Discussion:

As is shown above, it takes about **1.97 sec** to find the top 10 hashtags using map reduce approach in Python, while it only takes **0.18 sec** using Unix command.

2 Finding Reciprocal Followers

In [30]:

```
import pandas as pd

edges_orig = pd.read_csv("./Twitter-dataset/data/edges.csv")
edges = edges_orig.head(500000)
```

In [33]:

```
edges_test = edges_orig.head(1000)
```

In [34]:

```
def mapper_reciprocal(df):
    return list(map(list, df.values))

# Example:
mapper_reciprocal(edges_test)[:20]
```

Out[34]:

```
[[1, 8762940],
 [1, 8762941],
 [1, 688136],
 [1, 8762942],
 [3, 718952],
 [3, 3109655],
 [3, 562897],
 [3, 6],
 [3, 7],
 [3, 12852],
 [3, 90259],
 [3, 8762941],
 [3, 645510],
 [3, 427258],
 [3, 45567],
 [3, 1374301],
 [3, 38253],
 [3, 79994],
 [3, 16],
 [3, 9]]
```

In [35]:

```
def combiner_reciprocal(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_reciprocal(mapper_reciprocal(edges_test))
```

In [36]:

```
def reducer_reciprocal(userID, followingID, group):
    if followingID in group:
        if userID in group[followingID]:
            return (userID, followingID)

#Example:
g = {1:[2,3],2:[1],3:[2,4]}
reducer_reciprocal(1, 2, g)
```

Out[36]:

(1, 2)

In [37]:

```
def execute_reciprocal(edges):
    map_reciprocal = mapper_reciprocal(edges)
    groups = combiner_reciprocal(map_reciprocal)
    output = []
    for users in map_reciprocal:
            pair =  reducer_reciprocal(users[0],users[1], groups)
            if pair:
                output.append(pair)
    output.sort()
    return output

output = execute_reciprocal(edges)
```

In [38]:

```
import timeit

start = timeit.default_timer()
execute_reciprocal(edges)
stop = timeit.default_timer()
print('Time: ', stop - start)
```

Time: 1.2204441000000088

In [39]:

```
follower_graph = pd.DataFrame(output, columns =['userID', 'followerID'])
follower_graph.to_csv('follower_graph.csv', index=False)
follower_graph
```

Out[39]:

	userID	followerID
0	3682	5276
1	5276	3682
2	13232	18205
3	13232	63255
4	15574	15926
5	15926	15574
6	18205	13232
7	19628	19821
8	19628	20033
9	19821	19628
10	20033	19628
11	22196	76473
12	23503	41422
13	31866	32002
14	32002	31866
15	32173	32452
16	32452	32173
17	33099	62167
18	33884	34046
19	33884	34101
20	34046	33884
21	34101	33884
22	40704	40997
23	40704	41039
24	40997	40704
25	40997	41039
26	40997	62623
27	40997	201063
28	41039	40704
29	41039	40997
...	...	...
38	65411	65435
39	65435	63255
40	65435	65411
41	65435	93260
42	70696	60887
43	70696	70772

	userID	followerID
44	70772	70696
45	76473	22196
46	78182	78464
47	78464	78182
48	80092	80096
49	80096	80092
50	89222	89350
51	89350	89222
52	93260	65435
53	93260	93427
54	93427	93260
55	100591	100721
56	100721	100591
57	102898	122546
58	122546	102898
59	134409	134410
60	134410	134409
61	135546	135684
62	135684	135546
63	192865	192899
64	192899	192865
65	201063	40997
66	201078	201607
67	201607	201078

68 rows × 2 columns