210701120

Word Count Using Map Reduce

Input(word_count_data.txt):

```
hadoop@jeff-VirtualBox:~/Documents$ cat word_count_data.txt
Jeff Would Never Give up
They installed Hadoop
And Tested hadoop Successfully
Jeff is Very Happy
Hope Jeff will Succeed
To do Map Reduce
hadoop@jeff-VirtualBox:~/Documents$
```

mapper.py:

Open — mapper.py — ~/Documents

```
#!/usr/bin/env python

# import sys because we need to read and write data to STDIN and STDOUT
import sys

# reading entire line from STDIN (standard input)
for line in sys.stdin:
    # to remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()

    # we are looping over the words array and printing the word
    # with the count of 1 to the STDOUT
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        print ('%s\t%s' % (word, 1))
```

reducer.py:

```python
from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# read the entire line from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # splitting the data on the basis of tab we have provided in mapper.py
    word, count = line.split('\t', 1)
    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print ('%s\t%s' % (current_word, current_count))
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    print ('%s\t%s' % (current_word, current_count))
```

Output:

```
hadoop@jeff-VirtualBox:~/hadoop$ hdfs dfs -cat /word_count_in_python/new_output3/part-00000
And     1
Give    1
Hadoop  1
Happy   1
Hope    1
Jeff    3
Map     1
Never   1
Reduce  1
Succeed 1
Successfully    1
Tested  1
They    1
To      1
Very    1
Would   1
do      1
hadoop  1
installed       1
is      1
up      1
will    1
hadoop@jeff-VirtualBox:~/hadoop$
```