ДЗ 2

Кондраев Дмитрий

Исходники см. в репозитории https://github.com/mariohuq/2022-polis-ml. `git checkout` выполняется в нем.

1 Приведите пример Map only и Reduce задачи. (1 балл)

Пример Map-only задачи — поменять порядок колонок в дата-сете, добавить новую колонку по заранее известному правилу, зависящему только от текущей строки.

Reduce-only — подсчет общего количества входных данных.

2 Может ли стадия Reduce начаться до завершения стадии Map? Почему? (2 балл)

Да, копирование результатов уже закончивших работу Mappers и сортировка может начинаться до полного завершения Map-задач. Но выполнение самого Reduce может начаться только когда все Mappers закончат работу, так как записи сортируются по ключу и если начать Reduce раньше, не будет гарантии, что все записи с определенным ключом будут обработаны должным образом.

3 Разверните кластер hadoop, соберите WordCount приложение, запустите на датасете ppkm_sentiment и выведите 10 самых редких слов (1 балл)

Старт контейнера (созданного в предыдущем ДЗ)

```
$ systemctl start docker.service
$ docker start hdp -i
```

Сборка и копирование Jar с WordCount Job (версия на Java):

```
$ cd 02-mapreduce/wordcount
$ ./gradlew build
$ docker cp ./**/wordcount-1.0-SNAPSHOT.jar hdp:/home/hduser/
$ cd ..
```

Запуск первой задачи (подсчет слов):

```
hduser@localhost:~$ hdfs dfs -rm -r ppkm.out-java
hduser@localhost:~$ hadoop jar wordcount-1.0-SNAPSHOT.jar ok.ml.WordCount\
  -Dwordcount.input=ppkm/ppkm_dataset.csv -Dwordcount.output=ppkm.out-java
```

Сборка и копирование Jar с wordswap Job (версия на Scala):

```
$ cd 02-mapreduce/wordswap
$ sbt assembly
$ docker cp ./**/wordswap-assembly-0.1.0-SNAPSHOT.jar hdp:/home/hduser/
```

Запуск второй задачи (меняет местами слово и количество повторений):

```
hduser@localhost:~$ hdfs dfs -rm -r ppkm.out-scala
hduser@localhost:~$ hadoop jar wordswap-assembly-0.1.0-SNAPSHOT.jar\
  -Dswap.input=ppkm.out-java -Dswap.output=ppkm.out-scala
```

10 самых редких слов:

```
hduser@localhost:~$ hdfs dfs -cat ppkm.out-scala/part-r-00000 | head -11
1 ah
1 bs
1 |
1 y
1 wi
1 w
1 tu
1 (
1 to
1 sm
1 rs
```

4 Перепишите WordCount на Scala (2 балла)

Сборка решения на Scala:

```
$ cd wordcount-2
$ git checkout b823eedc8dccfa8ed99b120dc245fc8bc6c7a808
$ sbt assembly
$ docker cp ./**/wordcount-2-assembly-0.1.0-SNAPSHOT.jar hdp:/home/hduser/
```

Запуск и проверка:

```
hduser@localhost:~$ hdfs dfs -rm -r ppkm.out-v2
hduser@localhost:~$ hadoop jar wordcount-2-assembly-0.1.0-SNAPSHOT.jar\
  -Dwordcount.input=ppkm/ppkm_dataset.csv -Dwordcount.output=ppkm.out-v2
hduser@localhost:~$ hdfs dfs -tail ppkm.out-v2/part-r-00000 | tail -10
sm  1
to  1
tp  3
tu  1
w 1
wi  1
y 1
ya  6
yg  56
| 1
```

5 Измените маппер в WordCount так, чтобы он удалял знаки препинания и приводил все слова к единому регистру (Java: 1 балл, Scala: 2 балла)

Решение на Scala:

```
$ cd wordcount-2
$ git checkout c4ac8f2432e2e70d5adcd284a9696121c3221669
$ sbt assembly
$ docker cp ./**/wordcount-2-assembly-0.1.0-SNAPSHOT.jar hdp:/home/hduser/
```

Результат:

```
hduser@localhost:~$ hdfs dfs -rm -r ppkm.out-v2
hduser@localhost:~$ hadoop jar wordcount-2-assembly-0.1.0-SNAPSHOT.jar\
  -Dwordcount.input=ppkm/ppkm_dataset.csv -Dwordcount.output=ppkm.out-v2
hduser@localhost:~$ hdfs dfs -tail ppkm.out-v2/part-r-00000 | tail -10
t 84
to  1
tp  4
```

```
tu  1
w 1
wh  1
wi  1
y 1
ya  13
yg  60
```

6 На кластере лежит датасет, в котором ключами является id сотрудника и дата, а значением размер выплаты. Руководитель поставил задачу рассчитать среднюю сумму выплат по каждому сотруднику за последний месяц. В маппере вы отфильтровали старые записи и отдали ключ-значение вида: id-money. А в редьюсере суммировали все входящие числа и поделили результат на их количество. Но вам в голову пришла идея оптимизировать расчет, поставив этот же редьюсер и в качестве комбинатора, тем самым уменьшив шафл данных. Можете ли вы так сделать? Если да, то где можно было допустить ошибку? Если нет, то что должно быть на выходе комбинатора? (2 балла)

Нельзя просто так использовать редьюсер, вычисляющий среднее в качестве комбинатора, так как должно выполняться свойство

$$\text{mean}\left(\text{mean}(a_1, \dots, a_n), \dots, \text{mean}(z_1, \dots, z_n)\right) = \text{mean}(a_1, \dots, a_n, \dots, z_1, \dots, z_n)$$

Но оно, очевидно, не выполняется:

$$\frac{\frac{1+2}{2} + 3}{2} = 2.25 \quad \frac{1+2+3}{3} = 2$$

Для обеспечения корректной работы на выходе комбинатора должно быть не одно число, а два: среднее и количество или сумма и количество:

$$(\texttt{sum}_1, \texttt{count}_1), (\texttt{sum}_2, \texttt{count}_2) \to (\texttt{sum}_1 + \texttt{sum}_2, \texttt{count}_1 + \texttt{count}_2)$$