BIGDATA TEAM

#W5L105: Spark RDD. Workshop.

1. Цель занятия

1. Научиться запускать Spark с помощью PySpark и пробрасывать нужные порты;
2. Научиться работать со Spark RDD API;
3. Познакомиться на практике с преобразованиями (transformations) и действиями (actions);
4. Научиться чистить сырые данные, проводить агрегацию и запускать join'ы.

Вопросы оптимизации будут рассмотрены в следующих учебных модулях, но красивые картинки из Spark UI (см. DAG'и) можно смело постить в чате курса, чтобы хвастаться и обсуждать.

2. Запуск PySpark

PySpark можно запускать в трех различных режимах:

1. Запуск в интерактивной консоли;
2. Запуск с интерфейсом jupyter;
3. Запуск скрипта с помощью spark-submit.

Первые две опции были рассмотрены в рамках "W5L102. Архитектура Spark-приложения и Spark RDD". Команду для запуска можно найти в README.md проекта на github курса:

- https://github.com/big-data-team/big-data-course

Обратите внимание на правильность проброса портов, с этим бывает много проблем, поэтому не стесняйтесь спрашивать помощи у других слушателей. Вам могут пригодиться следующие порты для работы:

- 50070 (Namenode);
- 8088 (ResourceManager);
- 19888 (Job History Server);
- 18080 (Spark History Server);
- XXXXX (порт для Jupyter-интерфейса для интерактивной сессии PySpark).

Третий режим запуска не является интерактивным, он используется для запуска регулярных задач, а не в формате ad-hoc экспериментов. Его мы будем использовать для подготовки и сдачи домашнего задания и там об этом будет написано подробнее.

3. Задачи на Spark RDD и PairRDD

Полезные ссылки для выполнения задания:

- RDD Programming Guide
- PairRDDFunctions

Задача 1. Разминка

Вам нужно провести аналитику с помощью Spark на основе датасета о марках машин. Ваша задача - посчитать, сколько марок машин есть в датасете по интересующим срезам (например - по первой букве в названии марки).

акт первый, часть первая

Запустите PySpark сессию;

акт первый, часть вторая

Создайте свою первую коллекцию RDD с помощью sc.parallelize на основе первых букв английского алфавита (буквы в разном регистре будем считать разными). Рекомендуем воспользоваться переменной `ascii_letters` из стандартного Python-модуля `string`;

акт первый, часть третья

Прочитайте данные о марках машин в память на драйвере и сделайте из них коллекцию RDD. Данные доступны на github курса:

- github:big-data-team/big-data-course/.../spark/rdd/workshop/car_brands.txt

акт второй (последний)

Получите RDD, содержащий количество производителей автомобилей с разбиением по буквам, с которых начинается название производителя (буквы в разном регистре будем считать разными). Если для какой-то буквы нет ни одного производителя, количество должно быть равно 0. Результат должен иметь вид

```
RDD[(letter, count)]
```

Задача 2. Конкурсы и шарады с костылями и приседаниями

В этом задании вам предстоит поработать с “грязными” данными (все как в реальной жизни). Возьмите битый датасет, который содержит статистику о разных городах, доступный по адресу:

- github:big-data-team/big-data-course/.../spark/rdd/workshop/cities.jsonlines

В каждой строке находится строка в формате json, вам нужно построить на основе этих данных RDD из словарей с условием, что:

- если континент отсутствует - ставим Earth;
- если отсутствует population - ставим 0;
- если это невалидная запись json, то такую запись игнорируем;

После этого можем переходить непосредственно к аналитике на Spark:

1. Найдите все дубликаты и создайте новый RDD без дублей;
2. Посчитайте статистику по континентам: количество городов и сумма населения;
3. Найдите наиболее населенный город;

Бонусное задание:

- В MapReduce мы детально разбирали работу partitioner'а, в Spark он тоже позволяет определить сплиты для обработки (здесь называемые партициями). Посмотрите сколько партиций используется в вашем RDD и попробуйте задать кастомный partitioner на основе значения "continent". Проверьте, что данные распределяются правильно, собрав их с помощью `.glom().collect()`

4. Обратная связь

Обратная связь: http://rebrand.ly/mailbd2021q1_feedback_module

Просьба потратить 1-2 минут Вашего времени, чтобы поделиться впечатлением, описать что было понятно, а что непонятно. Мы учитываем рекомендации и имеем возможность переформатируем учебную программу под Ваши запросы.