Jeremy Beard – MSDS 610 – 20220515 – Week 2 Assignment

- What is a Hadoop distribution? What are some distributions of Hadoop? hint

A Hadoop distribution is a packaged installation of Hadoop that has been tested and contains components which work well together. Usually distributions are supported by individuals, groups, or corporations which can provide help and updates. Distributions also provide an easy way to install packages compared to just loading TAR balls (Kerzner & Maniyam, 2022).

Some distributions of Hadoop are Apache, Cloudera, HortonWorks, MapR, Intel, and Pivotal HD. Apache and HortonWorks are completely free while Intel and Pivotal HD are premium services. Cloudera, MapR, have options for either a free or a premium version.

- How does a Hadoop cluster work?

A Hadoop cluster is a hardware resource that is utilized by Hadoop to process data. Clusters can be deployed and managed by the Ambari microservice. Ambari also provides the ability to create alerts and enable metrics.

Clusters give us the ability to process data in parallel, enabling large quantities of data to be processed at one time. It enables easy scaling and availability (Kerzner & Maniyam, 2022).

- Why does Hadoop create mutiple output files? How does this relate to the reducer step and number of compute nodes? Map Reduce Output

Hadoop creates multiple files to distinguish output files from each cluster (compute node) and also to denote some result in the file structure. The _SUCCESS file is the latter case. The _SUCCESS file is created so that the system can quickly detect if a result has occurred, just from the existence of this file. This makes the checking quick and easy.

The 2nd type of output file is related to the number of clusters being used. These are the true output files containing information and they are named part-[A}-[B] where [A] is either 'm' or 'r' depending on the configuration of MapReduce (map, or reduce), and [B] is a 5-digit mapper/reducer task number, starting from 0 and increasing. So, a job that has 20 mappers would be named part-m-00000, then part-m-00001, and so on until finally, part-m-00019 (StackOverflow, 2012)

- Why should we use version control to store code for projects?

Version control should most definitely be used to store code for projects because it provides an easy way to track changes as the code grows and develops throughout its lifecycle. If a change turns out to be problematic, the code can easily be restored to the state it was at before the change was implemented. This makes it very useful for software engineers and also provides a running log of the work currently being done on the code.

- What are pros/cons of using a cloud provider for a Hadoop cluster vs using our own in-house machines?

There are specific situations where using Hadoop in the cloud or Hadoop on-prem (on the premises) makes more sense. No option is truly greater than the other overall. For example, using Hadoop on-prem means that there needs to be large amounts of servers and racks that is expensive, requires maintenance, and draws a lot of power. On the other hand, it also provides the user more control which can lend itself to a lower security risk. Having all the infrastructure locally also means that the user is closer to the infrastructure which likely increases the performance of Hadoop. Another added benefit of using Hadoop in the cloud is that in addition to the low cost and low maintenance, it can likely be accessed in more places than if it were hosted locally. This makes Hadoop more flexible and the employee more productive. All these aspects must be considered when choosing between using Hadoop on-prem and Hadoop in the cloud (Aggarwal, 2021).

- What do our top word counts look like now, and what do they tell us?

The word counts between the two jobs changed due to the stopwords filtering. You can see the results below:

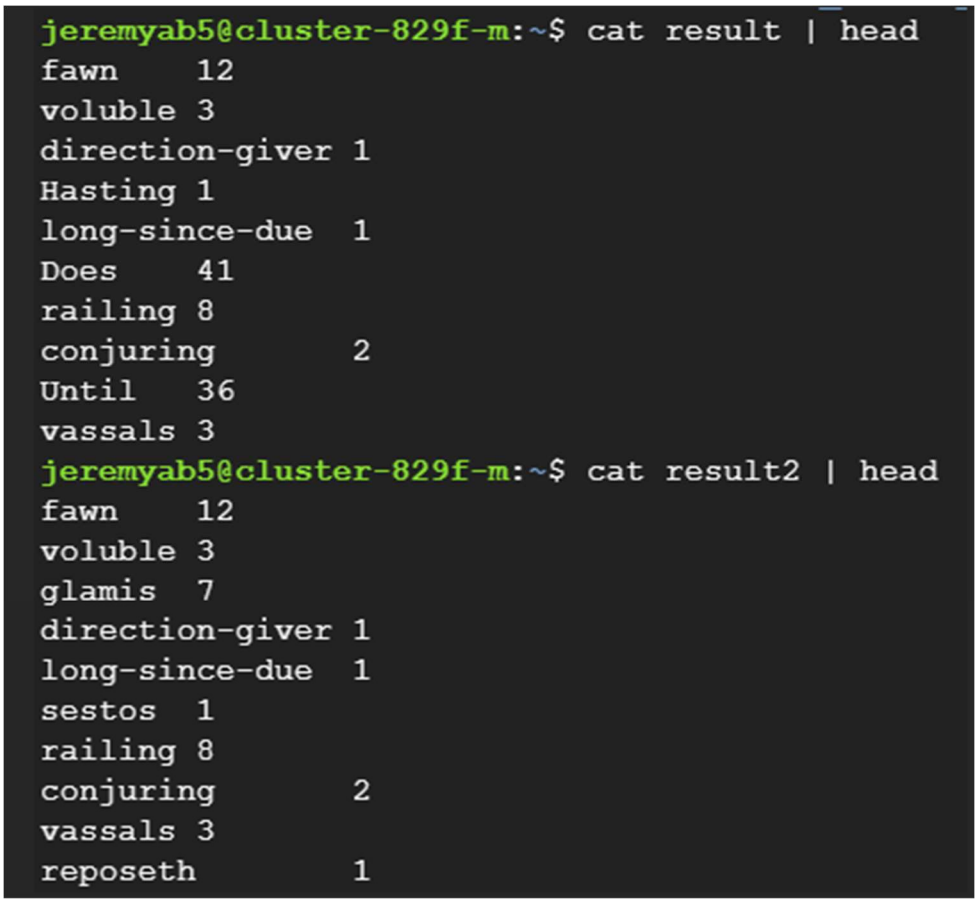

Figure A: Results comparison

Fawn, voluble, direction-giver, long-since-due, railing ,conjuring, and vassals remained in the first 10 lines in both jobs. However, that means there were three differences between the two jobs. This tells us that making all the words lowercase and filtering out the stopwords seems to have had a slight effect on the results.

Technical Section

For the lab this week, I actually already had a github account. However, I did have to fork the git repository from last week on my existing github account.

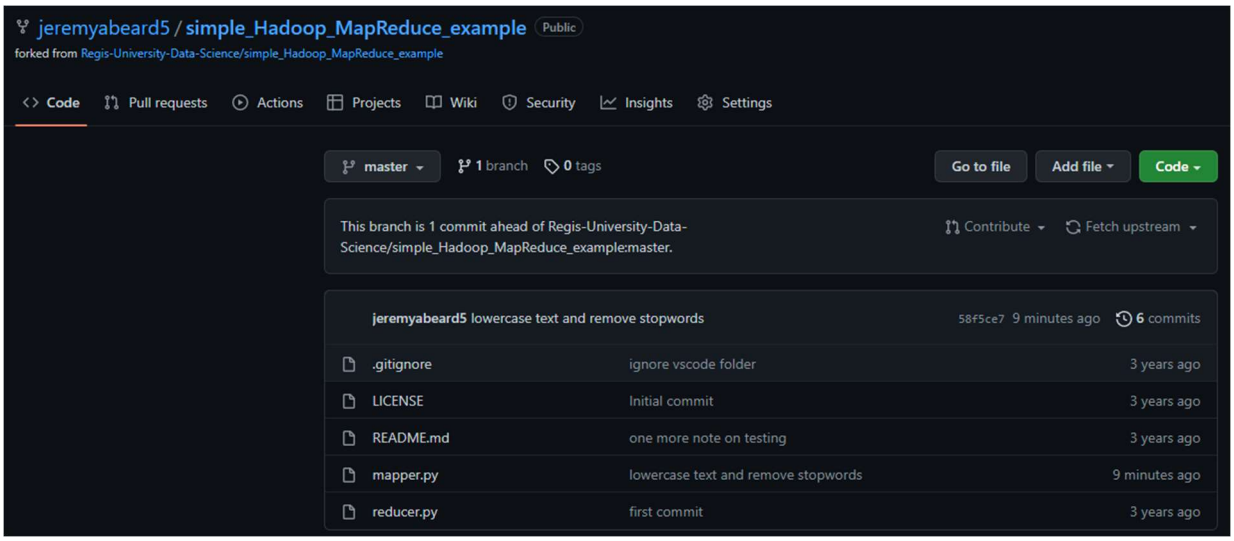

Figure 1: Forked GitHub repository

Next, I set up a cluster on Google Cloud. I chose 1 master node and 2 worker nodes. Each will have 2 vCPUs, 7.5 GB of memory, and 50 GB of disk size. I then verified that my configuration settings were exactly as the lab assignment dictated, and finally clicked “Create” to create the cluster.

After creating the clusters, I SSH’d into the cluster and cloned the github repository that I previously forked.

```
git clone https://github.com/jeremyabeard5/simple_Hadoop_MapReduce_example.git
```

Figure 2: cloning the forked repository

I repeated the commands of the assignment last week, creating directories in Hadoop and copying the Shakespeare.txt file into Hadoop.

```
 7  wget http://norvig.com/ngrams/shakespeare.txt
 8  ls
 9  docker
10  hadoop fs -mkdir /shakespeare
11  hadoop fs -mkdir /shakespeare/input
12  ls
13  hadoop fs -copyFromLocal shakespeare.txt /shakespeare/input
14  hadoop fs -ls /shakespeare/input
```

Figure 3: creating directories in Hadoop, and copying the downloaded Shakespeare.txt file

I then ran the Hadoop-streaming.jar executable to run the Hadoop job. This output was written to the /Shakespeare/directory defined previously. I then merged the output into a single ‘result’ file which was placed on the VM, outside of Hadoop. I viewed the results using the ‘cat’ command, piping the output into ‘head’ to only show the first 10 items.

```
   26  hadoop jar /usr/lib/hadoop/hadoop-streaming.jar -files mapper.py,reducer.py -mappe
r mapper.py -reducer reducer.py -input /shakespeare/input -output /shakespeare/output
```

Figure 4: running Hadoop-streaming.jar and specifying an output directory

```
27  hadoop fs -ls /shakespeare/output
28  hadoop fs -getmerge /shakespeare/output/ /home/jeremyab5/result
```

Figure 5: merging the Hadoop output into a single 'result' local file

```
jeremyab5@cluster-829f-m:~$ cat result | head
fawn    12
voluble 3
direction-giver 1
Hasting 1
long-since-due  1
Does    41
railing 8
conjuring       2
Until   36
vassals 3
```

Figure 6: Showing the results of the wordcount job

After creating a wordcount with stopwords included, I then modified the mapper.py file to filter through stopwords and exclude them from the wordcount. I then re-ran the job and merged the output into the local file 'result2'. I then performed the same 'cat' command to compare the two jobs and see how excluding stopwords can change the result.

```
#!/usr/bin/env python
import sys

# get all lines from stdin
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # make lowercase
    line = line.lower()

    # split the line into words; splits on any whitespace
    words = line.split()

    stopwords = set(['the', 'and'])

    # output tuples (word, 1) in tab-delimited format
    for word in words:
        if word not in stopwords:
            print '%s\t%s' % (word, "1")
~
```

Figure 7: Modifying the mapper.py file to filter stopwords

```
   44  hadoop jar /usr/lib/hadoop/hadoop-streaming.jar -files mapper.py,reducer.py -mappe
r mapper.py -reducer reducer.py -input /shakespeare/input -output /shakespeare/output2
   45  hadoop fs -ls /shakespeare/output2
   46  hadoop fs -getmerge /shakespeare/output2/ /home/jeremyab5/result2
```

Figure 8: Running the same Hadoop-streaming.jar file with the new mapper.py file as input

```
jeremyab5@cluster-829f-m:~$ cat result | head
fawn    12
voluble 3
direction-giver 1
Hasting 1
long-since-due  1
Does    41
railing 8
conjuring       2
Until   36
vassals 3
jeremyab5@cluster-829f-m:~$ cat result2 | head
fawn    12
voluble 3
glamis  7
direction-giver 1
long-since-due  1
sestos  1
railing 8
conjuring       2
vassals 3
reposeth        1
```

Figure 9: Comparing the output of the 2 jobs

After finishing the two jobs, I then configured my git account on the cluster, committed the changes, and pushed the changes to the master branch. I used my personal access token to login to git on the cluster.

```
55  git config --global user.name "jeremyabeard5"
56  git config --global user.email jeremyabeard5@gmail.com
57  git commit -m 'lowercase text and remove stopwords'
58  git add .
59  git commit -m 'lowercase text and remove stopwords'
60  git push origin master
61  git config --global user.email jeremyab5@gmail.com
62  git config
63  git config --list
64  git status
65  git push origin main
66  git push origin master
```

Figure 10: Configuring Git and pushing the changes to my remote repository

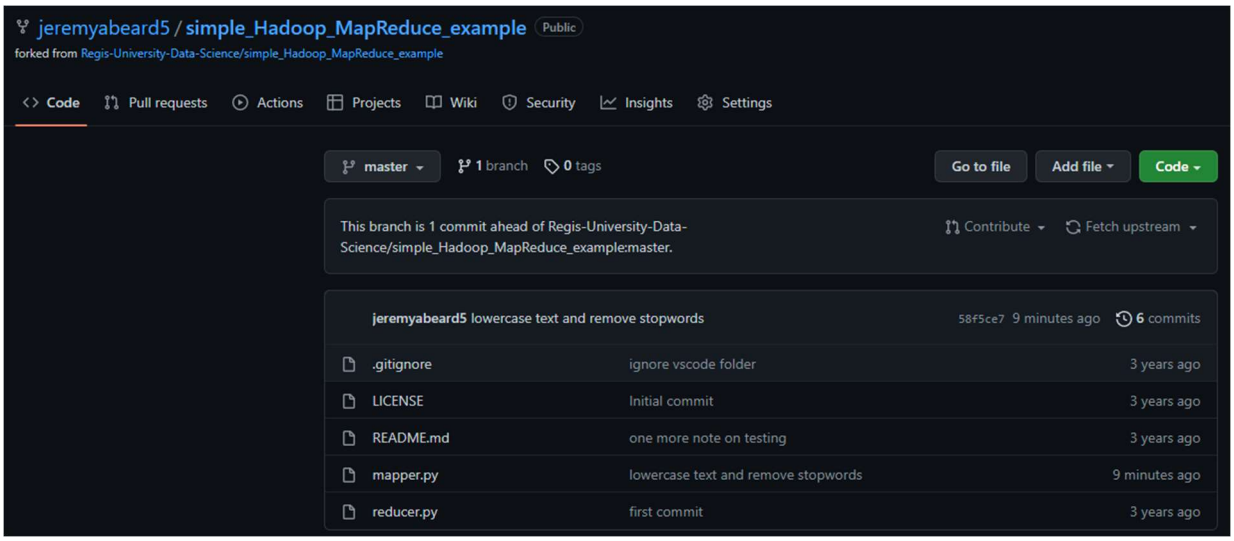

Figure 11: The changes are confirmed in github, mapper.py is shown as updated

This all went very smoothly.
Thank you!
Jeremy

References

1) Kerzner, M., & Maniyam, S. (2022). Hadoop Illuminated :: Hadoop Illuminated. Hadoop Illuminated. Retrieved May 15, 2022, from http://hadoopilluminated.com/hadoop_illuminated/index.html
2) What are SUCCESS and part-r-00000 files in hadoop. (2012, May 19). Stack Overflow. Retrieved May 15, 2022, from https://stackoverflow.com/questions/10666488/what-are-success-and-part-r-00000-files-in-hadoop/10666874#10666874
3) Aggarwal, S. (2021, September 15). Cloud vs. On-Premise Hadoop Providers: A Business Perspective. Qubole. Retrieved May 15, 2022, from https://www.qubole.com/blog/cloud-vs-on-premise-hadoop/#:%7E:text=The%20drawback%20of%20on%2Dpremise,tablets%20through%20an%20Internet%20connection.
4) NLTK :: Installing NLTK Data. (2022). NLTK Project. Retrieved May 15, 2022, from https://www.nltk.org/data.html