MIDS-W261-2015-AsyncQuiz-Week01-Velamur-Python-CLI-MR

January 14, 2016

1 DATASCI W261: Machine Learning at Scale

2 This notebook provides a poor man Hadoop through command-line and python.

3 Completed mapper.py and reducer.py

4 Map

```
In [58]: %%writefile mapper.py
         #!/usr/bin/python
         import sys
         import re
         count = 0
         WORD_RE = re.compile(r"[\w’]+")
         filename = sys.argv[2]
         findword = sys.argv[1]
         allWords = []
         with open (filename, "r") as myfile:
             allWords = WORD_RE.findall(myfile.read())
         allWords = [word.lower() for word in allWords]
         print allWords.count(findword.lower())
```

```
Writing mapper.py
```

```
In [59]: !chmod a+x mapper.py
```

5 Reduce

```
In [60]: %%writefile reducer.py
         #!/usr/bin/python
         import sys
         sum = 0
         for line in sys.stdin:
             sum += int(line)
         print sum
```

```
Writing reducer.py
```

```
In [61]: !chmod a+x reducer.py
```

6 Write script to file

```
In [1]: %%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 [$numOfInstances] [$FIND_WORD] in the file [$ORIGINAL_FILE]"
```

```
Overwriting pGrepCount.sh
```

7 Run the file

```
In [63]: !chmod a+x pGrepCount.sh
```

7.0.1 Usage: pGrepCount filename word chuncksize

Interesting: I see the word Copyright occuring 59 times in license.txt, but in the grep method, only 57 occurences are counted. However, a word-count done by a text editor returned 59 as the count. Will have to compare the splits between parallel grep and python to see where this discrepancy arises.

```
In [64]: !./pGrepCount.sh License.txt COPYRIGHT 4k
```

```
found [59] [COPYRIGHT] in the file [License.txt]
```

```
In [ ]:
```