CISC 5950 — Lab 2

Congratulations on successfully completing the project 1. At the current stage, you should feel comfortable to read and write MapReduce based programs.

In the project 1, you developed a MapReduce based K-Means classifier. You should have observed that developing a interative program, which involves multiple Maps and Reduces, with MapReduce programming framework is definitely not a trivial task. The have to use a loop in the shell program to start the interation and utilize an indicator to stop the loop (Fig. 1).

```
3  echo 'starting hdfs, running map-reduce'
4  ../../start.sh
5
6  counter=1
7  check="Start"
8
9  while [ "$check" != "DONE" ]; do
10     echo "iteration $counter"
11
12     /usr/local/hadoop/bin/hdfs dfs -rm -r /Q2P2/output/
13     /usr/local/hadoop/bin/hdfs dfs -rm -r /Q2P2/input/
14     /usr/local/hadoop/bin/hdfs dfs -mkdir -p /Q2P2/input/
15
16     if [ "$counter" == 1 ]; then
17         /usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../data/shot_logs.csv /Q2P2/input/
18     else
19        /usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../data/cz_output.csv /Q2P2/input/
20     fi
21
22     counter=$((counter+1))
23
24     /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.9.2.jar '
25     -file ../../mapreduce-test-python/Q2P2/mapper.py  -mapper 'python mapper.py' \
26     -file ../../mapreduce-test-python/Q2P2/reducer.py -reducer 'python reducer.py' \
27     -input /Q2P2/input/* -output /Q2P2/output/
28
29     /usr/local/hadoop/bin/hdfs dfs -cat /Q2P2/output/part-00000 > ../data/cz_output.csv
30
31     check=$(head -n 1 ../data/cz_output.csv)
32 done
33
34 echo 'Done!'
35 /usr/local/hadoop/bin/hdfs dfs -cat /Q2P2/output/part-00000
36 /usr/local/hadoop/bin/hdfs dfs -rm -r /Q2P2/output/
37 /usr/local/hadoop/bin/hdfs dfs -rm -r /Q2P2/input/
38 ../../stop.sh
39
```

Figure 1: MapReduce based interative programming

This challenge is caused by the fact that Hadoop is design to utilize the storage space in the cluster. However, each MapReduce program requires to output the data into the hard drive. The feature leads to a large amount of read/write of HDFS, which significantly limits the performance.

Spark Programming

The spark system implements the Resilient Distributed Dataset (RDD) to maximize the memory space in the cluster. With RDD, most of the operation is done in the memory (Fig. 2).

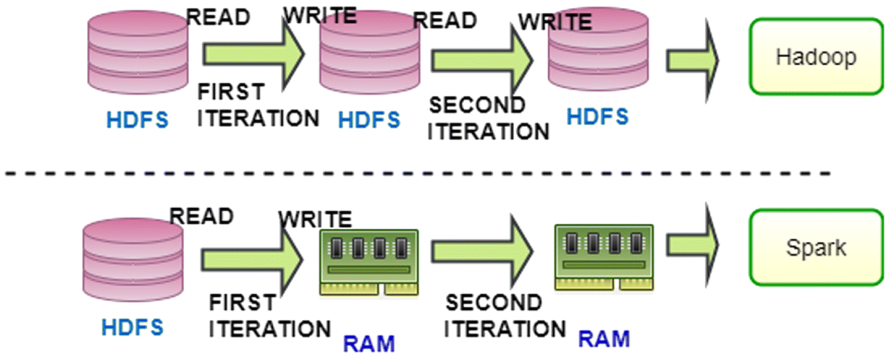

Figure 2: Hadoop v.s. Spark

To develop a K-Means algorithm in spark, you just need to transform the previous RDD into a new one for the next interation.

Programming in Lab 2

In this lab, please, based on your previous code, inplement the K-Means algorithm without using **spark ml** package.

1. Please redo Project 1 Part 2 Question 2 (Part 1).
2. Please redo Project 1 Bonus Question 1(Part 2).
3. Please try different levels of parallelism, 2, 3, 4, 5 (Part 3).

Installing the spark cluster GitHub Link.

Grading Rubric

You should complete the lab in groups of 2 students.

(90%) Lab 1;
(10%) Report;
(5% * 2) Part 2 and 3.

Submission

You should demostrated your program of part 1 in class on Tuesday, April 9 and submit your part 1 / part2 / part 3 package by the end of April 9 through email.

Submission Format: In a zip/tar file, you should have 3 folders for each of the part and one report.