с файлами на HDFS: запись, считывание и управление файлами Цель: На практике поработать с файлами на HDFS, научитесь записывать и считывать файлы, а также управлять ими Задача описана в репозитории https://github.com/Gorini4/hadoop_course_homework/tree/master/hw1
3	**YARN**	объяснять назначение систем контейнеризации; использовать парадигму MapReduce для решения задач обработки данных.
4	**Форматы данных**	объяснять разницу между основными форматами хранения данных в Hadoop; выбирать подходящие форматы для каждого из типов задач; настраивать сжатие при записи данных.

3 Spark

1	**Архитектура приложения Spark**	писать простейшие приложения на Apache Spark; объяснять назначение Spark; описывать архитектуру приложения Spark.
2	**RDD/Dataframe/Dataset**	использовать RDD API; использовать DataFrame API; использовать Dataset API. Домашние задания 1 Аналитическая витрина на основе сырых данных, используя Spark Цель: Выполнив домашнее задание Вы получите опыт работы с RDD API, DataFrame API,Dataset API. Научитесь строить аналитическую витрину на основе сырых данных, используя Spark и различные API. https://github.com/vadopolski/otus-hadoop-homework.git ДЗ предварительная инструкция 1. Скачать и установить Idea Community - https://www.jetbrains.com/idea/download/#section=windows 2. Установить плагин Скала 3. Скачать и установить Java JDK 11 - https://www.oracle.com/java/technologies/javase-jdk11-downloads.html 4. Скачать и установить git - https://git-scm.com/downloads 5. Скачать и учтановить локально дистрибутив Hadoop (инструкция для Windows - https://www.datasciencecentral.com/profiles/blogs/how-to-install-and-run-hadoop-on-windows-for-beginners) 6. Скачать стартовый проект с Гитхаб с помощью команды git clone https://github.com/vadopolski/otus-hadoop-homework 7. Запустить Idea и открыть скаченный проект File -> Open -> project folder/build.sbt 8. Открыть в проекте файл src/main/scala/homework2/DataApiHomeWorkTaxi.scala запустить его, Ctrl + Shift10 9. Скачать и установить docker-compose 10. Из корневой папки проекта запустить сделать запуск - docker-compose up ДЗ основная инструкция и задания к занятию по Spark Data API: Основная инструкция задание 1: Загрузить данные в первый DataFrame из файла с фактическими данными поездок в Parquet (src/main/resources/data/yellow_taxi_jan_25_2018). Загрузить данные во второй DataFrame из файла со справочными данными поездок в csv (src/main/resources/data/taxi_zones.csv) С помощью DSL

построить таблицу, которая покажет какие районы самые популярные для заказов. Результат вывести на экран и записать в файл Паркет.

Результат: В консоли должны появиться данные с результирующей таблицей, в файловой системе должен появиться файл. Решение оформить в github gist.

Основная инструкция задание 2:
Загрузить данные в RDD из файла с фактическими данными поездок в Parquet (src/main/resources/data/yellow_taxi_jan_25_2018). С помощью lambda построить таблицу, которая покажет В какое время происходит больше всего вызовов. Результат вывести на экран и в txt файл с пробелами.

Результат: В консоли должны появиться данные с результирующей таблицей, в файловой системе должен появиться файл. Решение оформить в github gist.

Основная инструкция задание 3:
Загрузить данные в DataSet из файла с фактическими данными поездок в Parquet (src/main/resources/data/yellow_taxi_jan_25_2018). С помощью DSL и lambda построить таблицу, которая покажет. Как происходит распределение поездок по дистанции? Результат вывести на экран и записать в бд Постгрес (докер в проекте). Для записи в базу данных необходимо продумать и также приложить инит sql файл со структурой.

(Пример: можно построить витрину со следующими колонками: общее количество поездок, среднее расстояние, среднеквадратическое отклонение, минимальное и максимальное расстояние)

Результат: В консоли должны появиться данные с результирующей таблицей, в бд должна появиться таблица. Решение оформить в github gist.

---

3 **Методы оптимизации приложений Spark**

оптимизировать запросы Spark;
использовать Spark UI для поиска проблем с производительностью.

---

4 **Написание коннекторов для Spark**

писать коннекторы для получения/отправки данных в любые сервисы с помощью Spark.

Домашние задания

1 Разработка собственного коннектора на Spark

Цель: В данном ДЗ вы поработаете с DataSource API V2 с целью научиться писать свои собственные коннекторы для Spark.
Задача - доработать data source для Postgres для партиционированного чтения.

1. Склонируйте репозиторий https://github.com/Gorini4/spark_datasource_example
2. Доработайте код в файле src/main/scala/org/example/datasource/postgres/PostgresDatasource.scala так, чтобы тест в файле src/test/scala/org/example/PostgresqlSpec.scala при выполнении читал таблицу users не в одну партицию, а в несколько (размер одной партиции должен задаваться через метод .option("partitionSize", "10")).

Рекомендация: Архитектуру решения можно обсудить в общей группе в Slack.

---

5 **Тестирование приложений Spark**

писать тесты для приложений Spark.

Домашние задания

1 Тесты для приложения Spark

Цель: Выполнив домашнее задание научитесь писать авто тесты для Spark jobs

ДЗ предварительная инструкция
Скачать и установить Idea Community - https://www.jetbrains.com/idea/download/#section=windows
Установить плагин Скала
Скачать и установить Java JDK 11 - https://www.oracle.com/java/technologies/javase-jdk11-downloads.html
Скачать и установить git - https://git-scm.com/downloads
Скачать и учтановить локально дистрибутив Hadoop (инструкция для Windows - https://www.datasciencecentral.com/profiles/blogs/how-to-install-and-run-hadoop-on-windows-for-beginners)
Скачать стартовый проект с Гитхаб с помощью команды
git clone https://github.com/vadopolski/otus-hadoop-homework
Запустить Idea и открыть скаченный проект File -> Open -> project folder/build.sbt
Открыть в проекте файл src/main/scala/homework2/DataApiHomeWorkTaxi.scala запустить его, Ctrl + Shift10
Скачать и установить docker-compose
Из корневой папки проекта запустить сделать запуск - docker-compose up

ДЗ основная инструкция и задания к занятию по Spark Testing:
!!!!! ВНИМАНИЕ, для корректной работы необходимо сделать reimport dependency

Основная инструкция задание 1:
Логически разбить на методы, написанный в домашнем задании к занятию Spark Data API. Пример src/main/scala/lesson2/OtusFragmentedByMethod.scala

Результат: В репозитории должен появиться код с описанием методов. Решение оформить в github.

Основная инструкция задание 2:
Сформировать ожидаемый результат и покрыть простым тестом (с библиотекой AnyFlatSpec) витрину из домашнего задания к занятию Spark Data API, построенную с помощью RDD. Пример src/test/scala/lesson2/SimpleUnitTest.scala

Результат: В репозитории должен появиться код с тестом. Тест должен успешно выполняться при запуске. Решение оформить в github.

Основная инструкция задание 3:
Сформировать ожидаемый результат и покрыть Spark тестом (с библиотекой SharedSparkSession) витрину из домашнего задания к занятию Spark Data API, построенную с помощью DF и DS. Пример src/test/scala/lesson2/TestSharedSparkSession.scala

Результат: В репозитории должен появиться код с тестом. Тест должен успешно выполняться при запуске. Решение оформить в github.

---

6 **Spark ML**

обучать и применять обученные модели на больших объемах данных с помощью Spark.

Домашние задания

1 Настройка применения предобученной модели в Spark Streaming

Цель: В этом ДЗ студент настроит применение предобученной модели в Spark Streaming

4 Streaming

1 **Kafka**

описывать архитектуру Apache Kafka;
использовать консольные утилиты и клиенты для работы с Kafka;
оптимизировать запись и чтение топиков Kafka.

Домашние задания

1 Приложение для чтения данных из Kafka

Цель: В этом ДЗ студент научится использовать Apache Kafka - управлять топиками и читать/писать данные с помощью Scala.

Перед началом выполнения требуется развернуть Kafka через docker-compose (https://github.com/Gorini4/kafka_scala_example) и создать топик books с 3 партициями.
Требуется написать приложение, которое будет выполнять следующее:
1. Вычитывать из CSV-файла, который можно скачать по ссылке - https://www.kaggle.com/sootersaalu/amazon-top-50-bestselling-books-2009-2019, данные, сериализовывать их в JSON, и записывать в топик books локльно развернутого сервиса Apache Kafka.
2. Вычитать из топика books данные и распечатать в stdout последние 5 записей (с максимальным значением offset) из каждой партиции. При чтении топика одновременно можно хранить в памяти только 15 записей.

---

2 **Spark Streaming**

писать приложения для потоковой обработки данных на Spark;
использовать DStreams и Structured Streaming.

---

3 **Structured Streaming**

писать приложения на Structured Streaming, используя state и агрегаты.

Домашние задания

1 Настроите применение предобученной модели в Spark Structured Streaming

Цель: Научимся обучать и сохранять модели Spark ML. Научимся использовать предобученные модели Spark ML в Spark Structured Streaming

1) Построить модель классификации Ирисов Фишера и сохранить её
Описание набора данных:
https://ru.wikipedia.org/wiki/%D0%98%D1%80%D0%B8%D1%81%D1%8B_%D0%A4%D0%B8%D1%88%D0%B5%D1%80%D0%B0
Набор данных в формате CSV:
https://www.kaggle.com/arshid/iris-flower-dataset
Набор данных в формате LIBSVM:
https://github.com/apache/spark/blob/master/data/mllib/iris_libs

		vm.txt Должен быть предоставлен код построения модели (ноутбук или программа) 2) Разработать приложение, которое читает из одной темы Kafka (например, "input") CSV-записи с четырми признаками ирисов, и возвращает в другую тему (например, "predictition") CSV-записи с теми же признаками и классои ириса Должен быть предоставлен код программы 3) Предоставить снимки экрана с записями в обеих темах Kafka
4	**Flink - часть 1**	выбирать подходящий фреймворк для задачи обработки потоковых данных; использовать базовые операции из API Flink.
5	**Flink - часть 2**	писать приложения потоковой обработки данных, используя Flink.

5 Apache Hive

1	**Обзор Hive**	описывать архитектуру Hive; объяснять назначение Hive и аналогичных инструментов.
2	**HiveQL**	писать запросы на HiveQL; создавать таблицы в Hive. Домашние задания 1 Код для построения аналитической витрины в Hive Цель: В это Дз студент напишет код для построения аналитической витрины в Hive

6 Обслуживающие системы

1	**Оркестрация процессов обработки данных**	объяснять назначение оркестраторов в ETL-процессах; использовать Oozie для оркестрации.
2	**Мониторинг и логирование для Spark-приложений**	настраивать мониторинг и логирование для Spark-приложений. Домашние задания 1 Настройка мониторинга для своего приложения Цель: В этом ДЗ студент настроит мониторинг для своего приложения
3	**CI/CD для Spark и Hive**	настраивать процессы сборки и деплоя для Spark и Hive.

7 Проектная работа

1	**Выбор темы и организация проектной работы**	выбрать и обсудить тему проектной работы; спланировать работу над проектом; ознакомиться с регламентом работы над проектом. Домашние задания 1 Проектная работа
2	**Консультация по проектам и домашним заданиям**	получить ответы на вопросы по проекту, ДЗ и по курсу.
3	**Защита проектных работ**	защитить проект и получить рекомендации экспертов.