CSC 369 Final Project Report

US Traffic Accidents (2016-2019)

Contributors: Derek Kelley, Kevin Yoo, Joulien Ivanov, Tyler Davis

Q1 – States, Road Types & Severity - Kevin Yoo

The program in Q1.scala calculates the total number of accidents for each state and for each of the top 5 states outputs the road types the accidents occurs on in order of greatest to least number of accidents along with the average severity level for the road type. The “street” column of the data file consist of many different formats of strings to describe the street name on which the accident occurred (i.e. “I-5 W”, “Capital Fwy N”, “US-55 S”, “Abalone St.”, etc…). Based on these descriptions, the road types (i.e. State Highway, US Highway, Interstate, etc…) are extracted using regular expressions.

First, 2 RDDS are created to hold:

1. (state, # of accidents)
2. (state, (road type, severity))

Then, I take RDD2 and group the data by state, then group the data by road type to find both the number of accidents which occurred and the average severity on that road type in order of the number of accidents. RDD1 and RDD2 are joined, sorted by total number of accidents for that state. Finally, I output the top 5 rows.

Q2 – Severity & Description – Derek Kelley

The dataset was populated from two separate APIs, MapQuest and Bing. For each API, the natural language description of the accidents had 2 main formats; for MapQuest, descriptions either started with “Accident on…” or “Lane blocked due to…”, and for Bing, descriptions either ended with “- Accident” or “- Road closed due to accident”. The program in Q2.scala calculates the average severity of accidents based on 1) the accident description format and 2) the API source.

First, 7 RDDs are created to hold:

1. All accidents
2. Accidents with MapQuest as the API and a description of “Accident on…”

3. Accidents with MapQuest as the API and a description of “Lane blocked due to…”
4. All other accidents provided by the MapQuest API
5. Accidents with Bing as the API and a description ending with “- Accident”
6. Accidents with Bing as the API and a description ending with “- Road closed due to accident”
7. All other accidents provided by the Bing API

Each record in every RDD was in the format (ID, Severity).

Then, I averaged the severity for each of the 7 RDDs I created. Results:

1. All accidents: 2.36
2. MapQuest, “Accident on…”: 2.19
3. MapQuest, “Lane blocked…”: 2.59
4. MapQuest, Other: 2.31
5. Bing, “- Accident”: 2.18
6. Bing, “- Road closed due to accident”: 4.0
7. Bing, Other: 2.38

Performance: 15.8 seconds

A few interesting conclusions:

- The “Other” categories for Bing and MapQuest, along with the average severity across the entire dataset, all have similar average severities. Thus, no correlations could be drawn from descriptions that didn’t follow the main description formats, regardless of API source
- For MapQuest, there was a 13% difference between the average severities for the “Accident on…” and “Lane blocked…” formats
- For Bing, there was a 60% difference between the average severities for the “- Accident” and “- Road closed due to accident” formats
- The “- Road closed due to accident” format had an average severity of 4.0, which means that every time this substring was apart of the natural language description of the accident, the severity was at its max (it’s on a 1-4 scale)
- Both APIs show that certain formats of the accident description entail higher severity rates

One set of experimental results was the average severity per description type **without** the API source distinction. As the results show above, the source distinction was

important to include in the analysis because the natural language descriptions have different formats depending on the API.

Q3 - Severity & Time of Year / Day – Joulien Ivanov

Part 1) What days of the year have the most accidents?
Part 2) What hours are most accident prone?
Part 3) What hours have most accidents with severity 4?

Two RDDs were created to hold:

1. All accidents (filtered accidents starting from 2017-2019)
    a. The real dataset starts in Feb 2016 but I needed to discard 2016 since missing Jan would potentially skew results.
2. Accidents where severities equal 4 (which is the max severity assignable)

Part 1)

I calculated the average number of accidents for any given day over the 2-year period given by our sample of accidents. I then map the accident data into pairs of month-day segments and a count of 1.0 like so *("10-27", 1.0)*. This allows me to reduceByKey and sum the counts, which I then mapValues into (totalCount / averageAccidents) rounded to two decimal places. To finish, I sort by descending values and take the top 5 month-day pairs.

Part 2 & 3)

I map the accident data into pairs of hour occurred and a count of 1.0 like so *("13", 1.0)*. I then reduceByKey by summing the counts, sort by the count value, and take the top 5 hour pairs. This process is essentially the exact same for Part 3; however, before mapping the lines, I filter the data to solely contain accidents with level 4 severities.

I originally experimented finding the same datasets with more complicated functions such as groupByKey and aggregate but ultimately decided to simplify my code using some of the first functions we learned about in scala and spark.

Q4 – Tyler Davis

Part 1) Do different reporting sources report accidents differently based on roadway?

First, I mapped the roads for each accident to their roadway types using a function designed by Kevin for Q1. The output of this mapping is in the form:

((<reporting source>,<road type>),1)

Then, I added the values for all accidents with the same key. This will be called rdd1. The result of this step is in the form:

((<reporting source>,<road type>),<# for this source and road type>)

Next, I used the natural key of the reporting source and summed the number of accidents to the total number of accidents (any roadway) reported by this source. This will be called rdd2. The output of this step looks like:

(<reporting source>,<total # of accidents>)

Finally, to find the proportion of the source's accidents that occured on that road type, I joined rdd1 and rdd2 where the sources are the same and divided the values to find the proportion. The output of this step is:

((<reporting source>,<road type>),<proportion of accidents from this source on this road type>)

Part 2) Do different reporting sources report severity differently?

Part 2 follows a similar set of steps to part 1; however, instead of road type, severity (1, 2, 3, or 4) is used instead.

Justification for Hadoop

Hadoop was a necessity when performing these calculations because we had a lot of data in our dataset (750,000 entries at ~300MB). Because our data contains so much information, the distributed capabilities of Hadoop and Spark made the program extremely efficient.

Time to Execute Each Question

(Measured using unix's "time" command: calculated by summing real time and sys time on three separate occasions and averaging)

Q1 – 12.515s

Q2 – 15.8s

Q3 – 18.551s

Q4 – 12.871s