Discussion:

As is shown above, it takes about **1.63 sec** to find the top 10 hashtags using map reduce approach in Python, while it only takes **0.25 sec** using Unix command. I think this is because Python is an interpreted language and it runs much slower than shell command.

1.2 (1)

In [12]:

```
# Read txt
with open("tweets.txt","r", encoding='utf-8') as file:
    lines = [next(file) for x in range(750000)]
```

In [4]:

```
import re

def extractUsernames(string):
    pattern = re.compile(r"@(\S+)")
    strs = re.findall(pattern, string)

    output = []
    for i in strs:
        output.append(i)
    return output

# Example:
extractUsernames("22077441      10470781081     @Confession.    I can't live with my mama!!! Especially if I don't have my own car!      2010-03-14 09:21:58")
```

Out[4]:

```
['Confession.']
```

In [5]:

```
def mapper_usernames_line(line):
    words = extractUsernames(line)
    output = []
    for word in words:
        if word:
            output.append((word,1))
    return output

# Example:
mapper_usernames_line("22077441 10470781081     @Confession.    I can't live with my mama!!! Especially if I don't have my own car!      2010-03-14 09:21:58")
```

Out[5]:

```
[('Confession.', 1)]
```

In [6]:

```
def mapper_usernames(lines):
    output = []
    for line in lines:
        list = mapper_usernames_line(line)
        if list:
            output += list
    return output

#Example:
test = ["@John. 2010", "@Jerry 2011", "@Tom 2012", "@Jerry 2013"]
mapper_usernames(test)
```

Out[6]:

```
[('John.', 1), ('Jerry', 1), ('Tom', 1), ('Jerry', 1)]
```

In [7]:

```
def combiner_usernames(mapper_output):
    groups = {} # group by key values
    for item in mapper_output:
        k = item[0]
        v = item[1]
        if k not in groups:
            groups[k] = [v]
        else:
            groups[k].append(v)
    return groups

#Example:
combiner_usernames(mapper_usernames(test))
```

Out[7]:

```
{'John.': [1], 'Jerry': [1, 1], 'Tom': [1]}
```

In [8]:

```
def reducer_usernames(keyWord, counts):
    return (keyWord, sum(counts))

reducer_usernames('jerry',[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

Out[8]:

```
('jerry', 14)
```

In [14]:

```
def execute_usernames(lines):
    groups = combiner_usernames(mapper_usernames(lines))
    output = [reducer_usernames(k,v) for k,v in groups.items()]
    output.sort()
    return output

usernames_freq = execute_usernames(lines)
```

In [15]:

```
def Sort(orig):

    orig.sort(key = lambda x: x[1], reverse = True)
    return orig

print(Sort(usernames_freq)[:10])
```

```
[('RevRunWisdom:', 1234), ('listensto', 939), ('DonnieWahlberg', 525), ('OGmuscle
s', 441), ('addthis', 429), ('breatheitin', 411), ('justinbieber', 354), ('MAV25',
347), ('karlievoice', 305), ('mtgcolorpie', 291)]
```

In [16]:

```
import timeit

start = timeit.default_timer()
usernames_freq = execute_usernames(lines)
print(Sort(usernames_freq)[:10])
stop = timeit.default_timer()
print('Time: ', stop - start)
```

```
[('RevRunWisdom:', 1234), ('listensto', 939), ('DonnieWahlberg', 525), ('OGmuscle
s', 441), ('addthis', 429), ('breatheitin', 411), ('justinbieber', 354), ('MAV25',
347), ('karlievoice', 305), ('mtgcolorpie', 291)]
Time:  3.0066348999999946
```