**Course:**

E63 (Big Data Analytics) Harvard Extension School - Spring 2016

**Professor:**

Prof. Zoran B. Djordjević

**Topic:**

Spark: Event Stream Processing and Spark mllib

**Purpose of the project:**

Read the live highway information feed provided by Massachusetts Department of Transportation. Send the live information from the feed to Kakfa cluster. Consume the processed feed with a windowed streaming Kafka consumer so that end user can see a graph of the average speed on Massachusetts highways for last N minutes (N should be configurable).

Also, predict the average speed on Massachusetts highways for last N/10 minutes based on the speed data of last N minutes. Show the predicted to the user on the same graph.

**Can the running instance of project be accessed on internet:**

Yes. Project is deployed on AWS and can be accessed at: **https://s3.amazonaws.com/e63-course-final-project/html/final\_project\_e63\_visualization.html**

**Data set:**

Massachusetts DoT live highway travel time feed ( <https://www.massdot.state.ma.us/feeds/traveltimes/RTTM_feed.aspx>) which is updated almost every minute.

More information about this feed is provided by MassDOT here: <http://www.massdot.state.ma.us/DevelopersData.aspx>

Challenges:

Keeping the application running continuously all the time.

**Real world use of the project:**

The tool that comes out of this project can be used by law enforcement officers, town planners, Department of Transport personnel in any situation/decision that needs live speed data on the highways.

**Potential future enhancements to the project:**

The time period for which average speed data is shown to the user, can be made configurable on the UI so that end user can change the time period in turn making the tool fully customizable.

More live information about the highways (like accidents, warnings, construction work) can also be shown on the graph.

The prediction engine that is used currently can be enhanced by constantly improving the prediction model. The prediction model can be improved by comparing the predictions to the real data available later corresponding to the predictions.

**There are four components to this project** (XML feed reader, message producer, message consumer, Speed predictor and visualization):

Live XML feed reader:

1. Reads the MassDoT live highway xml feed ( <https://www.massdot.state.ma.us/feeds/traveltimes/RTTM_feed.aspx>) once every 5 mins (this time is configurable)
2. Reads the speed info provided for different highways.
3. Calculates the average speed for each highway.
4. Sends the list of highways and average speeds to the message producer component

Message producer component does the following:

1. Reads the MassDoT live highway xml feed ( <https://www.massdot.state.ma.us/feeds/traveltimes/RTTM_feed.aspx>) once every 5 mins (this time is configurable)
2. Reads the speed info provided for different highways.
3. Calculates the average speed for each highway.
4. Creates a Kafka message for each highway name and average speed combination.
5. Sends the Kafka message to Kafka broker.

Message consumer does the following:

1. Creates a Spark windowed streaming consumer (in which window duration= 1 hour, slide interval=5 mins, streaming batch duration=5 mins). All are configurable.
2. Reads the Kafka messages from Kafka broker.
3. Once every slide interval:
   1. It deletes the output csv file if it exists
   2. Creates a new output csv file.
   3. It writes the highway and average speed information for each highway to the output csv file.
   4. So, the csv at any point contains highway speed information for last window duration (in this case 1 hour).
   5. Uses Linear Regression and predicts average speed of each highway for next 5 mins.
   6. Writes the prediction to the csv file.
4. So the csv file at any point contains speed information for last 1 hr and speed prediction for next 5 mins.

Speed predictor:

1. Reads the speed information of last 1 hr
2. Uses Spark mllib library and applies Linear Regression learning to the data.
3. Uses the linear regression model created to predict average speed for all highways for next 5 mins.
4. Writes the predicted speeds alongwith actual last 1 hr speeds to the csv file

Visualization component (is an html component with D3 javascript library):

1. Reads the csv file generated by Message consumer component.
2. Displays a graph of Time and Speed.
3. In this graph, a line is plotted for each highway.
4. The graph average speed for last 1 hr and speed predictions for next 5 mins.
5. Refreshing the page, displays latest speeds

**Information about the dataset:**

The dataset is live highway info xml feed hosted by Massachusetts Department of Transportation live highway travel time feed ( <https://www.massdot.state.ma.us/feeds/traveltimes/RTTM_feed.aspx>)

It is updated almost every minute.

More information about this feed is provided by MassDOT here: <http://www.massdot.state.ma.us/DevelopersData.aspx>

Size of the data:

Every access of data gives 1 MB of data.

And the xml feed is consumed once every 5 mins.

So this project processes 1MB of data every 5 mins.

Data visualized is for last 1 hr (60 mins).

And in 1 hr, xml feed is read 12 times (once every 5 mins).

So, in 60 mins of time, the amount of data consumed = 12 \* 1MB = 12MB.

So, to summarize this project takes as input a live stream (of 1MB on each load) and is processing 12MB of data at any given point.

**This is a diagram showing how the different components work**:

Live Xml Feed Reader

write

write

Kafka message <Highway name, speed >

Kafka broker

Kafka message <Highway name, speed >

Live XML feed hosted by MassDoT (updated every minute)

Read once every 5 mins

List of highway names and their average speeds

Spark Windowed Streaming consumer

started with spark-submit

(window=1 hr=60 mins)

(slide interval=5 mins)

(streaming batch=5mins)

Message Producer

(started with spark-submit)

Html file for visualization

(uses D3 library) and reads the CSV file

Graph showing average speeds on each highway (for last 1 hr and predicted speed for next 5 minseds for each highway)

display

read

CSV file on permanent storage

(Contains highway name and average speeds for last 1 hrand speed prediction for next 5 mins)

Speed predictor (uses linear regression to predict speed for next 5 mins)

**Where is the project deployed:**

I have this project deployed and running at 2 places:

1. On Cloudera VM on my local machine
2. On AWS (Amazon Web Services)

Below are pictorial representations of each environment.

This is a diagram showing how the different components work **on my local Cloudera VM**:

Live Xml Feed Reader

write

write

Kafka message <Highway name, speed >

Kafka message <Highway name, speed >

Live XML feed hosted by MassDoT (updated every minute)

Read once every 5 mins

List of highway names and their average speeds

Spark Windowed Streaming consumer

started with **spark-submit on the VM**

(window=1 hr=60 mins)

(slide interval=5 mins)

(streaming batch=5mins)

Message Producer

(started with **spark-submit on the VM**)

CSV file on the VM

(Contains highway name and average speeds for last 1 hrand speed prediction for next 5 mins)

read

Html file for visualization

(uses D3 library) and reads the CSV file

Kafka broker (**on VM**)

Graph showing average speeds on each highway (for last 1 hr and predicted speed for next 5 minseds for each highway)

display

Speed predictor (uses linear regression to predict speed for next 5 mins)

This is a diagram showing how the different components work when deployed **on AWS**:

Live Xml Feed Reader

write

write

Kafka message <Highway name, speed >

Kafka message <Highway name, speed >

Live XML feed hosted by MassDoT (updated every minute)

Read once every 5 mins

List of highway names and their average speeds

Message Producer

(started with spark-submit **on AWS Spark cluster**)

Spark Windowed Streaming consumer

started with **spark-submit on AWS Spark cluster**

(window=1 hr=60 mins)

(slide interval=5 mins)

(streaming batch=5mins)

CSV file **in AWS S3 bucket**

(Contains highway name and average speeds for last 1 hrand speed prediction for next 5 mins)

read

Html file for visualization

(uses D3 library) and reads the CSV file

Kafka broker (**on AWS EC2**)

Graph showing average speeds on each highway (for last 1 hr and predicted speed for next 5 minseds for each highway)

display

Speed predictor (uses linear regression to predict speed for next 5 mins)

**How to deploy and run the application locally (either on a VM or on the host machine):**

1) Get hold of the following files (I have provided in the assignment submission):

- FinalProject\_e63.jar

- html/final\_project\_e63\_visualization.html

2) Start the zoopker server

3) Start the kafka server

4) Create a kafka topic

5) Start a python server I the directory that contains *html/final\_project\_e63\_visualization.html* file

6) Access *html/final\_project\_e63\_visualization.html* through the web browser in this way: <http://localhost:8888/html/final_project_e63_visualization.html>

7) Run the consumer with this command:

spark-submit --class e63.course.streaming.consumer. HighwayInfoKafkaSlidingWindowConsumer FinalProject\_e63.jar <kafkaBrokers> <kafka\_topics> <zookeeper> &

e.g. when I run this on AWS Spark cluster I run it in the following way:

spark-submit --class e63.course.streaming.consumer. HighwayInfoKafkaSlidingWindowConsumer FinalProject\_e63.jar ec2-54-173-167-157.compute-1.amazonaws.com:9092 highway\_info\_kafka\_topic ec2-54-173-167-157.compute-1.amazonaws.com:2181 &

8) Run the producer with this command:

spark-submit --class e63.course.streaming.producer.HighwayInfoKafkaProducer FinalProject <kafkaServerIPAndPort> <kafkaTopic>

e.g. when I run this on AWS Spark cluster I run it in the following way:

spark-submit --class e63.course.streaming.producer.HighwayInfoKafkaProducer FinalProject-0.0.1-SNAPSHOT-jar-with-dependencies.jar ec2-54-173-167-157.compute-1.amazonaws.com:9092 highway\_info\_kafka\_topic &

9) Wait for around 10 mins for data to be collected.

10) Load the html file and check visual representation of the data:

<http://localhost:8888/html/final_project_e63_visualization.html>

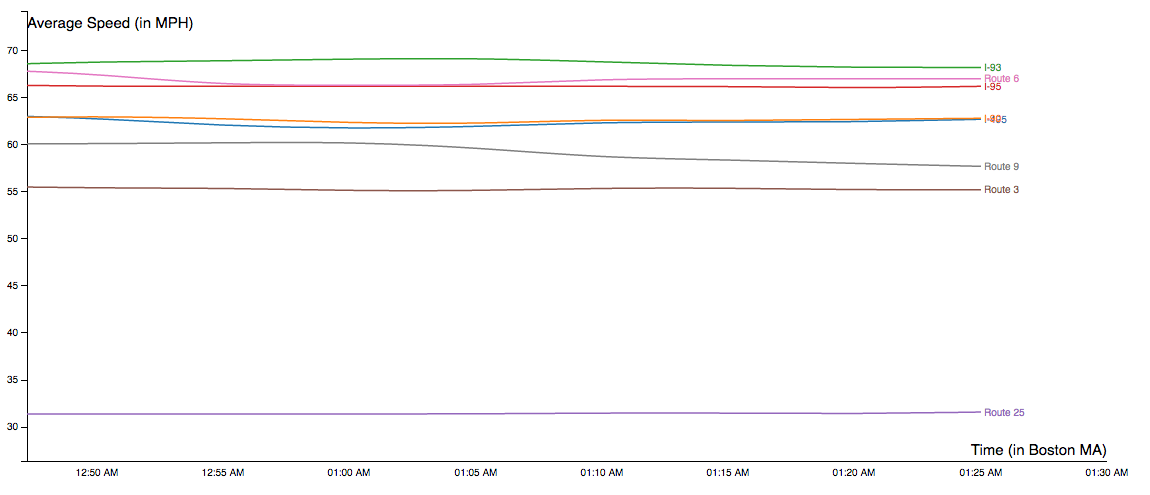
**Output results of the application:**

Output of the project is a graph that displays:

X-axis: Time scale of upto last 1 hr (separated by 5 mins)

Y-axis: Speed in MPH

Each line represents one highway.

**Lessons learnt:**

1. Processing a live stream and applying machine learning on it is not trivial
2. Keeping the streaming application running continuously is a challenge.

**Potential future enhancements to the project:**

1. Different streaming libraries other than Spark can be used e.g. Apache Storm
2. More pieces of live information about each highway can be processed and displayed to the user, e.g. accidents, construction work
3. The machine learning module that predicts the speed of next 5 mins can be improved by giving it the input of actual instances of predicted data.
4. The time durations of streaming window, streaming interval can be made more customizable so that the end user can customize them.