ENGR 516 — Assignment 2

In ENGR 5950, we have learned the following topics:

1. Set up a 3-node cluster with Hadoop Distributed File System and run examples.
2. On top of HDFS, set up the cluster with MapReduce programming framework.
3. Run examples of MapReduce programs.
4. Scheduling with YARN.

However, you should have observed that developing an integrative program, which involves multiple Maps and Reduces, with MapReduce programming framework is definitely not a trivial task. You have to use a loop in the shell program to start the iteration and utilize an indicator to stop the loop (Figure 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 iterative programming

This challenge is caused by the fact that Hadoop is designed to utilize the storage space in the cluster. Each MapReduce program requires to output the data into the disk. This leads to a large amount of HDFS reads/writes, 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).

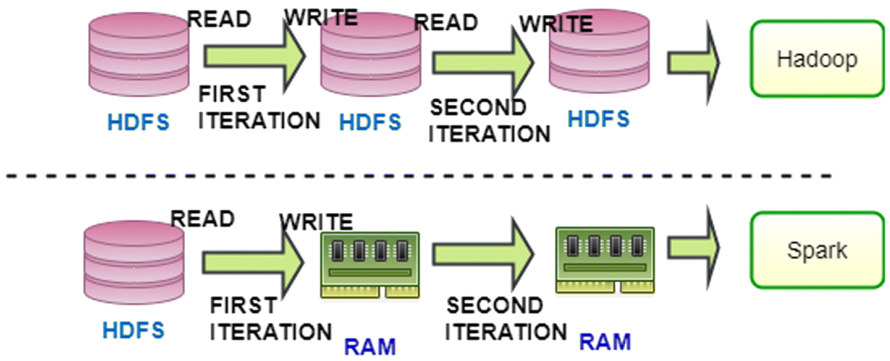

Figure 2: Hadoop v.s. Spark

In this project, you are going to design you own Spark programs to analyze the data. To develop a Spark program, you just need to transform the previous RDD into a new one for the next iteration. You can use any spark related library package in this project.

NY Parking Violations

The NYC Department of Finance collects data on every parking ticket issued in NYC. This data is made publicly available to aid in ticket resolution and to guide policymakers.

You can find the data from the Link of NYC Parking Data.

	# Summ...	A Plate ID	A Registr...	A Plate Ty...	Issue D...	# Violatio...	A Vehicle ...	A Vehicle ...	A Issuing ...	# Street ...
1	1283294138	GBB9093	NY	PAS	08/04/2013	46	SUBN	AUDI	P	37250
2	1283294151	62416MB	NY	COM	08/04/2013	46	VAN	FORD	P	37290
3	1283294163	78755JZ	NY	COM	08/05/2013	46	P-U	CHEVR	P	37030
4	1283294175	63009MA	NY	COM	08/05/2013	46	VAN	FORD	P	37270
5	1283294187	91648MC	NY	COM	08/08/2013	41	TRLR	GMC	P	37240
6	1283294217	T60DAR	NJ	PAS	08/11/2013	14	P-U	DODGE	P	37250
7	1283294229	GCR2838	NY	PAS	08/11/2013	14	VAN		P	37250
8	1283983620	XZ764G	NJ	PAS	08/07/2013	24	DELV	FORD	X	63430
9	1283983631	GBH9379	NY	PAS	08/07/2013	24	SDN	TOYOT	X	63430
10	1283983667	MCL78B	NJ	PAS	07/18/2013	24	SDN	SUBAR	H	0
11	1283983679	M367CN	NY	PAS	07/18/2013	24	SDN	HYUND	H	0
12	1283983734	GAR6813	NY	PAS	07/18/2013	24	SDN	TOYOT	H	0

The above figure shows several records, where each row represents a parking ticket and the columns are the details of the tickets.

First, please follow this instruction (GitHub Link) to install the spark cluster.

Then, by analyzing the data, you need to answer the following questions:

- When are tickets most likely to be issued? (15 pts)
- What are the most common years and types of cars to be ticketed? (15 pts)
- Where are tickets most commonly issued? (15 pts)
- Which color of the vehicle is most likely to get a ticket? (15 pts)

Based on a K-Means algorithm, please try to answer the following question:

- Given a Black vehicle parking illegally at 34510, 10030, 34050 (street codes). What is the probability that it will get an ticket? (very rough prediction). (20 pts)

Note that the biggest challenge when using K-Means is to decide on the number of clusters. Having more clusters creates some small classes with very few records, while having less clusters leads to classes that are too general.

NBA Shot Logs

This is the DATA on shots taken during the 2014-2015 season, who took the shot, where on the floor was the shot taken from, who was the nearest defender, how far away was the nearest defender, time on the shot clock, etc. The column titles are generally self-explanatory.

The below figure shows several records, where each row represents a shot and the columns are the details of the shot, e.g., game ID, who's defender, what's the distance between them.

	# GAME_ID	A MATCH...	A LOCATI...	A W	# FINAL_...	# SHOT_...	# PERIOD	GAME_...	# SHOT_...	# DRIBB...
1	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	1	1	1:09	10.8	2
2	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	2	1	0:14	3.4	0
3	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	3	1	0:00		3
4	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	4	2	11:47	10.3	2
5	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	5	2	10:34	10.9	2
6	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	6	2	8:15	9.1	2
7	21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	7	4	10:15	14.5	11

Please analyze the data and answer the following questions:

- For each pair of the players (A, B), we define the **fear sore** of A when facing B is the hit rate, such that B is closet defender when A is shooting. Based on the **fear sore**, for each player, please find out who is his "most unwanted defender". (10 pts)
- For each player, we define the **comfortable zone** of shooting is a matrix of,

  $$\{\text{SHOT\_DIST, CLOSE\_DEF\_DIST, SHOT\_CLOCK}\}$$

  Please develop a Spark-based algorithm to classify each player's records into 4 comfortable zones. Considering the hit rate, which zone is the best for James Harden, Chris Paul, Stephen Curry, and Lebron James. (10 pts)

Submission

You are expected to upload a zip or tar file by the deadline to Canvas. The zip file should include your codes, report, and README.

Useful Links

1. Analysis of NYC Parking Tickets.
2. Preliminary Data Visualization.
3. Exploring 42.3M NYC Parking Tickets.
4. NY Parking Violations Issued.
5. Insights From Raw NBA Shot Log Data.
6. Investigating the hot hand phenomenon in the NBA (CODE).
7. Parallelizing K-Means-Based Clustering on Spark.
8. NBA 16-17 regular season shot log.
9. The Fear Factor.
10. The Best And Worst Defenders.
11. NBA Classification.
12. Stephen Curry's Decision Tree.
13. Points per Match (ATL vs WAS only).
14. Spark Kmeans.