Faculty of Engineering & Applied Science

Final Project Report

Traffic flow optimization using highD dataset

Github link: https://github.com/Gutu12/Cloud-Computing-Final

Group : ***W1***

Section CRN: 74293

Name	Student Number
Gutu Shiferaw	100767090
Walid Ayub	100695612
Aayush Parikh	100724927
Muhmamad Hamza Zahid	100711463

Introduction

We wanted to create a traffic management system application that can reduce traffic congestion and enhance safety on highways. Traffic congestion is known to lead to various safety hazards, such as accidents and traffic gridlock, which can cause significant delays of commuters and businesses. Our project's goal was to develop a traffic management system using the HighD dataset that could be used to optimize traffic flow by accurately determining when a highway is congested and identify potential areas of congestion. In order to achieve this we employed a pub/sub system that can accurately detect and analyze traffic congestion.

The system reads data from the HighD dataset, which contains various pieces of information such as capturing visual recordings, and contains information about free-flowing traffic, jammed-up traffic, and traffic accidents. By analyzing this data, the application can determine whether the number of vehicles on a highway exceeds a specific threshold. Which will then indicate the presence of congestion. Once the application identifies areas of congestion, we are hoping this can help law enforcement or traffic planners suggest alternative routes or suggest adjustments to traffic flow. An example of this would be if the traffic application detects that the number of vehicles on the road are above a specified threshold, this information will then get sent to the traffic planner and the traffic planner is then able to reduce traffic on the congested road by altering the traffic signals.

Methodology

Design and Implementation of the Pub/Sub Architecture:

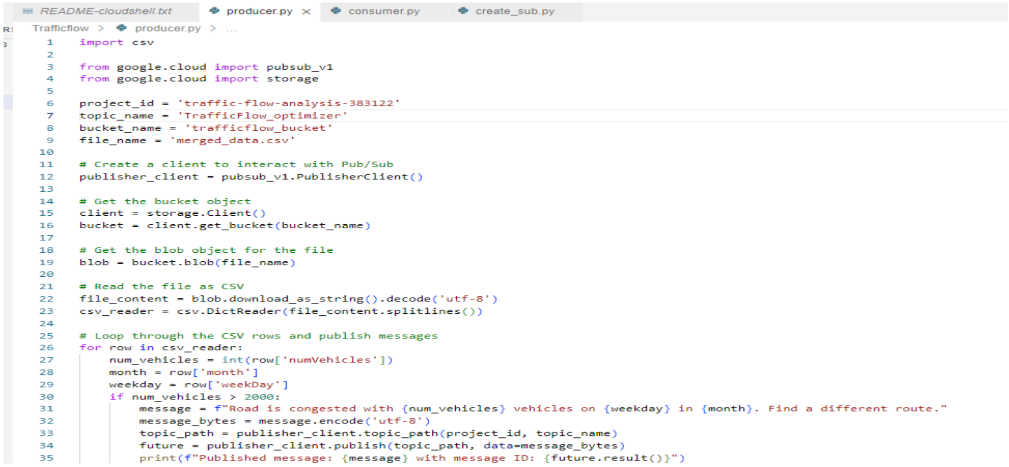
The pub/sub architecture was designed and implemented using GCP services, including Pub/Sub, Cloud Functions. The pub/sub system comprises a publisher that collects traffic data from the HighD dataset and multiple subscribers that analyze the data and provide real-time traffic information to drivers and traffic management authorities. Cloud Functions were used to process the data from Pub/Sub and trigger the analysis by the subscribers.

Collection and Processing of Data:

The HighD dataset contains traffic data from real-world scenarios, including highways and intersections. The data is collected from various sources, such as GPS and sensors on the vehicles, and is stored in a Google Cloud Storage bucket. Cloud Functions were used to trigger the collection of data from the bucket and publish it to Pub/Sub for analysis.

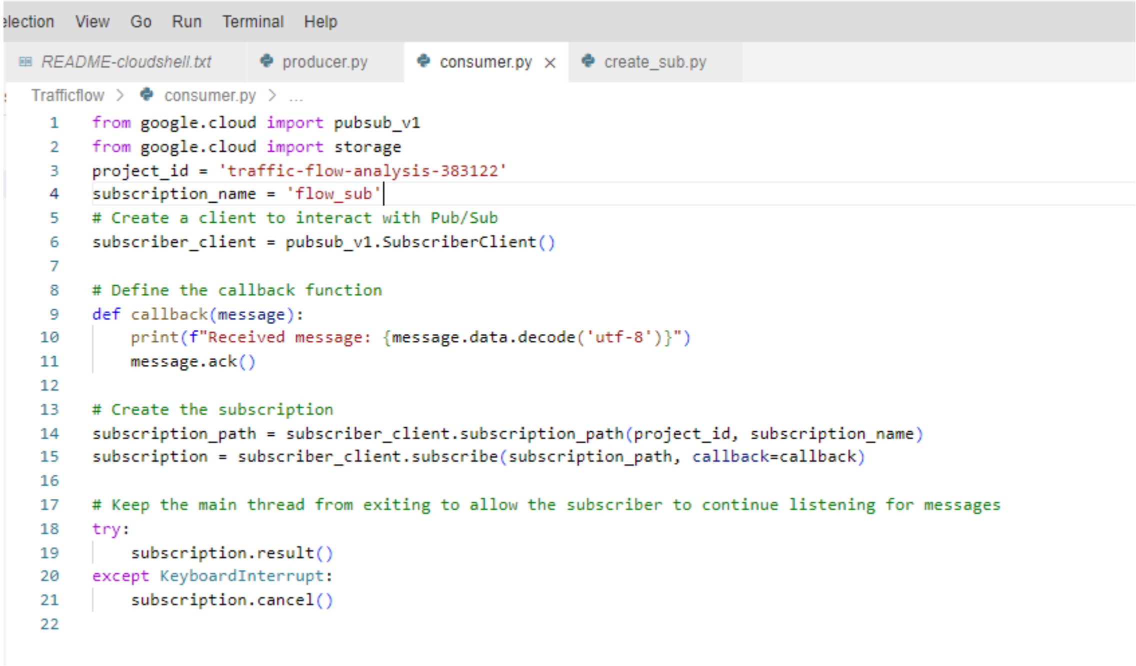
Algorithms Used for Traffic Congestion Analysis:

The subscribers analyze the data to identify congested areas and provide real-time traffic information to drivers and traffic management authorities.



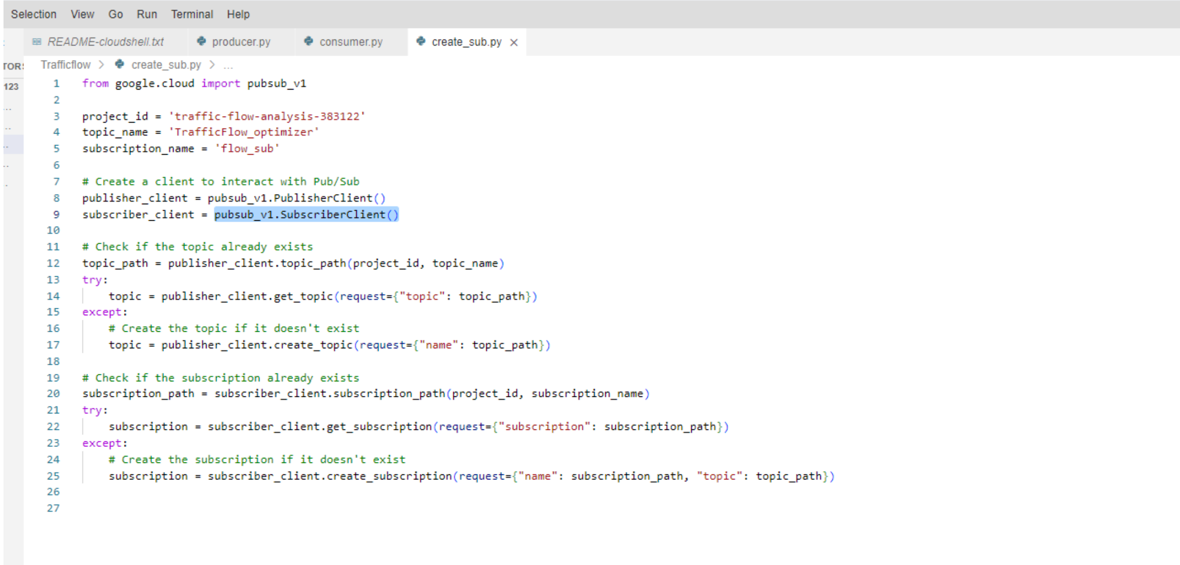
```python
import csv

from google.cloud import pubsub_v1
from google.cloud import storage

project_id = 'traffic-flow-analysis-383122'
topic_name = 'TrafficFlow_optimizer'
bucket_name = 'trafficflow_bucket'
file_name = 'merged_data.csv'

# Create a client to interact with Pub/Sub
publisher_client = pubsub_v1.PublisherClient()

# Get the bucket object
client = storage.Client()
bucket = client.get_bucket(bucket_name)

# Get the blob object for the file
blob = bucket.blob(file_name)

# Read the file as CSV
file_content = blob.download_as_string().decode('utf-8')
csv_reader = csv.DictReader(file_content.splitlines())

# Loop through the CSV rows and publish messages
for row in csv_reader:
    num_vehicles = int(row['numVehicles'])
    month = row['month']
    weekday = row['weekDay']
    if num_vehicles > 2000:
        message = f"Road is congested with {num_vehicles} vehicles on {weekday} in {month}. Find a different route."
        message_bytes = message.encode('utf-8')
        topic_path = publisher_client.topic_path(project_id, topic_name)
        future = publisher_client.publish(topic_path, data=message_bytes)
        print(f"Published message: {message} with message ID: {future.result()}")
```

This is the producer.py, it creates a client that will interact with the pub/sub and get the bucket object, Once it receives the bucket object will get the blob object. Once the connection has been established it will read the file as csv and loop through it and check if the number of vehicles exceeds a certain threshold, if it does it will send a message saying that the road is congested.



```python
from google.cloud import pubsub_v1
from google.cloud import storage
project_id = 'traffic-flow-analysis-383122'
subscription_name = 'flow_sub'
# Create a client to interact with Pub/Sub
subscriber_client = pubsub_v1.SubscriberClient()

# Define the callback function
def callback(message):
    print(f"Received message: {message.data.decode('utf-8')}")
    message.ack()

# Create the subscription
subscription_path = subscriber_client.subscription_path(project_id, subscription_name)
subscription = subscriber_client.subscribe(subscription_path, callback=callback)

# Keep the main thread from exiting to allow the subscriber to continue listening for messages
try:
    subscription.result()
except KeyboardInterrupt:
    subscription.cancel()
```

This is the consumer.py that will create a subscription and receive the messages from the producer.py



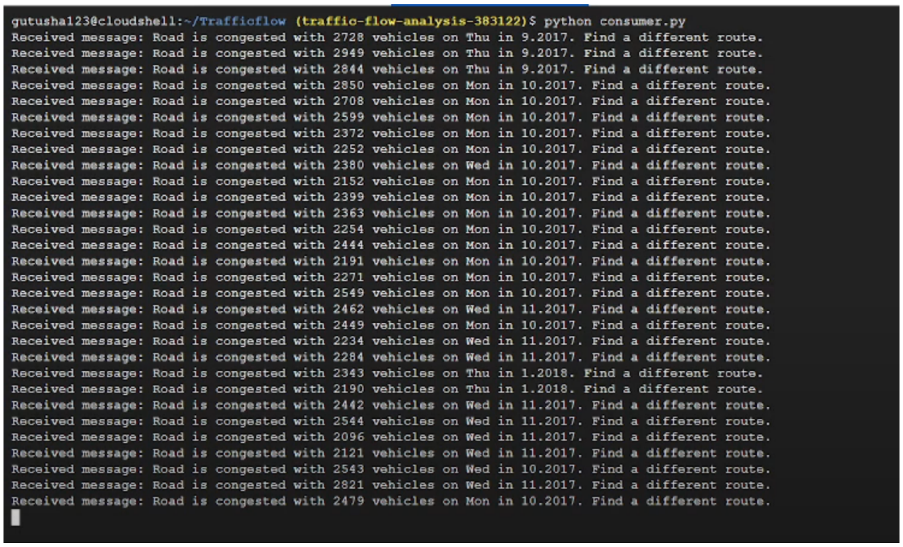
Selection View Go Run Terminal Help
README-cloudshell.txt
producer.py
consumer.py
create_sub.py ×
Trafficflow > create_sub.py > ...

```python
from google.cloud import pubsub_v1

project_id = 'traffic-flow-analysis-383122'
topic_name = 'TrafficFlow_optimizer'
subscription_name = 'flow_sub'

# Create a client to interact with Pub/Sub
publisher_client = pubsub_v1.PublisherClient()
subscriber_client = pubsub_v1.SubscriberClient()

# Check if the topic already exists
topic_path = publisher_client.topic_path(project_id, topic_name)
try:
    topic = publisher_client.get_topic(request={"topic": topic_path})
except:
    # Create the topic if it doesn't exist
    topic = publisher_client.create_topic(request={"name": topic_path})

# Check if the subscription already exists
subscription_path = subscriber_client.subscription_path(project_id, subscription_name)
try:
    subscription = subscriber_client.get_subscription(request={"subscription": subscription_path})
except:
    # Create the subscription if it doesn't exist
    subscription = subscriber_client.create_subscription(request={"name": subscription_path, "topic": topic_path})


```

This is the create.py that will create the topic, and allow for the consumers and producers to communicate with each other.

Results

The results from running this cloud based pub/sub traffic flow analysis application is that after the producer extracts the necessary data from the project bucket it analyzes it and according to the resultant decision sends a message as a callback to the Consumer.



```
gutusha123@cloudshell:~/Trafficflow (traffic-flow-analysis-383122)$ python consumer.py
Received message: Road is congested with 2728 vehicles on Thu in 9.2017. Find a different route.
Received message: Road is congested with 2949 vehicles on Thu in 9.2017. Find a different route.
Received message: Road is congested with 2844 vehicles on Thu in 9.2017. Find a different route.
Received message: Road is congested with 2850 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2708 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2599 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2372 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2252 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2380 vehicles on Wed in 10.2017. Find a different route.
Received message: Road is congested with 2152 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2399 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2363 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2254 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2444 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2191 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2271 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2549 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2462 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2449 vehicles on Mon in 10.2017. Find a different route.
Received message: Road is congested with 2234 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2284 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2343 vehicles on Thu in 1.2018. Find a different route.
Received message: Road is congested with 2190 vehicles on Thu in 1.2018. Find a different route.
Received message: Road is congested with 2442 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2544 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2096 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2121 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2543 vehicles on Wed in 10.2017. Find a different route.
Received message: Road is congested with 2821 vehicles on Wed in 11.2017. Find a different route.
Received message: Road is congested with 2479 vehicles on Mon in 10.2017. Find a different route.
```

Once the application has read the CSV, the traffic analysis application generates the results and highlights the data and time when a particular route is congested, it also tells us the amount of vehicles the route is congested with. These results can be utilized by commuters to identify alternative routes that can avoid congestion, and by traffic managers so that they can find more optimal routes for users.

There are many benefits of analyzing traffic data, but due to the limited time constraints we were only able to highlight possible areas of congestion. Utilizing personalized route recommendations individual commuters can reduce commute times, fuel consumption, and air pollution while also enhancing overall driving experience. In short, the analysis of traffic data can provide crucial information and provide better route options to commuters. We designed this in a way where it is flexible and can assist a wide range of users.

Conclusion

The development of a traffic management system application using the HighD dataset can potentially improve how we handle traffic on highways. With the population increase and the need for all households to need multiple vehicles, this will make it a bigger problem for traffic that'll affect the safety of commuters and possibly businesses.

This pub/sub architecture will detect and analyze traffic congestion by processing data from the HighD dataset and provide potential solutions. By figuring out the regions where it has an increase in traffic, the system will help law enforcement and regular drivers to find a different route to save time and lessen the traffic flow in that congested zone. This system has the potential to vastly reduce the congestion of traffic and make it easier for all commuters making it a valuable tool for any user or business.

By using a pub/sub architecture, the system can detect and analyze traffic congestion by processing data from the HighD dataset. The data can include real-time traffic information, such as vehicle speed, volume, and location, which can help identify congested areas. Once identified, the system can provide potential solutions to alleviate traffic congestion, such as suggesting alternate routes for drivers, adjusting traffic signals, or even temporarily closing off certain lanes to redirect traffic.

The system's ability to detect and analyze traffic congestion in real-time can be a game-changer for law enforcement and regular drivers alike. Law enforcement can quickly respond to accidents or incidents, while regular drivers can avoid congested areas and save time on their daily commute. This can improve the overall safety and efficiency of highways, reducing travel time, fuel consumption, and carbon emissions.

Moreover, businesses that rely on efficient transportation can benefit from the system's ability to optimize routes and avoid congested areas. By reducing transportation time and increasing efficiency, businesses can save money and increase their productivity.

Overall, the development of a traffic management system application using the HighD dataset has the potential to vastly improve the management of traffic congestion on highways. It can be a valuable tool for any user or business and can significantly enhance the safety and efficiency of highways.

Video Links:

1. Demo Video Link: https://drive.google.com/file/d/19RK9XBZR-Peh76tLYzc9vGPEEDBt91-b/view?usp=sharing
2. Presentation Video Link: https://drive.google.com/file/d/1s_-Kc_XQKT3IXRHvDyIFezejpLo49YPr/view?usp=sharing

References

1. Krajewski R, Bock J, Kloeker L, and Eckstein L (2018). The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems(2018). *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2118-2125, https://doi.org/10.1109/ITSC.2018.8569552