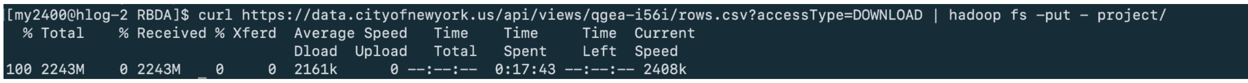
Data Source:

1. The dataset I used in this project is New York City Police Department (NYPD) Historic Complaint Data. This dataset includes all valid felony, misdemeanor, and violation crimes reported to the NYPD from 2006 to the end of 2019. [1]
2. The dataset is 2.243GB, containing 7.38M rows and 35 columns.
3. The following table describes the fields that are useful in this project. The description of all fields can be found at NYPD Complaint Incident Level Data Footnotes [2].

Field Name	**Description**
CMPLNT_NUM (int)	Randomly generated persistent ID for each complaint
KEY_CD (int)	"Key Code": Offense Classification Code (3 digits)
OFNS_DESC	Description of offense corresponding with key code
CMPLNT_FR_DT	Date of occurrence for the reported event
Latitude	Global Latitude of Location where Incident Occurred
Longitude	Global Longitude of Location where Incident Occurred

4. I used curl to download the dataset and redirected the output to HDFS. Downloading the dataset takes around 18 minutes. I renamed the output as project/crime.csv in HDFS. The shell command and output is shown in the following screenshot.

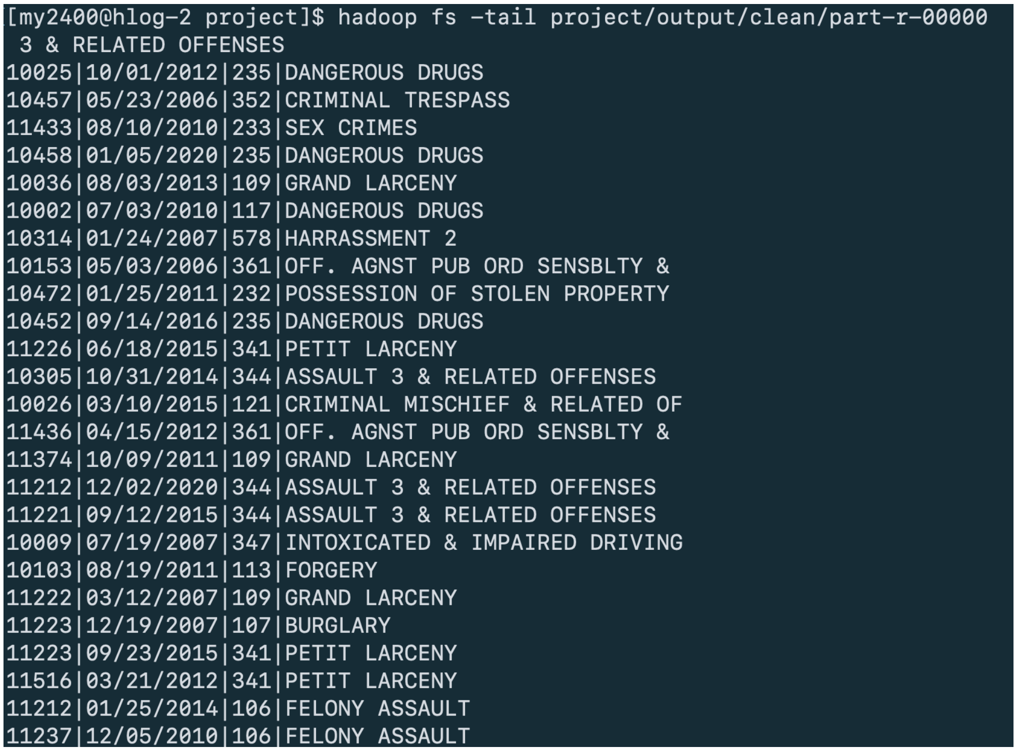


```
[my2400@hlog-2 RBDA]$ curl https://data.cityofnewyork.us/api/views/qgea-i56i/rows.csv?accessType=DOWNLOAD | hadoop fs -put - project/
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2243M    0 2243M    0     0  2161k      0 --:--:--  0:17:43 --:--:-- 2408k
```

5. In the code submission, small-crime.csv is a snippet of the dataset. It contains the first 10 rows of the dataset, which is a great resource for me to test the MapReduce code in a short debug cycle.

Data Cleaning:

1. In the cleaner mapper (GetZipcodeAndCleanMapper.java), I tokenized the input lines by comma (except for the enclosed string) and extracted the complaints_id, report_date, offense_key_code, offense_description, latitude and longitude. All the other columns are ignored.
2. If any of the fields mentioned in step 1 is empty, I treat the row as malformatted and ignore it. If latitude and longitude are not numeric, the row will be ignored as well.
3. I also converted latitude and longitude into zip code for each line in the cleaner mapper.
    a. I maintained an array called geo_zipcode in the mapper class. Each string in the array contains a NYC zip code and its associated latitude and longitude. I got the information from an open database collected from the US census [3].
    b. To convert the geolocation of each incident to its zip code, the mapper searches the geolocation that is closest to the incident's geolocation in the geo_zipcode array and returns the association zip code.
4. Overall, the cleaner mapper (GetZipcodeAndCleanMapper.java) removes unnecessary columns, malformatted rows, and converts latitude and longitude to zip code. The output of the mapper uses complaint_id as key and the combination of zip code, report_date, offense_code and offense_description as value. Each field in the value is separated by "|". The screenshot below is a tail of the output of the cleaner.
5. The cleaner reducer (GetZipcodeAndCleanReducer.java) is almost like an identity reducer. The only difference is that it ignores the complaint_id. The output of the reducer uses NullWritable as key and the value of mapper as the value. Here, I only used one reducer and the output is stored in project/output/clean
6. The screenshot below is a tail of the output of the cleaner:



```
[my2400@hlog-2 project]$ hadoop fs -tail project/output/clean/part-r-00000
 3 & RELATED OFFENSES
10025|10/01/2012|235|DANGEROUS DRUGS
10457|05/23/2006|352|CRIMINAL TRESPASS
11433|08/10/2010|233|SEX CRIMES
10458|01/05/2020|235|DANGEROUS DRUGS
10036|08/03/2013|109|GRAND LARCENY
10002|07/03/2010|117|DANGEROUS DRUGS
10314|01/24/2007|578|HARRASSMENT 2
10153|05/03/2006|361|OFF. AGNST PUB ORD SENSBLTY &
10472|01/25/2011|232|POSSESSION OF STOLEN PROPERTY
10452|09/14/2016|235|DANGEROUS DRUGS
11226|06/18/2015|341|PETIT LARCENY
10305|10/31/2014|344|ASSAULT 3 & RELATED OFFENSES
10026|03/10/2015|121|CRIMINAL MISCHIEF & RELATED OF
11436|04/15/2012|361|OFF. AGNST PUB ORD SENSBLTY &
11374|10/09/2011|109|GRAND LARCENY
11212|12/02/2020|344|ASSAULT 3 & RELATED OFFENSES
11221|09/12/2015|344|ASSAULT 3 & RELATED OFFENSES
10009|07/19/2007|347|INTOXICATED & IMPAIRED DRIVING
10103|08/19/2011|113|FORGERY
11222|03/12/2007|109|GRAND LARCENY
11223|12/19/2007|107|BURGLARY
11223|09/23/2015|341|PETIT LARCENY
11516|03/21/2012|341|PETIT LARCENY
11212|01/25/2014|106|FELONY ASSAULT
11237|12/05/2010|106|FELONY ASSAULT
```

Data Profiling and Some Initial Analysis:

- Get Total Complaints:

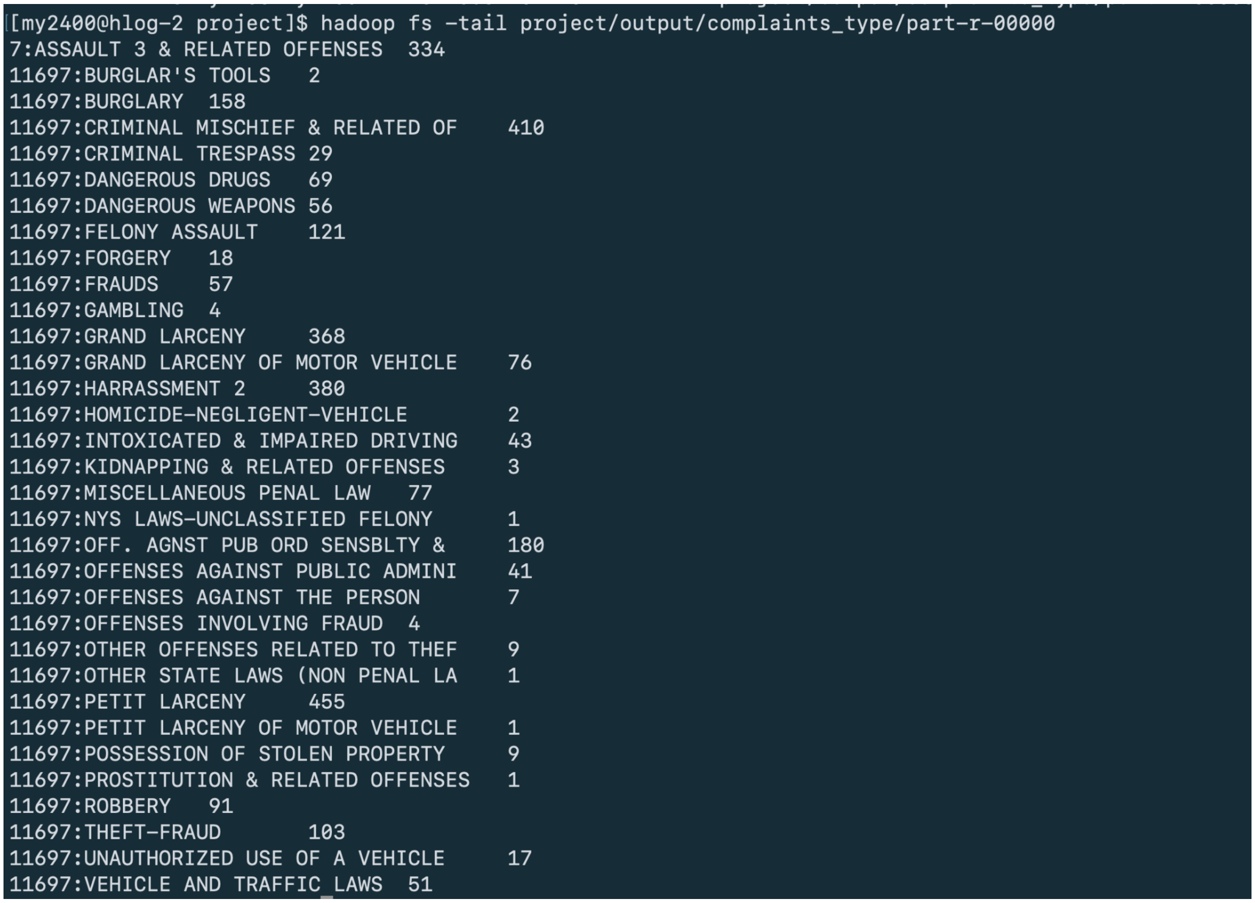
1. The input of this job is crime-data-clean.csv, the clean data with zip code that was obtained from the cleaner task. The output of this job is the total number of complaints for each zip code.
2. Mapper (ZipcodeMapper.java) for this job is relatively simple. It extracts the zip code from the input line and writes it as a key. Other parts of the input line will be written as value. This mapper is very generous and can be used in other jobs as well.
3. Reducer (GetTotalComplaintsReducer.java) for this job maintains a variable called total_complaints. For each information it receives of the zip code, it increments 1 to total_complaints. The output of the reducer has zip code as key, and total_complaints as value.
4. I added some data profiling in this task. I maintained an ArrayList, called statistics, in the reducer. Each time the reduce() function computes a total number of complaints for a zip code, it will add the number to the statistics array list. In the reducer cleanup() function, I computed the min, max, average, and median number of the total number of complaints in the dataset. The results are:

Minimum Number of Complaints for all Zip Codes	6
Max Number of Complaints for all Zip Codes	111038
Average Number of Complaints for all Zip Codes	27594
Median Number of Complaints for all Zip Codes	24837

- Categorize Complaints by Type:

1. The input of this job is the same with the GetTotalComplaints task. The output of this job is the total number of complaints for each offense type in each zip code area.
2. The Mapper of this job is still ZipcodeMapper.java, as described in the last task.
3. The Reducer of this job is GroupComplaintsTypeReducer.java. It maintains a TreeMap in the reduce() function, called statistics, to record the total number of complaints for each complaint type for each zip code area. The output uses the zip code and offense type as key, separated by “:”, and number of complaints as value.

4. Screenshot below is a tail of the task output.



```
[my2400@hlog-2 project]$ hadoop fs -tail project/output/complaints_type/part-r-00000
7:ASSAULT 3 & RELATED OFFENSES  334
11697:BURGLAR'S TOOLS   2
11697:BURGLARY  158
11697:CRIMINAL MISCHIEF & RELATED OF    410
11697:CRIMINAL TRESPASS 29
11697:DANGEROUS DRUGS   69
11697:DANGEROUS WEAPONS 56
11697:FELONY ASSAULT    121
11697:FORGERY   18
11697:FRAUDS    57
11697:GAMBLING  4
11697:GRAND LARCENY     368
11697:GRAND LARCENY OF MOTOR VEHICLE    76
11697:HARRASSMENT 2     380
11697:HOMICIDE-NEGLIGENT-VEHICLE        2
11697:INTOXICATED & IMPAIRED DRIVING    43
11697:KIDNAPPING & RELATED OFFENSES     3
11697:MISCELLANEOUS PENAL LAW   77
11697:NYS LAWS-UNCLASSIFIED FELONY      1
11697:OFF. AGNST PUB ORD SENSBLTY &     180
11697:OFFENSES AGAINST PUBLIC ADMINI    41
11697:OFFENSES AGAINST THE PERSON       7
11697:OFFENSES INVOLVING FRAUD  4
11697:OTHER OFFENSES RELATED TO THEF    9
11697:OTHER STATE LAWS (NON PENAL LA    1
11697:PETIT LARCENY     455
11697:PETIT LARCENY OF MOTOR VEHICLE    1
11697:POSSESSION OF STOLEN PROPERTY     9
11697:PROSTITUTION & RELATED OFFENSES   1
11697:ROBBERY   91
11697:THEFT-FRAUD       103
11697:UNAUTHORIZED USE OF A VEHICLE     17
11697:VEHICLE AND TRAFFIC LAWS  51
```

5. I added some data profiling to the task. The reducer class maintains a TreeMap called overall_statistics, which records the number of total complaints for each complaint type in the dataset. In the cleanup() function, I computed the number of distinct complaint types in the dataset, as well as the complaint type with the max number of complaints and the min number of complaints.

Distinct Number of Complaint Types	71
Complaint Type with the Max Number of Complaint Times	PETIT LARCENY
Complaint Type with the Min Number of Complaint Times	KIDNAPPING AND RELATED OFFENSES

Reference:

1. https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i
2. https://data.cityofnewyork.us/api/views/qgea-i56i/files/b21ec89f-4d7b-494e-b2e9-f69ae7f4c228?download=true&filename=NYPD_Complaint_Incident_Level_Data_Footnotes.pdf
3. https://gist.github.com/erichurst/7882666