SPARKIFY 4 – Parallelizing using IPython

<<TASK 0>> Create an instance; if not already exists – Refer sparkify 1 Task 0

you can either follow this tutorial, or the video tutorial – both are different; and both should work

<<TASK 1>> Set-up IPython parallel

0. SSH into the instance [existing or new]
1. Use super user mode [sudo su]
2. Update all packages [yum update] – will update if there are any to be updated
3. Activate python virtual env [if doesn't exist; refer subtask in Sparkify 1]
   source venv/bin/activate
4. Install the required packages – make note that there were no errors [warnings can be ignored]
   pip install pyyaml ipython jupyter ipyparallel pandas boto -U
5. Enable IPython Cluster
   ipcluster nbextension enable
6. Start an ipcluster with 4 engines
   ipcluster start -n 4
   [DO NOT stop/interrupt the process – otherwise you will be unable to finish this assignment]
7. Let the previous SSH terminal be as it is and start a new session (SSH again to the instance)
   source venv/bin/activate
   jupyter notebook --port=8888 --no-browser --ip=0.0.0.0
8. Add port 8888 in the inbound rules for that instance to allow access to Anywhere IPv4
   - Go to instance
   - Select security tab -> click on the security group
   - Choose add inbound rules
   - Port: 8888; source – anywhere Ipv4
   - Save rules
9. From your browser; go to url – instancePublicDNS:8888 OR instancePublicIPv4:8888
10. When prompted for token, use the value from the terminal where you started jupyter notebook

```
[C 16:26:11.610 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/ec2-user/.local/share/jupyter/runtime/nbserver-3486-open.html
    Or copy and paste one of these URLs:
        http://ip-172-31-0-252.ec2.internal:8888/?token=1d752c74ec3462633df554d0ea048704c8aa83b8226a30cf
     or http://127.0.0.1:8888/?token=1d752c74ec3462633df554d0ea048704c8aa83b8226a30cf
[I 16:29:52.231 NotebookApp] 302 GET / (73.219.136.147) 0.560000ms
[I 16:29:52.278 NotebookApp] 302 GET /tree? (73.219.136.147) 0.810000ms
[I 16:30:48.381 NotebookApp] 302 GET / (73.219.136.147) 0.540000ms
[I 16:30:48.436 NotebookApp] 302 GET /tree? (73.219.136.147) 0.740000ms
[I 16:30:54.561 NotebookApp] 302 POST /login?next=%2Ftree%3F (73.219.136.147) 1.080000ms
```

11. You should be able to see a jupyter notebook homepage

<<TASK 2>> finish the sparkify task using parallel computing

0. Make sure to add the AWS CLI to your instance [refer sparkify 3 tutorial – Task 0 steps 3 to 7]

Sparkify4.ipynb:

```
#ln[1]
from ipyparallel import Client
rc = Client()
print('Number of clusters running =', len(rc))
print('Client ids are:', rc.ids)
dview = rc[:]
```

```
#op[1]
 Number of clusters running = 4

 Client ids are: [0, 1, 2, 3]
```

```
#ln[2]
import boto3
s3 = boto3.resource('s3')
my_bucket = s3.Bucket('dsci6007yshah')
all_keys = []
for bucket_obj in my_bucket.objects.all():
    all_keys.append(bucket_obj.key)
print('Total number of json objects =', len(all_keys))
```

```
#op[2]
Total number of json objects = 8056
```

```
#ln[3]
%%px
def test1(keys):
    import json
    import boto3
    from collections import Counter
    bucket = 'dsci6007yshah'
    artistCounter = {}
    songCounter = {}
    s3 = boto3.client('s3')
    for key in keys:
        obj = s3.get_object(Bucket=bucket,Key=str(key))
        obj = json.loads(obj['Body'].read())
        try:
            #Avoid keeping count for 'None' artist
            if obj['artist']:
                artistCounter[str(obj['artist'])] += 1
        except:
            artistCounter[str(obj['artist'])] = 1
        try:
            #Avoid keeping count for 'None' song
            if obj['song']:
                songCounter[str(obj['song'])] += 1
        except:
            songCounter[str(obj['song'])] = 1
    return Counter(artistCounter), Counter(songCounter)
```

```
#In[4]
import time
start = time.perf_counter()
dview.scatter('keys',all_keys)
#%px print(keys[:5])
%px y = [test1(keys)]
y = dview.gather('y')
print('Time taken to get all results = {:.4f}s'.format(time.perf_counter() -
start))
```

#op[4]

%px: 100%|||||||||| 4/4 [01:40<4, 4tasks/s]
Time taken to get all results = 64.5747s

```
#In[5]
from collections import Counter
import pandas as pd
artistCounter = Counter({})
songCounter = Counter({})
for (artists, songs) in y:
    artistCounter += artists
    songCounter += songs

artistCounter = pd.DataFrame(artistCounter.most_common(10),
                  columns=['Artist Name', 'Count'], index=range(1, 11))

songCounter = pd.DataFrame(songCounter.most_common(10),
                  columns=['Song Name', 'Count'], index=range(1, 11))
```

```
#In[6]
pd.set_option('display.max_colwidth', None)
print('Top 10 Artists are:')
display(artistCounter)

print('\n\nTop 10 Songs are:')
display(songCounter)
```

#op[6]

```
Top 10 Artists are:
```

	Artist Name	Count
1	Coldplay	58
2	Kings Of Leon	55
3	Dwight Yoakam	38
4	The Black Keys	36
5	Jack Johnson	35
6	Muse	35
7	Florence + The Machine	35
8	BjÃ□Â¶rk	33
9	The Killers	31
10	John Mayer	31

```
Top 10 Songs are:
```

	Song Name	Count
1	You're The One	37
2	Undo	28
3	Revelry	27
4	Sehr kosmisch	21
5	Horn Concerto No. 4 in E flat K495: II. Romance (Andante cantabile)	19
6	Canada	17
7	Secrets	17
8	Dog Days Are Over (Radio Edit)	16
9	ReprÃ□Â©sente	14
10	Invalid	14