Data Intensive Computing
Lab 2: Data Aggregation, Big Data Analysis and Visualization

Arvind Thirumurugan - athirumu - 50289656
Vijay Jagannathan - vijayjag - 50290947

Outline:

This project aims at analyzing, comparing data collected from different sources and visualizing the outcome in the form of word clouds.

- We Chose Sports as the topic of interest and collected data from 3 data sources:-
    - NY Times - 500 articles (100 articles for each subtopic).
    - Twitter - 20000 tweets.
    - Common Crawl - 500 articles (100 articles for each subtopic).

- Setup big data infrastructure using the Hadoop Docker image:-
    - Placed the input files in .txt format in the mapped directory in the host machine.
    - Moved the input files to the hadoop infrastructure and ran the mapper and reducer files to get the require output.
    - Moved the output files from the hadoop infrastructure back to the host machine and used Tableau to visualize the word count and Co-occurrence for subtopic.

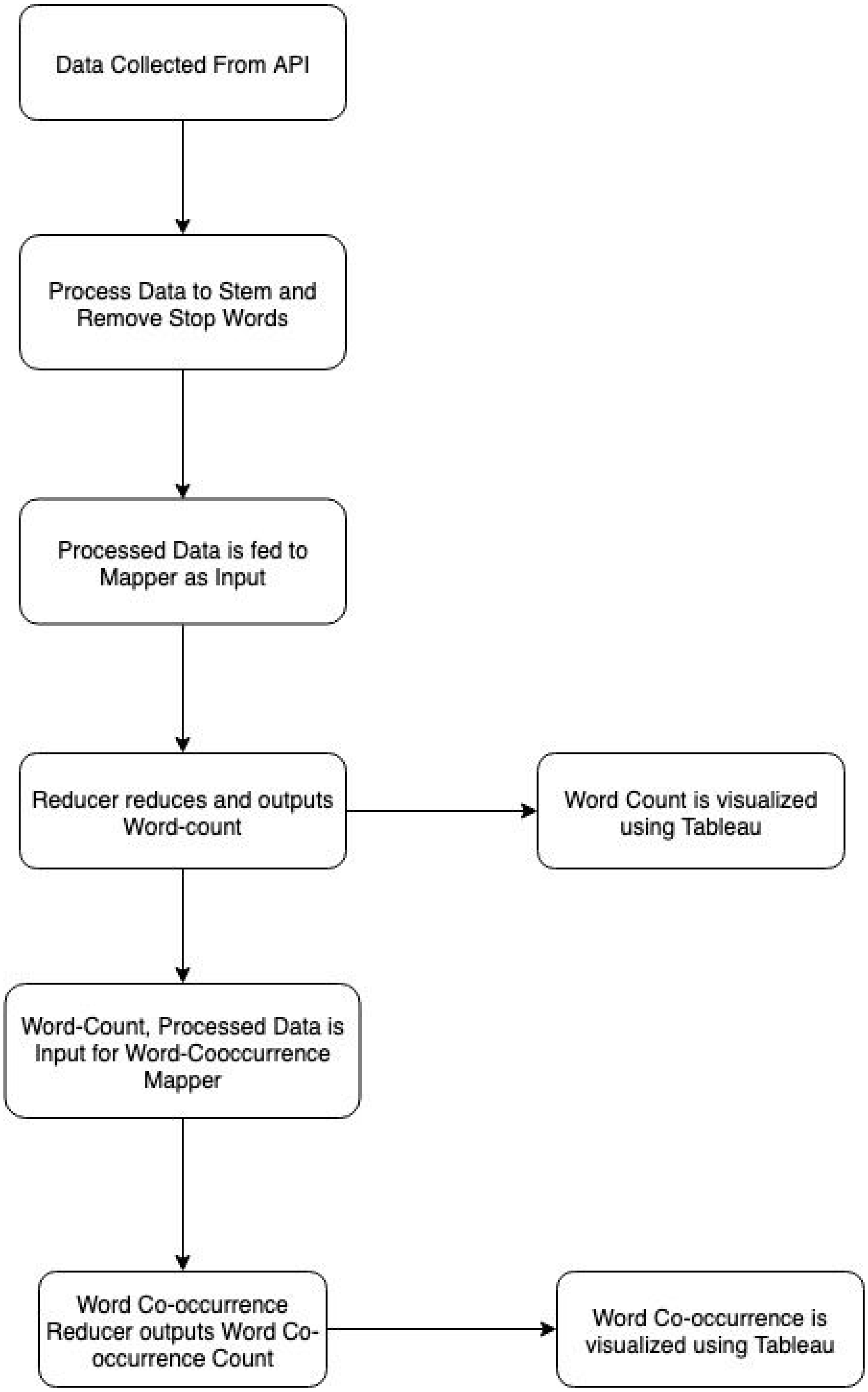
Map-Reduce Algorithm:

Before running the Mapper and Reducer files all the input data must be processed i.e. all the input data must be stemmed and all stop words must be removed this achieved by using a python script which performs pre-processing on raw input data. The mapper.py , reducer.py and required input are moved to the Hadoop file system where the corresponding command is used to run Mapper and Reducer.

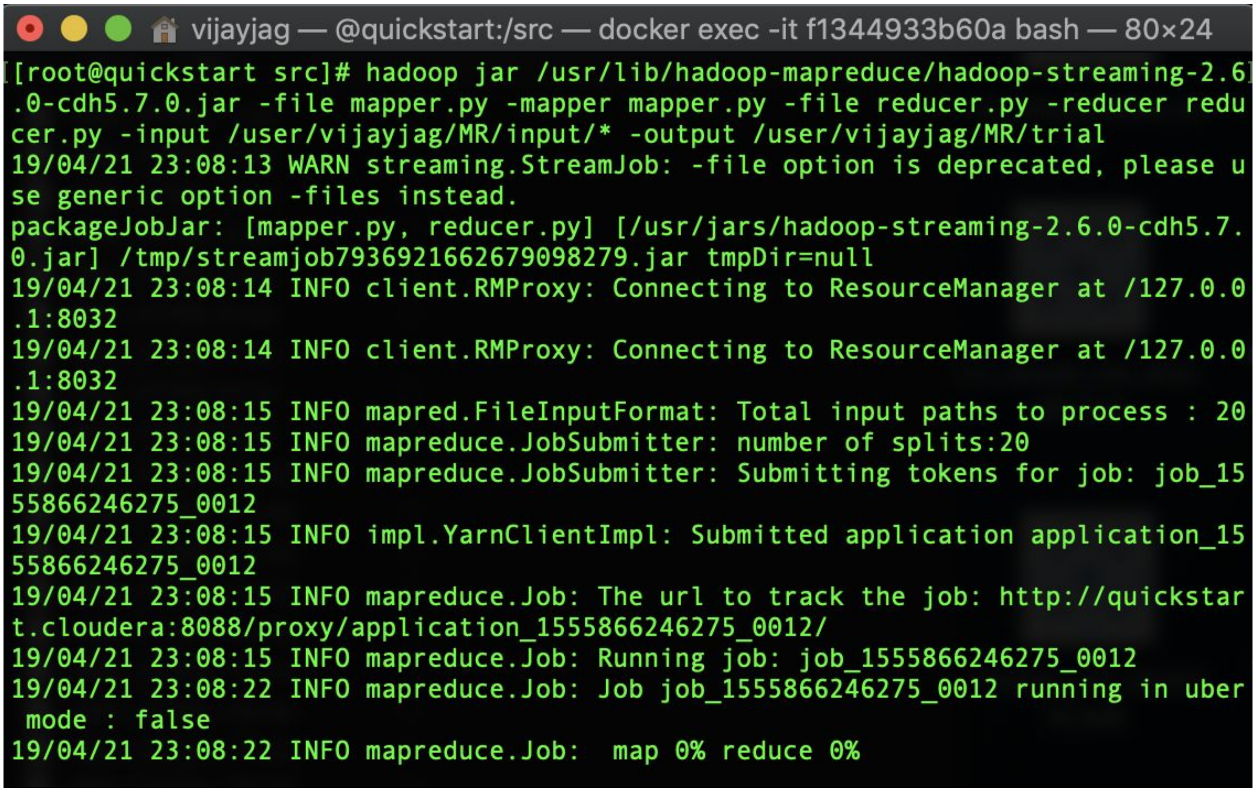
The Mapper file for Word-Count takes each word from the input file and emit <word,1> as output, this is the input for the reducer file which adds the count for all words and outputs the results. The Mapper file for Word Co-occurrence takes the top ten words from the Word Count output and input file and emits <word,neighbor,1> as output and the reducer file returns the count for each Co-occurrence.

The Output files generated by the reducer is moved from the Hadoop filesystem to the host system then the output from the Word count Reducer and Word co-occurrence Reducer are used to visualize data using Tableau and the necessary graphs are obtained.

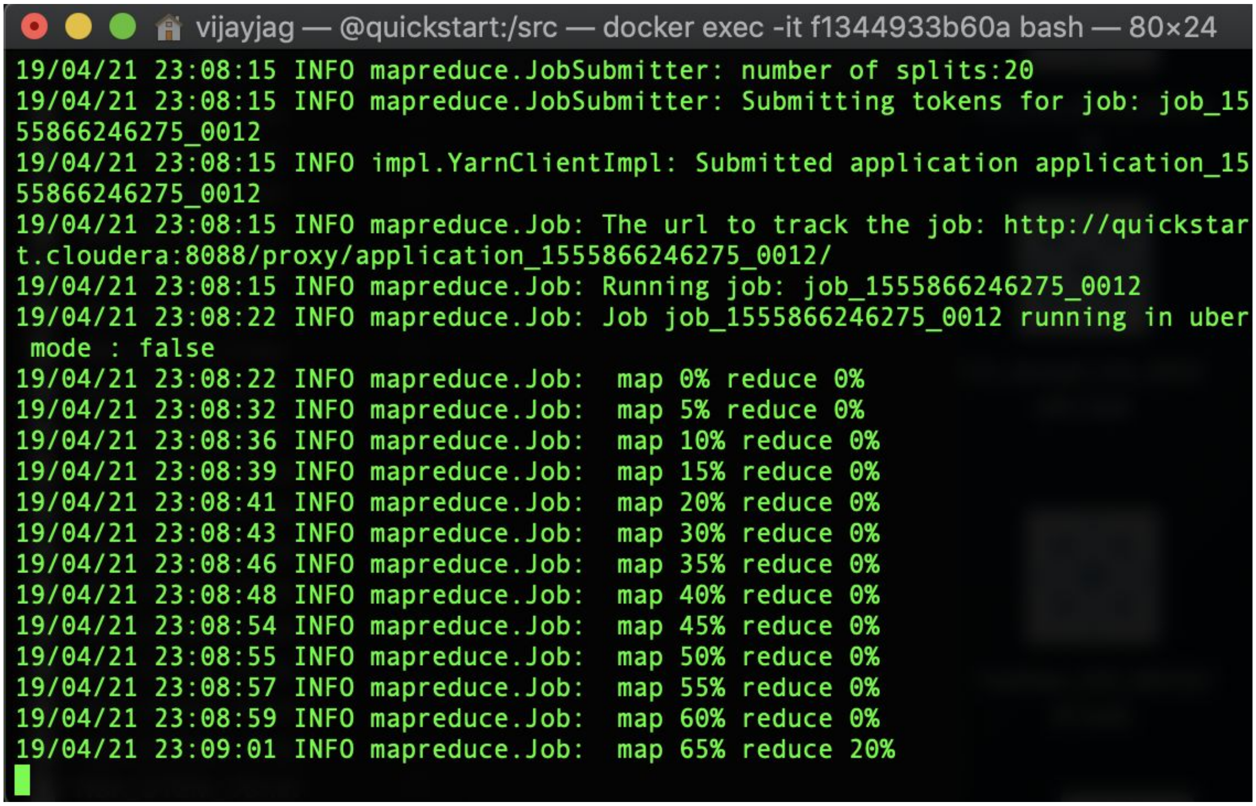
Data Pipeline for Hadoop Infrastructure:


Data Collected From API
Process Data to Stem and Remove Stop Words
Processed Data is fed to Mapper as Input
Reducer reduces and outputs Word-count
Word Count is visualized using Tableau
Word-Count, Processed Data is Input for Word-Cooccurrence Mapper
Word Co-occurrence Reducer outputs Word Co-occurrence Count
Word Co-occurrence is visualized using Tableau

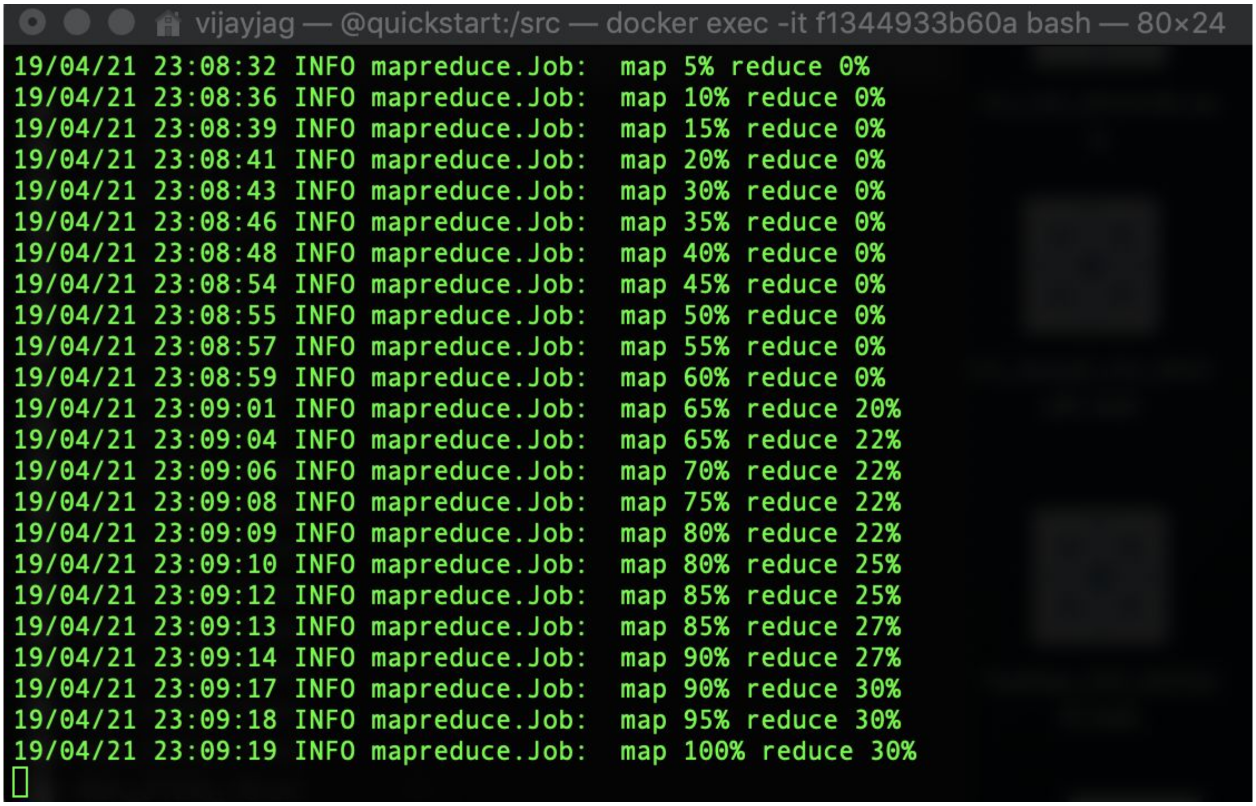
Screenshots of running Mapper and Reducer:


vijayjag — @quickstart:/src — docker exec -it f1344933b60a bash — 80×24

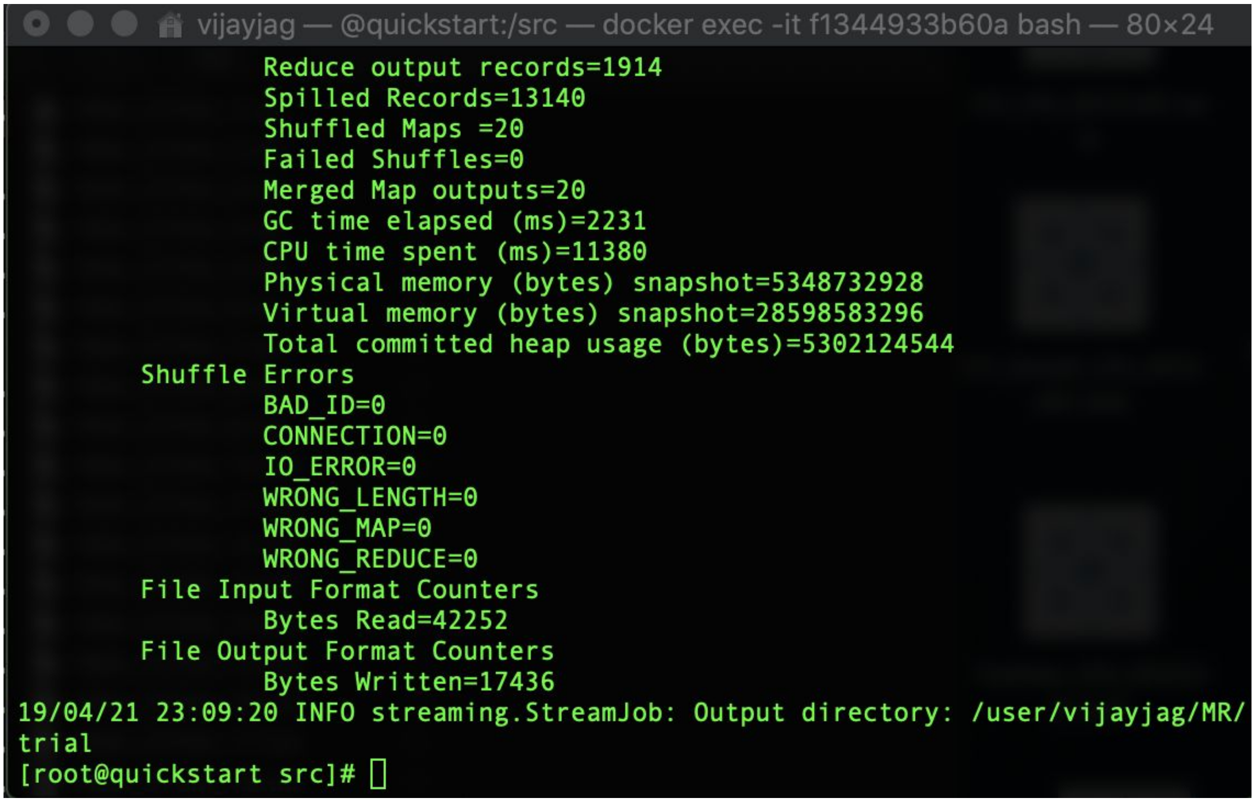
```
[[root@quickstart src]# hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming-2.6]
.0-cdh5.7.0.jar -file mapper.py -mapper mapper.py -file reducer.py -reducer redu
cer.py -input /user/vijayjag/MR/input/* -output /user/vijayjag/MR/trial
19/04/21 23:08:13 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [mapper.py, reducer.py] [/usr/jars/hadoop-streaming-2.6.0-cdh5.7.
0.jar] /tmp/streamjob7936921662679098279.jar tmpDir=null
19/04/21 23:08:14 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0
.1:8032
19/04/21 23:08:14 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0
.1:8032
19/04/21 23:08:15 INFO mapred.FileInputFormat: Total input paths to process : 20
19/04/21 23:08:15 INFO mapreduce.JobSubmitter: number of splits:20
19/04/21 23:08:15 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_15
55866246275_0012
19/04/21 23:08:15 INFO impl.YarnClientImpl: Submitted application application_15
55866246275_0012
19/04/21 23:08:15 INFO mapreduce.Job: The url to track the job: http://quickstar
t.cloudera:8088/proxy/application_1555866246275_0012/
19/04/21 23:08:15 INFO mapreduce.Job: Running job: job_1555866246275_0012
19/04/21 23:08:22 INFO mapreduce.Job: Job job_1555866246275_0012 running in uber
 mode : false
19/04/21 23:08:22 INFO mapreduce.Job:  map 0% reduce 0%
```


vijayjag — @quickstart:/src — docker exec -it f1344933b60a bash — 80×24

```
19/04/21 23:08:15 INFO mapreduce.JobSubmitter: number of splits:20
19/04/21 23:08:15 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_15
55866246275_0012
19/04/21 23:08:15 INFO impl.YarnClientImpl: Submitted application application_15
55866246275_0012
19/04/21 23:08:15 INFO mapreduce.Job: The url to track the job: http://quickstar
t.cloudera:8088/proxy/application_1555866246275_0012/
19/04/21 23:08:15 INFO mapreduce.Job: Running job: job_1555866246275_0012
19/04/21 23:08:22 INFO mapreduce.Job: Job job_1555866246275_0012 running in uber
 mode : false
19/04/21 23:08:22 INFO mapreduce.Job:  map 0% reduce 0%
19/04/21 23:08:32 INFO mapreduce.Job:  map 5% reduce 0%
19/04/21 23:08:36 INFO mapreduce.Job:  map 10% reduce 0%
19/04/21 23:08:39 INFO mapreduce.Job:  map 15% reduce 0%
19/04/21 23:08:41 INFO mapreduce.Job:  map 20% reduce 0%
19/04/21 23:08:43 INFO mapreduce.Job:  map 30% reduce 0%
19/04/21 23:08:46 INFO mapreduce.Job:  map 35% reduce 0%
19/04/21 23:08:48 INFO mapreduce.Job:  map 40% reduce 0%
19/04/21 23:08:54 INFO mapreduce.Job:  map 45% reduce 0%
19/04/21 23:08:55 INFO mapreduce.Job:  map 50% reduce 0%
19/04/21 23:08:57 INFO mapreduce.Job:  map 55% reduce 0%
19/04/21 23:08:59 INFO mapreduce.Job:  map 60% reduce 0%
19/04/21 23:09:01 INFO mapreduce.Job:  map 65% reduce 20%
```

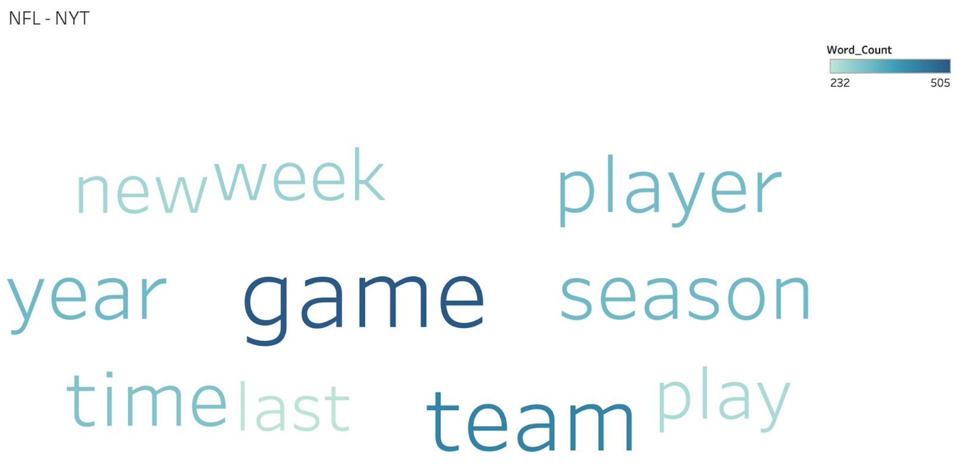

vijayjag — @quickstart:/src — docker exec -it f1344933b60a bash — 80×24

```
19/04/21 23:08:32 INFO mapreduce.Job:  map 5% reduce 0%
19/04/21 23:08:36 INFO mapreduce.Job:  map 10% reduce 0%
19/04/21 23:08:39 INFO mapreduce.Job:  map 15% reduce 0%
19/04/21 23:08:41 INFO mapreduce.Job:  map 20% reduce 0%
19/04/21 23:08:43 INFO mapreduce.Job:  map 30% reduce 0%
19/04/21 23:08:46 INFO mapreduce.Job:  map 35% reduce 0%
19/04/21 23:08:48 INFO mapreduce.Job:  map 40% reduce 0%
19/04/21 23:08:54 INFO mapreduce.Job:  map 45% reduce 0%
19/04/21 23:08:55 INFO mapreduce.Job:  map 50% reduce 0%
19/04/21 23:08:57 INFO mapreduce.Job:  map 55% reduce 0%
19/04/21 23:08:59 INFO mapreduce.Job:  map 60% reduce 0%
19/04/21 23:09:01 INFO mapreduce.Job:  map 65% reduce 20%
19/04/21 23:09:04 INFO mapreduce.Job:  map 65% reduce 22%
19/04/21 23:09:06 INFO mapreduce.Job:  map 70% reduce 22%
19/04/21 23:09:08 INFO mapreduce.Job:  map 75% reduce 22%
19/04/21 23:09:09 INFO mapreduce.Job:  map 80% reduce 22%
19/04/21 23:09:10 INFO mapreduce.Job:  map 80% reduce 25%
19/04/21 23:09:12 INFO mapreduce.Job:  map 85% reduce 25%
19/04/21 23:09:13 INFO mapreduce.Job:  map 85% reduce 27%
19/04/21 23:09:14 INFO mapreduce.Job:  map 90% reduce 27%
19/04/21 23:09:17 INFO mapreduce.Job:  map 90% reduce 30%
19/04/21 23:09:18 INFO mapreduce.Job:  map 95% reduce 30%
19/04/21 23:09:19 INFO mapreduce.Job:  map 100% reduce 30%
```


vijayjag — @quickstart:/src — docker exec -it f1344933b60a bash — 80×24

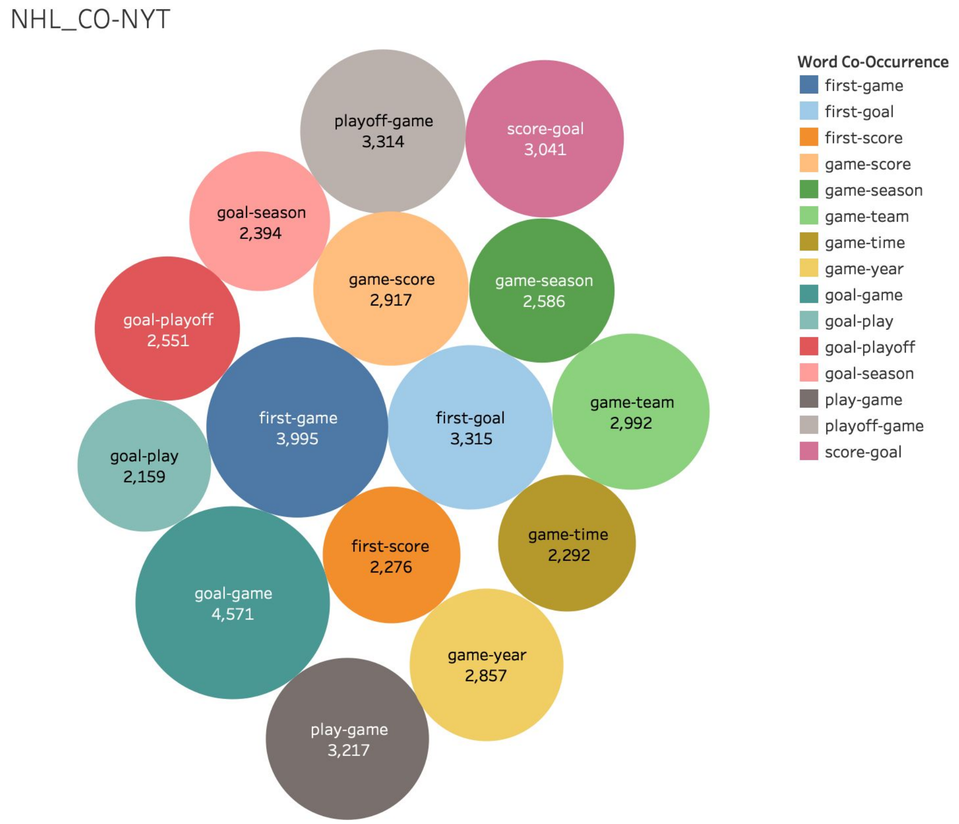
```
                Reduce output records=1914
                Spilled Records=13140
                Shuffled Maps =20
                Failed Shuffles=0
                Merged Map outputs=20
                GC time elapsed (ms)=2231
                CPU time spent (ms)=11380
                Physical memory (bytes) snapshot=5348732928
                Virtual memory (bytes) snapshot=28598583296
                Total committed heap usage (bytes)=5302124544
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=42252
        File Output Format Counters
                Bytes Written=17436
19/04/21 23:09:20 INFO streaming.StreamJob: Output directory: /user/vijayjag/MR/
trial
[root@quickstart src]#
```

Visualized Output for Word-Count:



F1. Color shows sum of F2. Size shows sum of F2. The view is filtered on F1, which keeps 10 of 10,098 members.

Visualized Output for Word Co-occurrence:



References:

[1] https://www.bellingcat.com/resources/2015/08/13/using-python-to-mine-common-crawl/
[2] http://commoncrawl.org/the-data/get-started/
[3] https://developer.nytimes.com/
[4] https://developer.twitter.com/en/docs/tweets/search/overview