Гид по безопасному Бостону

В этом задании предлагается собрать статистику по криминогенной обстановке в

разных районах Бостона. В качестве исходных данных используется датасет

https://www.kaggle.com/AnalyzeBoston/crimes-in-boston

С помощью PySpark соберите агрегат по районам (поле district) со следующими

метриками:

--● crimes\_total - общее количество преступлений в этом районе

--● crimes\_monthly - медиана числа преступлений в месяц в этом районе

● frequent\_crime\_types - три самых частых crime\_type за всю историю наблюдений

в этом районе, объединенных через запятую с одним пробелом “, ” ,

расположенных в порядке убывания частоты

○ crime\_type - первая часть NAME из таблицы offense\_codes, разбитого по

разделителю “-” (например, если NAME “BURGLARY - COMMERICAL -

ATTEMPT”, то crime\_type “BURGLARY”)

● lat - широта координаты района, расчитанная как среднее по всем широтам

инцидентов

● lng - долгота координаты района, расчитанная как среднее по всем долготам

инцидентов

Результат должен быть сохранен в один файл в формате .parquet.

Для джойна со справочником необходимо использовать broadcast.

Прислать ссылку на репозиторий с кодом.

Вам могут пригодиться следующие материалы:

1. Документация по Spark: https://spark.apache.org/docs/latest/

2. Документация по SQL функциям Spark:

https://spark.apache.org/docs/latest/api/sql/index.html

Подсказки:

● В спарке есть функция percentile\_approx, которая может посчитать медиану.

● Не забывайте, что конкретный месяц идентифицируется не только номером

месяца, но и номером года.

● В справочнике кодов есть дубликаты. Нужно выбрать уникальные коды, взяв

любое из названий.

Вопросы level1:

Что общего у spark и map reduce?

Они являются методами для распределенных вычислений над очень большими, вплоть до нескольких петабайт, наборами данных в компьютерных кластерах.

2. Чем отличается spark и map reduce?

Spark выполняет обработку данных в памяти и, по сути, почти не обращается к диску. А при возникновении ситуации, когда объем обрабатываемых данных превышает объем RAM, Spark просто сбрасывает часть обрабатываемых данных на диск. В Spark существует API для различных языков программирования, в результате чего писать код заметно проще, да и сам программный код более компактен. Также spark поддерживает анализ данных в режиме реального времени([Spark Streaming](https://www.qubole.com/blog/dstreams-vs-dataframes-two-flavors-of-spark-streaming/" \t "_blank)).

3. Что такое RDD, DataFrame, Dataset, Spark SQL?

RDD - это неизменяемая распределённая совокупность элементов данных, которые могут храниться в памяти или на диске в кластере машин. Данные распределены между машинами на кластере.

DataFrame В отличии от RDD, данные организованы в именованные столбцы, как таблица в реляционной базе данных. Разработан для того, чтобы делать обработку больших данных удобнее.

DataSet – это расширение DataFrame, обеспечивающее дополнительную функциональность.

Spark SQL – это модуль Sparka для структурированной обработки распределенных данных, позволяющий выполнять запросы на языке SQL

4. Что такое Yarn?

YARN – это система планирования заданий и управления, набор системных программ, обеспечивающих совместное использование, масштабирование и надежность работы распределенных приложений. YARN является интерфейсом между аппаратными ресурсами кластера и приложениями, использующих его мощности для вычислений и аналитики больших данных.

5. Что такое драйвер и экзекутор?

Каждое Spark-приложение состоит из управляющего процесса – **драйвера (Driver)** – и набора распределённых рабочих процессов – **исполнителей (Executors).**

Driver запускает метод main() нашего приложения. Здесь создаётся SparkContext.

Исполнитель (Executor) – распределённый процесс, который отвечает за выполнение задач. У каждого приложения Spark собственный набор исполнителей.

6. Что такое Shuffle?

Shuffle - это операция перераспределения данных всего набора по ключу на новые блоки данных, так, чтобы строки с одним и тем же ключом находились только в одном блоке.

7. Что такое UDF на разных Api, особенности?

Пользовательские функции (User Defined Functions, UDF) – это функции, которые не являются встроенными (уже имеющимися) и создаются самим пользователем во время работы с данными при необходимости получить желаемый результат.

Например, мы использовали собственную функцию st, для того чтобы обрезать строку по символу, при помощи питоновской функции partition

def st(x):

    return x.partition(' -')[0]

convertUDF = F.udf(lambda z: st(z))

8. Основные параметры запуска spark приложений?

Напомним, Spark-приложения можно запускать как в локальном режиме (на клиенте, по умолчанию), так и в кластере. Варианты и параметры запуска указываются в специальном скрипте для распределения заданий независимо от типа диспетчера кластера, который называется spark-submit. В общем случае синтаксис этого скрипта выглядит следующим образом:

spark-submit –class <CLASS\_NAME> –num-executors ? –executor-cores ? –executor-memory ?

Назначение указываемых параметров:

–class – главный класс приложения в ООП-языках JAVA или Scala;

–num-executors – число узлов-исполнителей распределенной программы;

–executor-cores – число процессоров для исполнителя;

— executor-memory – объем памяти в байтах для исполнителей;

–driver-memory – объем памяти, выделяемой для driver-программы.

9. Что такое lazy evaluation и несколько примеров?

Это **отложенные (ленивые) вычисления,** которые откладываются до тех пор, пока не понадобится их результат.

Примеры: group\_by, map, union

\*\*\*\*\*

Вопросы level2:

1. У таблицы половина значений одного из полей - null. Какие проблемы могут

возникнуть при джоине с этим полем? Как с ними бороться?

Если нужно, чтобы при джоине скеливались строчки по условию NULL=NULL.

То перед этим логично заполнить все значения Null каким-нибудь общим значением, которое не присутствует в этом поле, например 999999999

2. Спарк приложение запустили с 1000 ядрами, но при джоине используется

только 200, что делать?

Kubernetes в облаке умеет задействовать огромное количество ресурсов на то время, когда они реально используются. Допустим, ваше приложение обычно использует 10 ядер процессора, но иногда для сложной обработки ему нужно 500 или 1000 ядер. В этом случае вам нужно подключить функцию автомасштабирования кластера. Тогда, если ваше приложение запросит 500 ядер, облако их выделит. А когда приложение перестанет требовать такую нагрузку, ресурсы автоматически вернутся в облако.

3. Как использовать dataframe в spark.sql?

Можно создать датафрейм при помощи функции: createDataFrame function и передать в нее данные:

data = [{"Category": 'A', "ID": 1, "Value": 121.44, "Truth": True}, {"Category": 'B', "ID": 2, "Value": 300.01, "Truth": False}, {"Category": 'C', "ID": 3, "Value": 10.99, "Truth": None}, {"Category": 'E', "ID": 4, "Value": 33.87, "Truth": True} ]

df = spark.createDataFrame(data)

4. Написать udaf для определения топ 5 значений категориального поля.

 агрегатные функции (User-Defined Aggregate Function, UDAF) – программируемые пользователем процедуры, которые действуют одновременно с несколькими строками и в результате возвращают одно агрегированное значение.

5. В плане запроса видим, что некая udf вычисляется каждый раз при обращении,

что можно с этим сделать? –

6. С таблицей в цикле осуществляется джоин с сохранением результатов в каждой

итерации, на каждой итерации один и тот же шафл этой таблицы. Как этого

можно избежать? -

7. Что такое листинг? Как его запустить в несколько потоков? –

8. Как cache датафрейма влияет на динамическую аллокацию ресурсов?

Spark предоставляет механизм динамической регулирования ресурсов, которые требуются приложениям, на основе загрузки кластера. Это означает, что приложение может высвободить занятые им ресурсы в пользу кластера при условии, что приложение не использует эти ресурсы в данный момент.

9. На ноде нет ресурсов чтобы создать контейнер, что спарк будет делать?

Докер?

10. Ошибка org.apache.spark.SparkException: Kryo serialization failed: Buffer overflow,

с чем может быть связана?

Проблема связана с переполнением буфера, возможно следует посмотреть настройки в spark.kryoserializer.buffer.max

SQL задача осталась с собеса, дорешать.

Student

Id, Name

----

1, Ivanov

2, Petrov

...

Marks

Student\_id, Mark, Subject, Date

---

1, 5, Math, ....

1, 2, Literature, ...

...

Выбрать id,name учеников у которых никогда не было двоек(без подзапроса, через join):

SELECT student.id, Name

FROM Student

LEFT JOIN Marks

ON Marks. Student \_id= Student.id and marks.mark = 2

WHERE marks.mark IS NULL