Discussion:

As is shown above, it takes about **3.01 sec** to find the top 10 hashtags using map reduce approach in Python, while it only takes **1.12 sec** using Unix command.

1.2 (2)

In [27]:

```
import re

def extractTwohashtags(string):
    pattern = re.compile(r"#\S+#\S+")
    strs = re.search(pattern, string)
    output = []
    if strs:
        output.append(strs.group(0))
    else:
        output.append([])
    return output

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

```
[[]]
['#Confession#Disappointment#Desperation.']
```

In [18]:

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

# Example:
mapper_twohashtags_line("22077441	10470781081	#Confession#Disappointment	I can't liv
e with my mama!!! Especially if I don't have my own car!	2010-03-14 09:21:58")
```

Out[18]:

```
[('#Confession#Disappointment', 1)]
```

In [19]:

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

#Example:
test = ["#John.#2010", "#Jerry#2013", "#Tom2012", "#Jerry#2013"]
mapper_twohashtags(test)
```

Out[19]:

```
[('#John.#2010', 1), ('#Jerry#2013', 1), ('#Jerry#2013', 1)]
```

In [20]:

```
def combiner_twohashtags(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_twohashtags(mapper_twohashtags(test))
```

Out[20]:

```
{'#John.#2010': [1], '#Jerry#2013': [1, 1]}
```

In [21]:

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

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

Out[21]:

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

In [22]:

```
def execute_twohashtags(lines):
    groups = combiner_twohashtags(mapper_twohashtags(lines))
    output = [reducer_twohashtags(k,v) for k,v in groups.items()]
    output.sort()
    return output

twohashtags_freq = execute_twohashtags(lines)
```

In [23]:

```
def Sort(orig):

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

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

```
[('#affiliate#marketing', 8), ('####', 5), ('#Celebrity,#Philanthropy', 4), ('#39;
Green&#39;', 3), ('#39;What&#39;s', 3), ('#39;streaming&#39;', 3), ('#??PFoundersd
ay#??PFoundersday', 3), ('#39;A&#39;', 2), ('#39;SNL&#39;:', 2), ('#39;Twilight&#3
9;', 2)]
```

In [24]:

```
import timeit

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

```
[('#affiliate#marketing', 8), ('####', 5), ('#Celebrity,#Philanthropy', 4), ('#39;
Green&#39;', 3), ('#39;What&#39;s', 3), ('#39;streaming&#39;', 3), ('#??PFoundersd
ay#??PFoundersday', 3), ('#39;A&#39;', 2), ('#39;SNL&#39;:', 2), ('#39;Twilight&#3
9;', 2)]
Time:  1.9740761999999847
```