Lab 3
COSC760 - Big Data Analytics

Devere Anthony Weaver

---

1 Downloading Files

To download the necessary given Java files, I simply downloaded the files from Blackboard and transferred them to the virtual machine via Git.

Figure 1: Initial state with needed files

2 Load Data to HDFS

The next step is to move the 2008.csv input file into the HDFS on the machine so that Hadoop can read it. To do, I used `hdfs dfs -put <input file> <hdfs directory>` to get the CSV file into an HDFS file I created specifically for this lab.

The following screenshot shows the input CSV file has been placed into an HDFS directory called `input`.

Figure 2: 2008.csv file in HDFS directory

3 Compile Java Files

After getting the data where it needs to be, I complied the Java files into class files using `hadoop com.sun.tools.javac.Main ./*.java -Xlint`.

Figure 3: Compiling Java Files

One thing to note is that for any version of Java I compiled with, there were several warnings being thrown; however, the class files were created.

4   Create JAR

The next step in the lab was to create the JAR file using `jar cf flight.jar *.class`

Figure 4: Creating JAR file

5   Hadoop JAR

This is the first section where I ran into trouble and spent several days trying to find a solution to. When attempting to run `hadoop jar flight.jar FlightByCarrier <input location> <output location>`, I was constantly receiving errors. After making sure I was pointing the command to the correct input location, I was finally able to execute the MapReduce program. However, again I was met with a final error after the program ran.

```
[devere@devere-research:~/COSC760-Big-Data-Analytics/Lab3$ ls              ]
CSVParser.class          FlightsByCarrierMapper.class    flight.jar
CSVParser.java           FlightsByCarrierMapper.java     input
FlightsByCarrier.class   FlightsByCarrierReducer.class   screenshots
FlightsByCarrier.java    FlightsByCarrierReducer.java
devere@devere-research:~/COSC760-Big-Data-Analytics/Lab3$  hadoop jar flight.jar Flig
htsByCarrier hdfs://localhost:9000/input/2008.csv ./output
Caused by: java.lang.ClassNotFoundException: org.apache.hadoop.yarn.exceptions.YarnRu
ntimeException
        at java.net.URLClassLoader.findClass(URLClassLoader.java:387)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
        ... 7 more


[2024-10-10 19:45:38.653]Container exited with a non-zero exit code 1. Error file: pr
elaunch.err.
Last 4096 bytes of prelaunch.err :
Last 4096 bytes of stderr :
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/hadoop/yarn/exc
eptions/YarnRuntimeException
        at java.lang.Class.getDeclaredMethods0(Native Method)
        at java.lang.Class.privateGetDeclaredMethods(Class.java:2701)
        at java.lang.Class.privateGetMethodRecursive(Class.java:3048)
        at java.lang.Class.getMethod0(Class.java:3018)
        at java.lang.Class.getMethod(Class.java:1784)
        at sun.launcher.LauncherHelper.validateMainClass(LauncherHelper.java:670)
        at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:652)
Caused by: java.lang.ClassNotFoundException: org.apache.hadoop.yarn.exceptions.YarnRu
ntimeException
        at java.net.URLClassLoader.findClass(URLClassLoader.java:387)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
        ... 7 more


For more detailed output, check the application tracking page: http://devere-research
:8088/cluster/app/application_1728595897863_0028 Then click on links to logs of each
attempt.
. Failing the application.
2024-10-10 19:45:38,921 INFO mapreduce.Job: Counters: 0
```

As you can see from the images, the error is a `java.lang.NoClassDefFoundError`, `java.lang.ClassNotFoundException`, and a JNI error. These errors were recurring and after attempting to find a solution for a few days, the issue *may* have something to do with mismatched Java compiler versions, although I was still unable to solve the issue after fixing the versions.

Apparently, this is a known issue that occurs from some people and I attempted each of the fixes I found online:

- Error: A JNI error has occurred, please check your installation and try again in IntelliJ IDEA for Scala-Spark Program using SBT

- I am getting an error trying to use Hadoop and YARN. Assistance Recquired,

- Application fails to start with java.lang.ClassNotFoundException 101

- NoClassDefFoundError for simple Yarn application

Above were simply a few pages I consulted to figure out what was going on with this error, ultimately I was unable to fix it even after reinstalling Hadoop and applying the same configurations.

6 Moving Forward

I was unable to successfully implement this MapReduce program for some reason, despite being able to successfully implement the programs from the previous labs. Moving forward, I would like to complete this lab and have the MapReduce program working, so my plan of action is to perform a clean setup of a new virtual machine and dump the old one. Then perform the same reinstall of Hadoop along with the same configurations as the first lab with the goal of being able to avoid this bug.