BIGDATA TEAM

HW #11: Cassandra + Spark

Deadline: 02.09.2019 08:00

---

1. Описание задания.

В этом задании вы будете разрабатывать схему данных в Cassandra для видео-сервиса, основанного на данных Movielens. От вас потребуется:

1. спроектировать и реализовать в Cassandra схему данных для получения ответов на поставленные вопросы
2. написать запросы для получения ответов
3. написать на Spark SQL код загрузки данных из HDFS в Cassandra

2. Критерии оценивания.

Балл за задачу складывается из:

- 60% - правильное решение задачи
- 20% - эффективность решения
- 20% - поддерживаемость и читаемость кода (Clean Code, см. например Google Python Style Guide)

Штрафы:

- 10% - за несоответствие правилам оформления задания
- 30% - за просрочку дедлайн

3. Описание данных.

3.1 Данные о фильмах

`movies.csv`:

- Путь на кластере: полный датасет - **`/data/movielens/movies.csv`**
- Формат: текст
- В каждой строке находятся следующие поля, разделенные запятой:
    1. `movieId` - id фильма
    2. `title` - заголовок фильма (содержит год выпуска)
    3. `genres` - список жанров в виде строке слов, разделенных символом “|”

Пример:

```
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
```

3.2 Данные о рейтингах

`ratings.csv`:

- Путь на кластере: полный датасет - **`/data/movielens/ratings.csv`**
- Формат: текст
- В каждой строке находятся следующие поля, разделенные запятой:
    1. `userId` - id пользователя
    2. `movieId` - id фильма
    3. `rating`- оценка
    4. `timestamp` - время оценивания

Пример:

```
1,1441,4.0,945544871
```

```
1,1609,3.0,945544824
1,1961,3.0,945544871
```

3. Дополнительная информация.

Запуск cqlsh на virtual-client:

```
$ cqlsh virtual-node01
```

Запуск Spark со Spark Cassandra Connector:

```
sh /home/mf2019q2/mf_surname/pyspark-jupyter.sh port1 port2
```

Как создать DataFrame из таблицы Cassandra можно прочитать здесь

4. Задача №1

Создайте keyspace со стратегией репликации SimpleStrategy и фактором репликации 2. Назовите keyspace в соответствии с вашим логином.

5. Задача №2

moveid	int
title	text
year	int
genres	set<text>

1. Создайте таблицу movies со схемой, приведенной выше. Таблица должна давать возможность выбрать всю информацию по конкретному названию фильма и году (если есть несколько версий одного фильма). Заметьте, что жанры в исходном файле представляют из себя строку слов, разделенных символом вертикальной черты (|). В таблице в Cassandra жанры должны быть представлены типом set.
2. Напишите Spark SQL код, который будет считывать файл movies.csv, приводить DataFrame к правильному формату и запишет данные в таблицу movies в Cassandra. Обратите внимание, что в датасете могут быть ошибки, поэтому ошибки записи в Cassandra могут быть устранены предварительной очисткой данных[1].

[1] (1) предобработка quotes и trailing whitespaces в полях (до и после обработки сырых данных) (2) удаление строк без указания года и других характеристик (3) правильная обработка placeholder для данных, где не указаны жанры (найдите какой это placeholder)

3. Напишите запрос, который посчитает, сколько записей содержится в таблице.

5. Задача №3

1. Создайте таблицу `movies_by_genre`, которая бы отвечала на вопрос: получить все фильмы в этом жанре, отсортированные по убыванию года.
2. Напишите Spark SQL код, который будет заполнять эту таблицу.
3. Напишите запрос, который подсчитает, сколько фильмов жанра Horror было снято в период с 1980 по 1990 год.

6. Задача №4

Ту же информацию, что и в задаче №3, можно получить из исходной таблицы movies, если создать вторичный индекс на столбце genres. Создайте вторичный индекс на этом столбце и напишите запрос, который подсчитает тот же результат, что и в задаче №3. Заметьте, что такой запрос невозможно выполнить без опции ALLOW FILTERING. Внимательно прочитайте про эту опцию и запомните, что ее ни в коем случае нельзя использовать в продакшен запросах! Нужно создать другую схему данных.

7. Задача №5

1. Создайте таблицу `movies_by_genre_rating`, которая позволит отвечать на такой вопрос: вывести все фильмы данного жанра в заданном диапазоне лет, отсортированные по убыванию среднего рейтинга фильма.
2. Напишите Spark SQL код, который будет брать данные из файлов movies.csv и ratings.csv и заполнит таблицу.
3. Подсчитайте минимальный, средний и максимальный рейтинг у фильмов жанра Sci-Fi вышедших в 21 веке.

8. Сроки сдачи и правила оформления задания.

Deadline: 02.09.2019 08:00

Оформление задания:

- Код задания (Short name): **HW11:NoSQL**
- Решения задач должны содержаться в одной папке.
- В результате выполнения домашней работы должны получиться три файла:

1. `movielens_ddl.cql` - файл, в котором содержаться команды создания keyspace, таблиц и индексов
2. `movielens.cql` - файл, в котором содержатся запросы, отвечающие на поставленные вопросы
3. `spark_cassandra.ipynb` или `spark_cassandra.py` - ноутбук или модуль, в котором написан Spark SQL код.

- В обязательном порядке оставьте вывод исполнения ipynb/py в файле `spark_cassandra.out`, чтобы можно было визуально проверить результат без перезапуска решения.
- Выполненное ДЗ запакуйте в архив MF2019Q2_<фамилия>_HW#.zip , например -- MF2019Q2_Ivanov_**HW11**.zip. Например, ваше решение лежит в папке my_solution_folder, тогда чтобы на Linux и Mac OS создать архив под названием hw.zip и пожать его с помощью zip выполните команду[2]:
  - zip -r hw.zip my_solution_folder/

  На Windows 7/8/10: необходимо нажать правую кнопку мыши на директорию my_solution_folder/, выбрать в открывшемся меню "Отправить >", затем "Сжатая ZIP-папка". Теперь можно переименовать архив.
- Присылайте выполненное задание на почту bigdata_mf2019q2@bigdatateam.org с темой письма “Short name. ФИО”. Например: “**HW11:NoSQL**. Иванов Иван Иванович”.
- Перед отправкой письма, оставьте, пожалуйста, отзыв о домашнем задании по ссылке: http://rebrand.ly/mf2019q2_feedback_11_cassandraspark. Это позволит нам скорректировать учебную нагрузку по следующим заданиям (в зависимости от того, сколько часов уходит на решение ДЗ), а также ответить на интересующие вопросы.

Любые вопросы / комментарии / предложения можно писать:

- в телеграм-канале: http://rebrand.ly/mf2019q2_telegram_join
- На почту: bigdata_mf2019q2@bigdatateam.org

9. Дорешка.

Решения после получения фидбека на решение ДЗ можно улучшить. Разрешается одна досылка в течение 1й недели после окончания дедлайна за ДЗ. Соответственно, фидбек за дорешенные ДЗ вы получите в течение 24 часов после окончания deadline следующего ДЗ.

Дорешивать неработающие задания - нельзя. Это позволит исправить дисбаланс между

[2] Флаг -r значит, что будет совершен рекурсивный обход по структуре директории

присланными **работающими** заданиями **после** deadline

VS

присланными **НЕработающими** заданиями **до** deadline

Всем удачи!