

Домашнее задание Урок 2. Apache Spark DSL

```
pwd
export SPARK_HOME=/home/hduser/spark
ls spark/bin
echo $PATH
spark-submit
ls -la
vim .bashrc
```

```
hdfs dfs -ls /user/hduser/
hdfs dfs -ls ppkm_out
hdfs dfs -cat ppkm_out/*
hdfs dfs -cat /user/hduser/ppkm_out/*
hdfs dfs -ls /user/hduser/ppkm-rdd-out
hdfs dfs -cat /user/hduser/ppkm-rdd-out/*
```

Редактирование в vim .bashrc:

```
export SPARK_HOME=/home/hduser/spark
export PATH=$PATH:$SPARK_HOME/bin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

Запускаем приложение:

```
spark-submit --class WordCount --master yarn --deploy-mode cluster scalasparkrdd_2.11-0.1.jar /user/hduser/ppkm /user/hduser/scala-ppkm-rdd-scala
```

1. Какие плюсы и недостатки у Merge Sort Join в отличии от Hash Join?

Объединение «сортировка слиянием» выполняется путем сортировки двух наборов данных, которые должны быть объединены, в соответствии с ключами объединения, а затем их объединения. Слияние очень дешево, но сортировка может быть чрезмерно дорогой, особенно если сортировка переносится на диск. Стоимость сортировки может быть снижена, если к одному из наборов данных можно получить доступ в отсортированном порядке через индекс, хотя доступ к высокой доле блоков таблицы через сканирование индекса также может быть очень дорогостоящим по сравнению с полным сканированием таблицы.

Хеш-соединение выполняется путем хеширования одного набора данных в память на основе столбцов соединения и чтения другого и проверки хэш-таблицы на совпадения. Хэш-соединение очень дешево, когда хеш-таблица может храниться полностью в памяти, при этом общая стоимость очень немногим превышает стоимость чтения наборов данных. Стоимость возрастает, если хеш-таблица должна быть перенесена на диск при однократной сортировке, и значительно возрастает при многократной сортировке.

(В версиях до 10g внешние соединения большой таблицы с малой были проблематичными с точки зрения производительности, поскольку оптимизатор не мог решить необходимость сначала обращаться к меньшей таблице для хеш-соединения, а к большей таблице сначала для внешнего соединения. Следовательно, в этой ситуации хеш-соединения были недоступны).

Стоимость хэш-соединения можно снизить, разделив обе таблицы по ключу (-ам) соединения. Это позволяет оптимизатору сделать вывод, что строки из раздела в одной таблице найдут совпадение только в конкретном разделе другой таблицы, а для таблиц, имеющих n разделов, хеш-соединение выполняется как n независимых хеш-объединений. Это имеет следующие эффекты:

1. Размер каждой хэш-таблицы уменьшается, следовательно, уменьшается максимальный объем требуемой памяти и потенциально устраняется необходимость для операции, требующей временного дискового пространства.
2. Для параллельных операций запроса объем обмена сообщениями между процессами значительно сокращается, что снижает использование ЦП и повышает производительность, поскольку каждое хеш-соединение может выполняться одной парой процессов PQ.
3. Для непараллельных операций запроса потребность в памяти уменьшается в n раз, и первые строки проецируются из запроса ранее.

Следует отметить, что хеш-соединения могут использоваться только для равных объединений, но объединения слиянием более гибкие.

В общем, если вы объединяете большие объемы данных в равное соединение, то хеш-соединение будет лучшим выбором.

2. **Соберите WordCount приложение, запустите на датасете [ppkm_sentiment](#)**

WordCount на java

```
Терминал - hduser@localhost: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
(seberangnya,1)
(negatif,Sampai,1)
(contoh,1)
(https://arahkita.com/news/read/22277/depok_siapkan_sanksi_pidana_bagi_pelanggar_psbbs,1)
(Atau,1)
(diperpanjang,13)
(milyaran,1)
(Selorejo,2)
(tmp,1)
(#NewsUpdate,3)
(didukung,,1)
(Bersama,1)
(melakukan,2)
(https://t.co/vlTMzJ1gUY,1)
(#ppkm,2)
(Pemberlakuan,7)
(netral,"Mau,1)
(#Beritaonline,1)
(https://t.co/YM3sB4c68t,1)
(#JogjaElingLanWaspada,21)
(TOLONG,1)
(Kembali,1)
(amin...salam,1)
(#PPKMMikro,34)
(Hahaha,1)
(supaya,3)
(dg,3)
(pemberlakuan,4)
(mulu,,1)
(salahnya,1)
(DAN,1)
(buka,3)
(Munir,1)
(sudah,9)
hduser@localhost:~$
```

```
Терминал - hduser@localhost: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
(Atau,1)
(diperpanjang,13)
(milyaran,1)
(Selorejo,2)
(tmp,1)
(#NewsUpdate,3)
(didukung,,1)
(Bersama,1)
(melakukan,2)
(https://t.co/vlTMzJ1gUY,1)
(#ppkm,2)
(Pemberlakuan,7)
(netral,"Mau,1)
(#Beritaonline,1)
(https://t.co/YM3sB4c68t,1)
(#JogjaElingLanWaspada,21)
(TOLONG,1)
(Kembali,1)
(amin...salam,1)
(#PPKMMikro,34)
(Hahaha,1)
(supaya,3)
(dg,3)
(pemberlakuan,4)
(mulu,,1)
(salahnya,1)
(DAN,1)
(buka,3)
(Munir,1)
(sudah,9)
hduser@localhost:~$
```

WordCount на scala

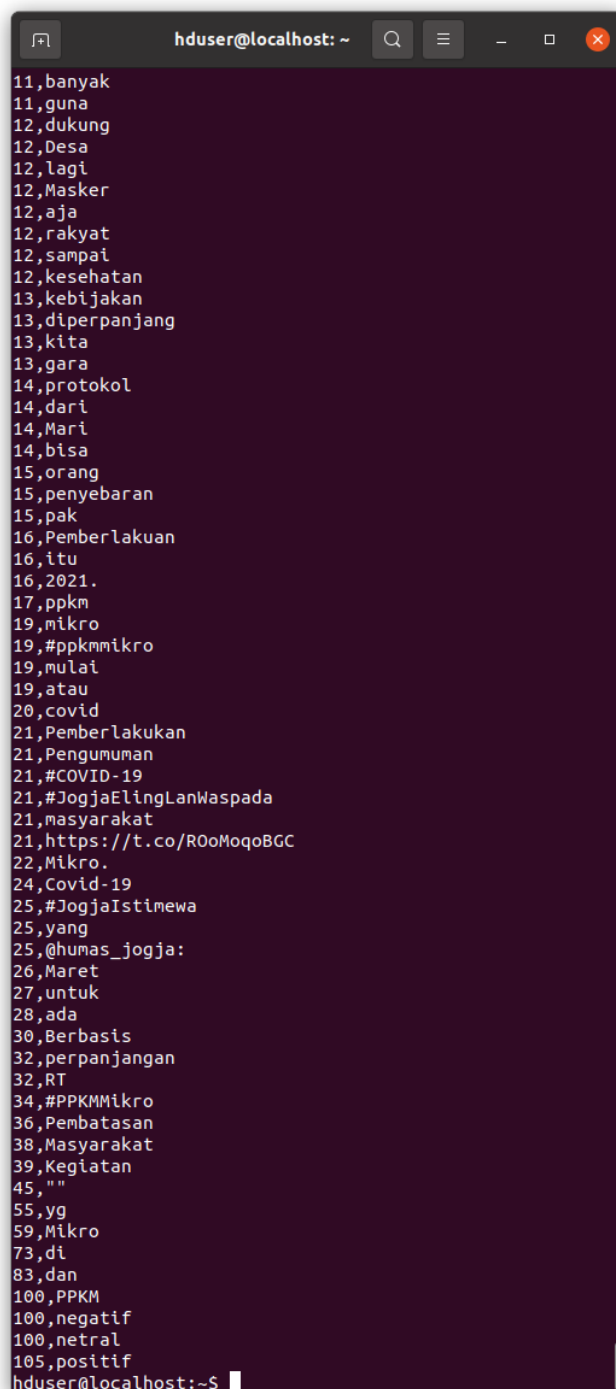
lagi 12
Desa 12
Masker 12
sampai 12
ini 13
kebijakan 13
kita 13
bisa 13
diperpanjang 13
protokol 14
itu 15
pak 15
penyebaran 15
dari 16
ppkm 16
2021. 16
orang 16
mikro 18
mulai 19
atau 19
#ppkmmikro 19
masyarakat 20
covid 20
netral,RT 21
<https://t.co/R0oMoqoBGC> 21
Pemberlakukan 21
#COVID-19 21
Pengumuman 21
#JogjaElingLanWaspada 21
Mikro. 22
Covid-19 23
@humas_jogja: 25
#JogjaIstimewa 25
Maret 26
untuk 27
yang 27
ada 29
Berbasis 30
perpanjangan 32
#PPKMMikro 34
Pembatasan 36
Masyarakat 38
Kegiatan 39
41
yg 55
Mikro 56
di 74
PPKM 84
dan 84
hduser@localhost:~\$

3. Измените WordCount так, чтобы он удалял знаки препинания и приводил все слова к единому регистру

```
String[] words = instring.replaceAll("[^a-zA-Z ]", "").toLowerCase().split("\\s+");
```

4. Измените выход WordCount так, чтобы сортировка была по количеству повторений, а список слов был во втором столбце, а не в первом

```
.groupBy($"word").count()  
.sort($"count")  
.select($"count", $"word")
```



```
11,banyak  
11,guna  
12,dukung  
12,Desa  
12,lagi  
12,Masker  
12,aja  
12,rakyat  
12,sampai  
12,kesehatan  
13,kebijakan  
13,diperpanjang  
13,kita  
13,gara  
14,protokol  
14,dari  
14,Mari  
14,bisa  
15,orang  
15,penyebaran  
15,pak  
16,Pemberlakuan  
16,itu  
16,2021.  
17,ppkm  
19,mikro  
19,#ppkmmikro  
19,mulai  
19,atau  
20,covid  
21,Pemberlakukan  
21,Pengumuman  
21,#COVID-19  
21,#JogjaElingLanWaspada  
21,masyarakat  
21,https://t.co/R0oMoqoBGC  
22,Mikro.  
24,Covid-19  
25,#JogjaIstimewa  
25,yang  
25,@humas_jogja:  
26,Maret  
27,untuk  
28,ada  
30,Berbasis  
32,perpanjangan  
32,RT  
34,#PPKMMikro  
36,Pembatasan  
38,Masyarakat  
39,Kegiatan  
45," "  
55,yg  
59,Mikro  
73,di  
83,dan  
100,PPKM  
100,negatif  
100,netral  
105,positif  
hduser@localhost:~$
```

5. Измените выход *WordCount*, чтобы формат соответствовал *CSV*

```
wc.write().mode(SaveMode.Overwrite).csv(output);
```

```
implicit class PimpedStringRDD(rdd: RDD[String]) { def write(p: String)(implicit ss: SparkSession): Unit = { import ss.implicits._  
rdd.toDF().as[String].write.mode(SaveMode.Overwrite).text(p) } }
```