TASK: parallel grep Facebook posts Log

- **TASK find all occurrences of a word, say, "apple" in the Facebook posts log file**
- **In Unix solve using**
  - *grep "apple" facebook.log*

  [2007/12/25 09:15] Ate an apple
  [2008/02/12 12:37] Thought about apples
  [2009/01/09 19:55] MMmmm.. Apples
- **Assume 1 Terabyte of log files a single core will take about 5 hours to grep the entire file**
- **How can you bring this down to a minute or two using 100 core machine?**
- **Using the Unix map-reduce framework:**
  - Given a mapper and a reducer function in python fill in the blanks

Challenge

- **Fill in the blanks with comments and python code in the following to get this version of the parallel grep to work:**
  - Mapper
  - Reducer

Python Notebook
Mapper and reducers in Command Line

```
%%writefile pGrepCount.sh
ORIGINAL_FILE=$1
FIND_WORD=$2
BLOCK_SIZE=$3
CHUNK_FILE_PREFIX=$ORIGINAL_FILE.split
SORTED_CHUNK_FILES=$CHUNK_FILE_PREFIX*.sorted
usage()
{
    echo Parallel grep
    echo usage: pGrepCount filename word chuncksize
    echo greps file file1 in $ORIGINAL_FILE and counts the number of lines
    echo Note: file1 will be split in chunks up to $ BLOCK_SIZE chunks each
    echo $FIND_WORD each chunk will be grepCounted in parallel
}
#Splitting $ORIGINAL_FILE INTO CHUNKS
split -b $BLOCK_SIZE $ORIGINAL_FILE $CHUNK_FILE_PREFIX
#DISTRIBUTE
for file in $CHUNK_FILE_PREFIX*
do
    grep -i $FIND_WORD $file|wc -l >$file.intermediateCount &
done
wait
#MERGEING INTERMEDIATE COUNT CAN TAKE THE FIRST COLUMN AND TOTOL...
numOfInstances=$(cat *.intermediateCount | cut -f 1 | paste -sd+ - |bc)
echo "found [$numOfInstances] occurences of [$FIND_WORD] in the file [$ORIGINAL_FILE]"
```

Mapper

Reducer

```
Writing pGrepCount.sh
```

Run the file

```
!chmod a+x pGrepCount.sh
```

```
Usage: usage: pGrepCount filename word chuncksize
```

```
!./pGrepCount.sh License COPYRIGHT 4k
```

```
found [57] occurences of [COPYRIGHT] in the file [License]
```

Mapper for grep

```
1  %%writefile mapper.py
2  #!/usr/bin/python
3  import sys
4  import re
5  count = 0
6  filename = sys.argv[2]
7  findword = sys.argv[1]
8  with open (filename, "r") as myfile:
9      for line in myfile.readlines():
10
11
12 print count
```

```
Overwriting mapper.py
```

```
1 !chmod a+x mapper.py
```

Read each line
Increment counter if desired is present in the line

1 of 2: Fill in the blanks

Python Notebook Mapper and reducers in Python!!

Reducer for grep

```
1  %%writefile reducer.py
2  #!/usr/bin/python
3  import sys
4  sum = 0
5  for countStr in sys.stdin:
6
7  print sum
```

```
Overwriting reducer.py
```

```
1 !chmod a+x reducer.py
```

Read mapper output (single number per line)
Sum up
Output total count

2 of 2: Fill in the blanks

Map-Reduce framework in the Comm

Python Notebook
Mapper and reducers in Python!!

```
%%writefile pGrepCount.sh
ORIGINAL_FILE=$1
FIND_WORD=$2
BLOCK_SIZE=$3
CHUNK_FILE_PREFIX=$ORIGINAL_FILE.split
SORTED_CHUNK_FILES=$CHUNK_FILE_PREFIX*.sorted
usage()
{
    echo Parallel grep
    echo usage: pGrepCount filename word chuncksize
    echo greps file file1 in $ORIGINAL_FILE and counts the number of lines
    echo Note: file1 will be split in chunks up to $ BLOCK_SIZE chunks each
    echo $FIND_WORD each chunk will be grepCounted in parallel
}
#Splitting $ORIGINAL_FILE INTO CHUNKS
split -b $BLOCK_SIZE $ORIGINAL_FILE $CHUNK_FILE_PREFIX
#DISTRIBUTE
for file in $CHUNK_FILE_PREFIX*
do
    #grep -i $FIND_WORD $file|wc -l >$file.intermediateCount &
    ./mapper.py $FIND_WORD $file >$file.intermediateCount &
done
wait
#MERGEING INTERMEDIATE COUNT CAN TAKE THE FIRST COLUMN AND TOTOL...
#numOfInstances=$(cat *.intermediateCount | cut -f 1 | paste -sd+ - |bc)
numOfInstances=$(cat *.intermediateCount | ./reducer.py)
echo "found [$numOfInstances] occurences of [$FIND_WORD] in the file [$ORIGINAL_FILE]"
```

Mapper

Reducer

```
Overwriting pGrepCount.sh
```

Run the parallel grep using python-based mapper and reducer

```
!chmod a+x pGrepCount.sh
```

```
Usage: usage: pGrepCount filename word chuncksize
```

```
!./pGrepCount.sh License COPYRIGHT 4k
```

```
found [57] occurences of [COPYRIGHT] in the file [License]
```