KMeans over MapReduce

1. Start by logging on to your master node, and transferring "KMeans.zip" over to the node by typing the following at the command prompt:

```
wget http://cmj4.web.rice.edu/KMeans.zip
```

2. Then unzip "KMeans.zip":

```
sudo apt-get install unzip
unzip KMeans.zip
```

3. Next, follow the instructions to prepare the data, just like you did for the sequential KMeans activity earlier. The one difference is that rather than just using 100 files for each subdirectory, we will use all 1000 of them (after all, we have three machines with two cores each at our disposal, so we will be able to handle more data). More specifically, perform steps 4 and 5 from KMeans.html, except that now when you are prompted with:

```
Enter the max number of docs to use in each subdirectory:
```

You will enter in "1000" rather than "100".

4. Now we'll copy the "vectors" and "clusters" files into HDFS:

```
hdfs dfs -mkdir /data
hdfs dfs -mkdir /clusters
hdfs dfs -copyFromLocal vectors /data
hdfs dfs -copyFromLocal clusters /clusters
```

5. Now we are ready to run KMeans over Hadoop. Start by downloading "MapRedKMeans.zip" from here to your local machine (not your cluster!). Unzip "MapRedKMeans.zip" on your local machine, which will create a directory called "MapRedKMeans". There will be a bunch of ".java" files in there, as well as two ".jar" files. Transfer "MapRedKMeans.jar" over to the master.

6. SSH into the master and run KMeans on Hadoop by typing:

```
hadoop jar MapRedKMeans.jar KMeans /data /clusters 3
```

This will run 3 iterations of the KMeans algorithm on top of all 20,000 documents in the 20_newsgroups data set. "/data" is the directory in HDFS where the data are located, "/clusters" is the directory where the initial clusters are located, and "3" is the number of iterations to run; this means that three separate MapReduce jobs will be run in sequence.

The centroids produced at the end of iteration 1 will be put into the HDFS directory "/clusters1", those from the end of iteration 2 in "/clusters2", and those from the end of iteration 3 in "/clusters3".

7. Now it is your turn to implement your own version of KMeans over MapReduce. Back on your local machine, in the "MapRedKMeans" directory, using whatever development environment you prefer, create a project that includes all of the ".java" files plus the "Hadoop.jar" file.

Start out by taking a look at "KMeans.java". This file contains the code that will set up the MapReduce jobs that run each iteration.

Next, take a look at "KMeansMapper.java" and "KMeansReducer.java". To complete this activity, you are going to write some code that goes into these two files. Most of your code will be copied and pasted from the sequential (non-distributed) KMeans code from the last activity, but you'll have to write a few lines of new code to get this to work. If you get totally stuck and nothing other than seeing a working code is going to help, you can check out "KMeansMapper.java" and "KMeansReducer.java".

By the way, it is possible to check the quality of the clustering result in the same way that you checked the clustering quality on a single machine. When the program completes, type:

```
ubuntu@ip-10-182-102-114:~$ hdfs dfs -copyToLocal /clustersNewXXX/part-r-00000 .
```

This will copy the result from HDFS to the local machine (replace XXX with the last iteration number that your clustering algorithm ran). Then:

```
ubuntu@ip-10-182-102-114:~$ java -jar GetDistribution.jar
Enter the file with the data vectors: vectors
Enter the name of the file where the clusers are loated: part-r-00000
```