Pig – Hands-On

Objetivo: O objetivo deste hands-on é instalar e testar o software Pig.

Instalação do Pig

1. Faça download da versão 0.17 do Pig (http://pig.apache.org).

```
$ cd /vagrant
$ wget -c http://ftp.unicamp.br/pub/apache/pig/pig-0.17.0/pig-0.17.0.tar.gz
```

2. Descompacte o arquivo e teste se o pig está funcionando.

```
$ tar -xvf pig-0.17.0.tar.gz
$ export PIG_HOME=/vagrant/pig-0.17.0
$ $PIG_HOME/bin/pig --version
```

3. Faça o download do arquivo stations.csv. Crie uma pasta chamada stations no HDFS e faça upload do arquivo para esta pasta.

```
$ wget -c
https://raw.githubusercontent.com/GerInfraBigDataPUCRS/CursoDataScience2018/master/06_Pig/stations.csv
$ $HADOOP_HOME/bin/hadoop fs -mkdir stations
$ $HADOOP_HOME/bin/hadoop fs -put stations.csv stations
```

4. Execute o serviço MapReduce JobHistory utilizado pelo Pig para coletar estatísticas e mensagens

```
$ $HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver
```

5. Abra uma sessão no grunt (pig shell)

```
$ $PIG_HOME/bin/pig
```

6. Execute os seguintes commandos no terminal:

```
stations = LOAD 'stations' USING PigStorage(',') AS
      (station_id:int, name:chararray, lat:float, long:float,
       dockcount:int, landmark:chararray, installation:chararray);
station_ids_names = FOREACH stations GENERATE station_id, name;
ordered = ORDER station_ids_names BY name;
```

7. Teste os comandos DESCRIBE e ILLUSTRATE

```
DESCRIBE stations;
ILLUSTRATE ordered;
```

8. Solicite para que o conteúdo de ordered seja gerado e impresso no terminal.

```
DUMP ordered;
```

9. Saia da sessão

```
QUIT;
```

10. Crie um arquivo chamada list_stations.pig com o conteúdo abaixo (mesmo programa executado na shell, porém escrevendo a saída no HDFS usando o comando STORE).

```
$ cat > list_stations.pig << EOF
stations = LOAD 'stations' USING PigStorage(',') AS
      (station_id:int, name:chararray, lat:float, long:float,
       dockcount:int, landmark:chararray, installation:chararray);
station_ids_names = FOREACH stations GENERATE station_id, name;
ordered = ORDER station_ids_names BY name;
STORE ordered INTO 'ordered';
EOF
```

11. Execute o programa em modo batch

```
$ PIG_HOME/bin/pig list_stations.pig
```

12. Consulte a saída gerada no HDFS

```
$ $HADOOP_HOME/bin/hadoop fs -ls ordered
$ $HADOOP_HOME/bin/hadoop fs -tail ordered/part-r-00000
```

WordCount usando Pig

1. Baixe o arquivo stop-word-list.csv e coloque no HDFS

```
$ cd /vagrant
$ wget -c
https://raw.githubusercontent.com/GerInfraBigDataPUCRS/CursoDataScience2018/master/06_Pig/stop-word-list.csv
$ $HADOOP_HOME/bin/hadoop fs -mkdir stopwords
$ $HADOOP_HOME/bin/hadoop fs -put stop-word-list.csv stopwords
```

2. Inicie uma sessão no grunt

```
$ $PIG_HOME/bin/pig
```

3. Utilize o comando SET para renomear o nome do programa (nome visto no YARN ResourceManager UI)

```
SET job.name 'Word Count in Pig';
```

4. Carregue o dataset Shakespeare (usado no HandsOn de MapReduce)

```
shakespeare = LOAD 'shakespeare' AS (lineoftext:chararray);
```

5. Carregue o dataset stopwords

```
stopwords = LOAD 'stopwords' USING PigStorage()
            AS (stopword:chararray);
```

6. Use os comandos TOKENIZE, FLATTEN e normalize o texto (alterando todas as letras para minúsculas) no dataset Shakespeare, criando uma bag de palavras

```
words = FOREACH shakespeare GENERATE
        FLATTEN(TOKENIZE(REPLACE(LOWER(TRIM(lineoftext)),
        '[\\p{Punct},\\p{Cntrl}]','''))) AS word;
```

7. Remova as palavras vazias.

```
realwords = FILTER words BY SIZE(word) > 0;
```

8. Use os comandos TOKENIZE e FLATTEN no dataset stopwords

```
flattened_stopwords = FOREACH stopwords GENERATE
       FLATTEN(TOKENIZE(stopword)) AS stopword;
```

9. Execute um RIGHT OUTER JOIN entre os datasets stopwords e realwords.

```
right_joined = JOIN flattened_stopwords
               BY stopword RIGHT OUTER,
               realwords BY word;
```

10. Remova os stopwords

```
meaningful_words = FILTER right_joined BY
          (flattened_stopwords::stopword IS NULL);
```

11. Remova entradas duplicadas

```
shakespeare_real_words = FOREACH meaningful_words
          GENERATE realwords::word AS word;
```

12. Agrupe as palavras

```
grouped = GROUP shakespeare_real_words BY word;
```

13. Conte as palavras com a função COUNT

```
counted = FOREACH grouped GENERATE group AS word,
          COUNT(shakespeare_real_words) AS wordcount;
```

14. Ordene o bag counted com o operador ORDER

```
ordered = ORDER counted BY wordcount;
```

15. Verifique a saída com o commando DUMP

```
DUMP ordered;
```