BIGDATA TEAM

# #03: MapReduce Optimization. Workshop

## 1. Цель занятия

Научиться тестировать MapReduce Streaming приложения, оптимизировать их с помощью Combiner. Практика на использование Partitioner и Comparator доступна в домашнем задании.

*DISCLAIMER*: Не надо беспокоиться, если Вы что-либо не успели. Всегда остается возможность продолжить погружение дома и иметь возможность задавать вопросы в Telegram-канале.

## 2. Общие рекомендации

Чтобы пробросить порты 8088, 50070, 19888 воспользуйтесь инструкцией из User Guide.

Для удобства копирования, исходные файлы лежат в папках:

**/usr/local/share/big_data_course/public_examples/map_reduce**

**/usr/local/share/big_data_course/public_examples/map_reduce_word_count**

**/usr/local/share/big_data_course/public_examples/map_reduce_distributed_cache**

## 3. MapReduce Python Streaming, -files, тестирование

См. слайды 5 и 8 для написания задачи “Line Count” полностью на Python.

Предлагается действовать в 2 этапа:

1. Написать Python-скрипты и протестировать их локально.
2. Обновить run.sh, чтобы запустить полностью Python Streaming MR-приложение.

Эмуляция работы Hadoop локально достигается следующим образом:

- Скачиваем часть данных себе локально из HDFS (см. **/data/wiki/en_articles_part**) под названием **sample.txt**:

  ```
  hdfs dfs -cat /data/wiki/en_articles_part/* | head -n 50 > sample.txt
  ```

- Тестируем MapReduce-скрипты:

  ```
  cat sample.txt | python3 mapper.py | sort | python3 reducer.py > out.txt
  ```

Таким образом, мы можем оперативно проверить работоспособность скриптов до запуска в распределенном режиме.

## 4. Задача Word Count на Python

Исходный код базового WordCount возьмите по адресу:

```
/usr/local/share/big_data_course/public_examples/map_reduce_word_count
```

Задача 4.1. Написать приложение WordCount, не учитывающее мусор (знаки пунктуации).

Задача 4.2. Написать приложение WordCount, приводящее все слова к нижнему регистру.

## 5. Distributed Cache

1. Найдите какую-нибудь статистику употребления имен / слов в интернете.
2. На основе статистики подготовьте как минимум два файла для фильтрации слов (например: male.txt и female.txt[1]).
3. Сделайте из них tar-архив.
4. Посчитайте WordCount на основе (семпла) Википедии (`/data/wiki/en_articles_part`) с помощью Distributed Cache.
5. Сравните полученные статистики.
6. Поделитесь в телеграм-чате что нашли интересного.

## 6. Обратная связь

Обратная связь: http://rebrand.ly/x5bd2021q1_feedback_03_mro

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

[1] Исходные мужские и женские имена находятся по адресу: /usr/local/share/big_data_course/public_examples/map_reduce_distributed_cache