DATASCI W261: Machine Learning at Scale

- **Sayantan Satpati**
- **sayantan.satpati@ischool.berkeley.edu**
- **W261**
- **Week-0**
- **Assignment-1**
- **Date of Submission: 28-AUG-2015**

This notebook provides a poor man Hadoop through command-line and python. Please insert the python code by yourself. ¶

Map

In [30]:

```
%%writefile mapper.py
#!/usr/bin/python
import sys
import re
count = 0
filename = sys.argv[2]
findword = sys.argv[1]
with open (filename, "r") as myfile:
    for line in myfile:
        # Case Insensitive Regex Search for the word in the line
        match = re.search("\\b" + findword + "\\b", line, re.IGNORECASE)
        if match:
            count += 1

# Print count from each mapper
print count
```

```
Overwriting mapper.py
```

In [31]:

```
!chmod a+x mapper.py
```

Reduce

In [32]:

```
%%writefile reducer.py
#!/usr/bin/python
import sys
sum = 0
for line in sys.stdin:
    # Sum the counts across all mappers
    sum += int(line)
# Final Sum
print sum
```

```
Overwriting reducer.py
```

In [33]:

```
!chmod a+x reducer.py
```

In [34]:

```
# Remove split files from last runs
! rm License.txt.*
```

Write script to file

In [35]:

```
%%writefile pGrepCount.sh
ORIGINAL_FILE=$1
FIND_WORD=$2
BLOCK_SIZE=$3
CHUNK_FILE_PREFIX=$ORIGINAL_FILE.split
SORTED_CHUNK_FILES=$CHUNK_FILE_PREFIX*.sorted
usage()
{
    echo Parallel grep
    echo usage: pGrepCount filename word chuncksize
    echo greps file file1 in $ORIGINAL_FILE and counts the number of lines
    echo Note: file1 will be split in chunks up to $ BLOCK_SIZE chunks each
    echo $FIND_WORD each chunk will be grepCounted in parallel
}
#Splitting $ORIGINAL_FILE INTO CHUNKS
split -b $BLOCK_SIZE $ORIGINAL_FILE $CHUNK_FILE_PREFIX
#DISTRIBUTE
for file in $CHUNK_FILE_PREFIX*
do
    #grep -i $FIND_WORD $file|wc -l >$file.intermediateCount &
    ./mapper.py $FIND_WORD $file >$file.intermediateCount &
done
wait
#MERGEING INTERMEDIATE COUNT CAN TAKE THE FIRST COLUMN AND TOTOL...
#numOfInstances=$(cat *.intermediateCount | cut -f 1 | paste -sd+ - |bc)
numOfInstances=$(cat *.intermediateCount | ./reducer.py)
echo "found [$numOfInstances] [$FIND_WORD] in the file [$ORIGINAL_FILE]"
```

```
Overwriting pGrepCount.sh
```

Run the file

```
In [36]: !chmod a+x pGrepCount.sh
```

Usage: usage: pGrepCount filename word chuncksize

```
In [37]: !./pGrepCount.sh License.txt COPYRIGHT 4k

         found [57] [COPYRIGHT] in the file [License.txt]
```

```
In [ ]:
```