BIGDATA TEAM

HW #06: Spark SQL

автор задания: BigData Team, коллективная работа.

1. Описание задания: (Task ID: spark.sssp) Single Source Shortest Path algorithm

В данном ДЗ нужно решить **1 задачу**. Решение надо выполнить с помощью Spark SQL (Dataframe).

Вам необходимо реализовать алгоритм поиска кратчайшего пути от одного пользователя Twitter к другому, используя поиск в ширину (BFS). Для успешной сдачи задания необходимо найти длину кратчайшего пути от follower'а 12 к пользователю 34. Каждый follower в свою очередь может выступать в роли пользователя.

1.1. Описание данных

Социальный граф Twitter

`twitter`:

- Путь на кластере:
  - полный датасет: **`/data/twitter/twitter.txt`**
  - Семпл (для тестирования): **`/data/twitter/twitter_sample_small.txt`**
  - Семпл-2 (для тестирования): **`/data/twitter/twitter_sample.txt`**
- Формат: текст
- В каждой строке находятся следующие поля, разделенные знаком табуляции:
  - `INT` - ID пользователя
  - `INT` - ID follower'а
- Граф считаем направленным: `user` ← `follower`.

Пример:

```
12	999
1	12
2	1
3	1
34	3
...
```

1.2. Требования к реализации:

- PySpark-скрипт для запуска решения следует назвать `task_<Surname>_<Name>_sssp.py`;
- решение будет запускаться с помощью команды: `PYSPARK_DRIVER_PYTHON=python3.6 PYSPARK_PYTHON=python3.6 spark-submit "task_<Surname>_<Name>_sssp.py";`
- в ходе реализации **запрещается** использовать collect более 1-го раза (можно для получения финального результата - длина пути). Запрещено использовать collect в циклах и рекурсиях;
- ваше решение должно вывести в STDOUT ровно одно число - длину кратчайшего пути между этими пользователями;
- если для выполнения этого задания вам потребуется реализовать UDF, то ее необходимо реализовать именно как **`pandas_udf`** для ускорения работы алгоритма. Также посмотрите, нет ли необходимой вам функции в модуле **`pyspark.sql.functions`** (возможно, она там действительно есть).
- Вывод STDOUT сохранить в файле `task_<Surname>_<Name>_sssp.out`[1]

Для тестирования решения предлагается пользоваться неполными датасетами. Длина кратчайшего пути между заданными вершинами в тестовых датасетах одинаковая, но отличается от длины пути в полных данных!

Пример вывода:

```
10
```

[1] Для подготовки архива с решением и выводом результатов запуска можно воспользоваться командой “tee”

2. Критерии оценивания

Балл за задачу складывается из:

- **60%** - правильное решение задачи
- **20%** - поддерживаемость и читаемость кода
  - в общем случае см. Clean Code и Google Python Style Guide
  - оценка качества будет проводиться автоматическим вызовом pylint:
    - `pylint *.py -d invalid-name,missing-docstring --ignored-modules=pyspark.sql.functions`
    - качество кода должно оцениваться выше 8.0 / 10.0
    - проверяем код **Python версии 3** с помощью `pylint==2.5.3`
- **20%** - эффективность решения (такие как потребляемые CPU-ресурсы, скорость выполнения (в предположении свободного кластера)).

Discounts (скидки и другие акции):

- **100%** за плагиат в решениях (всем участникам процесса)
- **100%** за посылку решения после hard deadline
- **30%** за посылку решения в после soft deadline и до hard deadline
- **5%** за каждую дополнительную посылку в тестирующую систему (всего можно делать до 3-х посылок без штрафа):

Пример работы системы штрафов:

День	Посылка	Штраф
День 1	Посылка 1	Без штрафа
День 1	Посылка 2	Без штрафа
День 1	Посылка 3	Без штрафа
День 1	Посылка 4	-5%
День 2	Посылка 5	-5%
День 3	Посылка 6	-5%
Итоговый штраф: -15%		

Для подсчета финальной оценки **всегда** берется **последняя** оценка из Grader.

3. Правила оформления задания

Перед отправкой задания оставьте, пожалуйста, отзыв о домашнем задании по ссылке: https://rebrand.ly/bdmade2022q2_feedback_hw. 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.

Оформление задания:

- Код задания (Short name): **HW06:Spark_SQL**.
- Выполненное ДЗ запакуйте в архив **BD-MADE-2022-Q2**_<Surname>_<Name>_HW#**.zip**, например, для Алексея Драля -- **BD-MADE-2022-Q2**_Dral_Alexey_**HW06**.zip. Если ваше решение лежит в папке my_solution_folder, то для создания архива hw.zip на Linux и Mac OS выполните команду[2]:
  - `zip -r hw.zip my_solution_folder/*`
- На Windows 7/8/10: необходимо выделить все содержимое директории my_solution_folder/ нажать правую кнопку мыши на одном из выделенных объектов, выбрать в открывшемся меню "Отправить >", затем "Сжатая ZIP-папка". Теперь можно переименовать архив.
- Решения заданий должны содержаться в одной папке.
- Перед проверкой убедитесь, что дерево вашего архива выглядит так:
  - **| BD-MADE-2022-Q2_<Surname>_<Name>_HW06.zip**
  - **| ---- task_<Surname>_<Name>_sssp.py**
  - **| ---- task_<Surname>_<Name>_sssp.out**
  - При несовпадении дерева вашего архива с представленным деревом, ваше решение будет невозможно автоматически проверить, а значит, и оценить его.
- Для того, чтобы сдать задание необходимо:
  - Зарегистрироваться и залогиниться в сервисе Everest
  - Перейти на страницу приложения: MADE BigData Grader
  - Выбрать вкладку Submit Job (если отображается иная).
  - Выбрать в качестве “Task” значение: **HW06:Spark_SQL**[3]
  - Загрузить в качестве “Task solution” файл с решением
  - В качестве Access Token указать тот, который был выслан по почте

Любые вопросы / комментарии / предложения можно писать в телеграм-канал курса или на почту bd_made2022q2@bigdatateam.org.

Всем удачи!

[2] Флаг -r значит, что будет совершен рекурсивный обход по структуре директории
[3] Сервисный ID: spark.sssp

4. FAQ (часто задаваемые вопросы)

"You are not allowed to run this application", что делать?
Если Вы видите надпись "You are not allowed to run this application" во вкладке Submit Job в Everest, то на данный момент сдача закрыта (нет доступных для сдачи домашних заданий, по техническим причинам или другое). Попробуйте, пожалуйста, еще раз через некоторое время. Если Вы еще ни разу не сдавали, у коллег сдача работает, но Вы видите такое сообщение, сообщите нам об этом.

Grader показывает 0 или < 0, а отчет (Grading report) не помогает решить проблему
Ситуации: • система оценивания показывает оценку (Grade) < 0, а отчет (Grading report) не помогает решить проблему. Пример: в случае неправильно указанного access token система вернет -401 и информацию о том, что его нужно поправить; • система показывает 0 и в отчете (Grading report) не указано, какие тесты не пройдены. Пример: вы отправили невалидный архив (rar вместо zip), не приложили нужные файлы (или наоборот приложили лишние - временные файлы от Mac OS и т.п.), рекомендуется проверить содержимое архива в консоли: `unzip -l your_solution.zip` Если Вы столкнулись с какой-то из них, присылайте ссылку на выполненное задание (Job) в чат курса. Пример ссылки: https://everest.distcomp.org/jobs/67893456230000abc0123def

Что в отчете Grader означает проверка X ?
Как читать отчет: Для каждого теста • Raw_score - балл за конкретный тест. Может быть как бинарным (1\0), так и находиться в интервале от 0 до 1 • Score - Raw_score*weight (вес теста в общей оценке). Вес указан для каждого теста ниже

Итоговая оценка: смотрите строку Score (сумма Score всех индивидуальных тестов) внизу отчета.

Правильность решения задачи:
test_unzip_is_succesful (weight = 0) - ДЗ заархивировано в .zip архив и грейдер может его разархивировать.

test_expected_format_for_spark_sssp_stdout (weight = 0.0) - корректность вывода (stdout)

test_sssp_is_found_correctly (weight = 0.6) - верный ответ задачи

Поддерживаемость и читаемость кода:
test_py_files_min_lint_score (weight = 0.2) - качество кода в .py файлах оценивается выше 8.0

Эффективность решения:
test_solution_finish_within_max_execution_time (weight = 0.2) - оценка скорости исполнения скрипта. Граница скорости: 4 мин.