1. Used putty to connect to the cluster:
2. Created a filw named as “WordCount.java”
3. Set the environment variable :
   1. export HADOOP\_HOME=/home/me/hadoop-3.1.1
   2. export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64
4. Compile it in local
   1. javac -classpath `hadoop classpath` WordCount.java
   2. jar cf wordcount.jar WordCount\*.class
5. To run Map Reduce:

1- Are there any parts of the original WordCount that still confuse you? If so, what?

I understood the relation between the Mapper, Combiner, Shuffle, and Reducer. As well as how these classes must define and connect to each other in yarn.

But there are still a couple of subjects that I need to spend more time on, such as, configuration object, and how yarn internally call these codes and so you might be able to improve the internal behaviour and efficiency of yarn 😊.

2- How did the output change when you submitted with -D mapreduce.job.reduces=3? Why would this be necessary if your job produced large output sets?

In the output folder, there were three files, named part-r-00000, part-r-00001, and part-r-00002. Having a large output of a job means we have a large amount of data to process (input dataset). In theory, it is better to use more than one reducer to process the input data set parallelly, which cause to decrease the reducing phase’s execution time. But, as far as there is an overhead for managing these reducers, the input and output must be large enough to use more than one reducer.

I executed the given code on (wordcount.java) on the larger input dataset (wordcount-2) 3 times.

1) 1 reducer: Total megabyte-milliseconds taken by all reduce tasks=3845120

2) 3 reducers: Total megabyte-milliseconds taken by all reduce tasks=10767360

3) 10 reducers: Total megabyte-milliseconds taken by all reduce tasks=35613696

3- How was the -D mapreduce.job.reduces=0 output different?

In this execution, as it is specified in the command there executed no reducer. As a result, we could see the mapper’s output files in the output directory.

The output directory includes part-m-00000, part-m-00001, part-m-00002, part-m-00003. these files are the output of the mapper execution (One file for each Mapper).

4- Was there any noticeable difference in the running time of your RedditAverage with and without the combiner optimization?

I just ran with and without combiner code optimization, on reddit-1 reddit-3 (which was the largest dataset) but the reducer time was 3 times faster. In other words, the combiner makes execution of the reducer 3 times faster.

export LIBJARS=./json-20180813.jar

yarn jar redditaverage.jar RedditAverage -libjars ${LIBJARS} /courses/732/reddit-1 output-Red-13