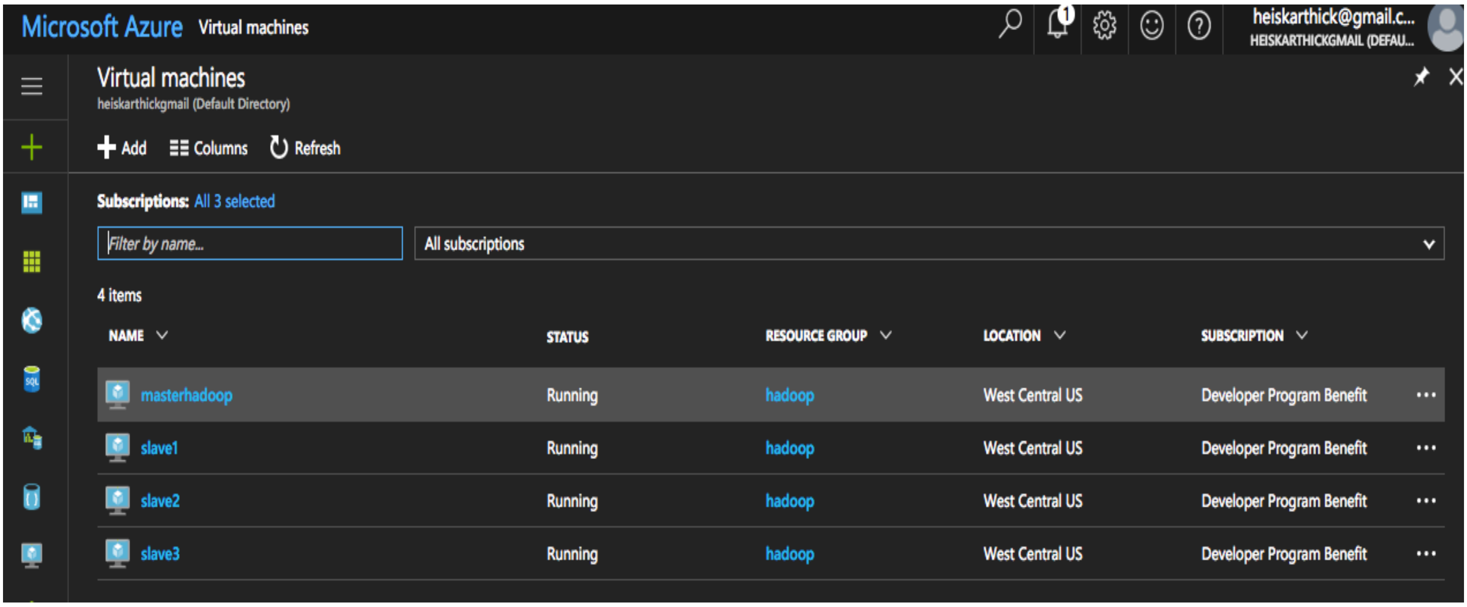
Introduction:

The aim of this project is to set up a HDFS cluster in Microsoft azure and to run MapReduce program to process on Yellow Cab data, from New York City Taxi And Limousine Commission dataset which include fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts.

System behaviour:

1. This system consists of a master and 3 clones of the master called as slaves.
2. This cluster has one name node and 3 data nodes with a replication factor of 3.
3. Start the Hadoop deamons.
4. Download the Yellow Cab data from New York City Taxi in master which is then pushed to datanode in all 3 slaves.
5. MapReduce program is written in python language to run a map reduce function on the downloaded dataset.

Cluster formation



Virtual machines

heiskarthickgmail (Default Directory)

Subscriptions: All 3 selected

Filter by name...

All subscriptions

4 items

NAME	STATUS	RESOURCE GROUP	LOCATION	SUBSCRIPTION
masterhadoop	Running	hadoop	West Central US	Developer Program Benefit
slave1	Running	hadoop	West Central US	Developer Program Benefit
slave2	Running	hadoop	West Central US	Developer Program Benefit
slave3	Running	hadoop	West Central US	Developer Program Benefit

Description of MapReduce program:

There are two separate python files called as map and reduce in master node.

Pre-processing the dataset in Map.py

The first 5 months data of Yellow Cab NYC in the year 2010 is taken for this project.

1. Firstly, Removing the headers and blank space is done at the beginning as it is required to remove these in the .csv data files.
2. Remove the white spaces that are there in the file.
3. separate each word as tokens that are separated by comma (,).
4. Take the vendor name separately at line 0, passenger count at line 3, trip distance at line 4, fare amount at line 12, payment at line 11 and date at line 10.
5. Finally, Create dictionaries to store the mapped data of each vendor according to the trip distance, fare amount etc.

The dictionaries are:

- No. of trips per vendor

  The trip distance will be appended to this dictionary at every iteration
- Fare amount vendor

  Fare amount of each vendor will be appended to this dictionary
- Payment types

  We append the total number of times each payment method was used per vendor
- Months in year

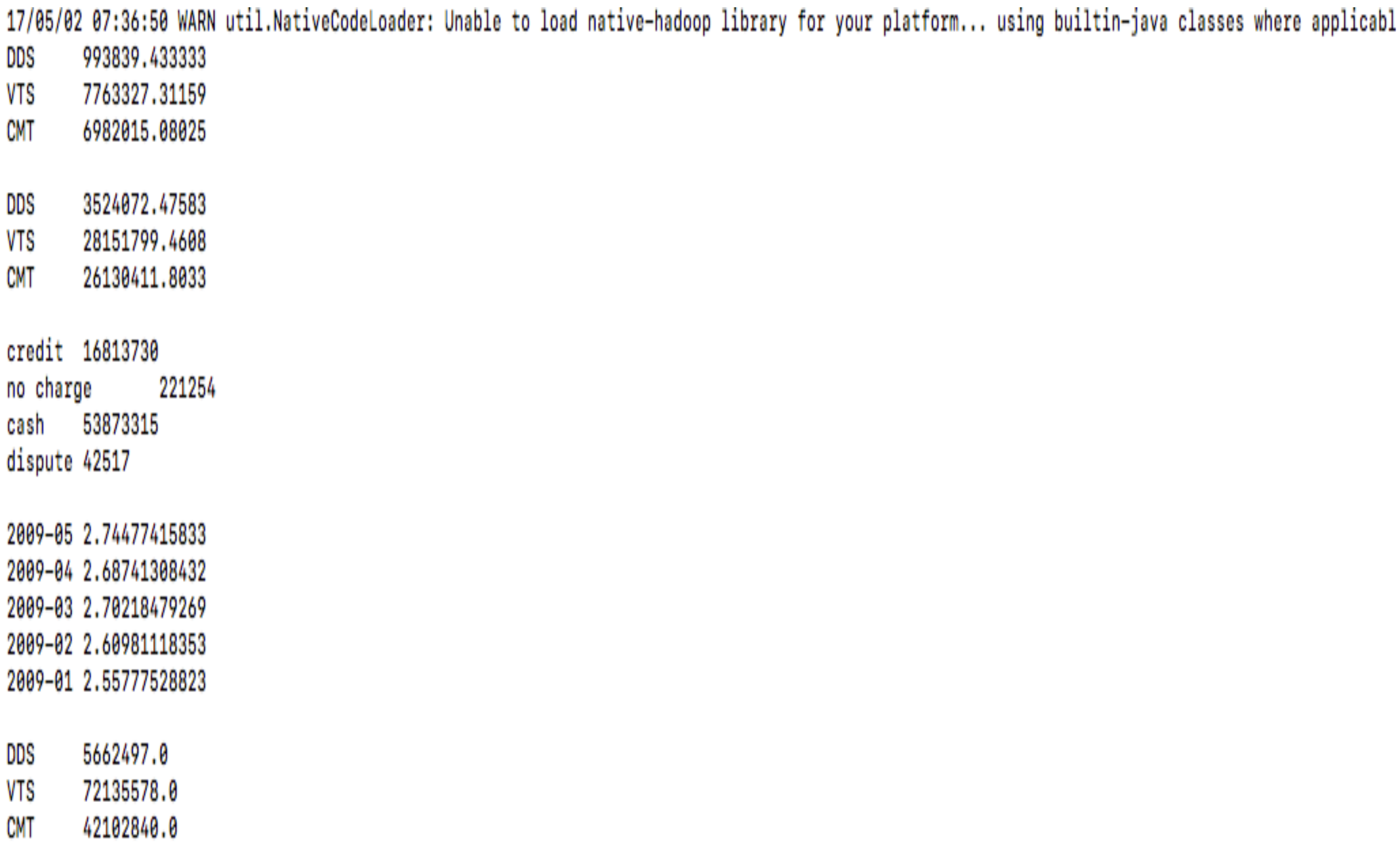
  Month will be the key and the trip distance per month will be kept in this dictionary
- Total no. of passengers per vendor

  We keep the passenger count per vendor here

Reduce.py

Reducer takes all the key, value pairs from mapper, aggregates and generates the desired output.

Data Analysis:



```
17/05/02 07:36:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicabl
DDS     993839.433333
VTS     7763327.31159
CMT     6982015.08025

DDS     3524072.47583
VTS     28151799.4608
CMT     26130411.8033

credit  16813730
no charge       221254
cash    53873315
dispute 42517

2009-05 2.74477415833
2009-04 2.68741308432
2009-03 2.70218479269
2009-02 2.60981118353
2009-01 2.55777528823

DDS     5662497.0
VTS     72135578.0
CMT     42102840.0
```

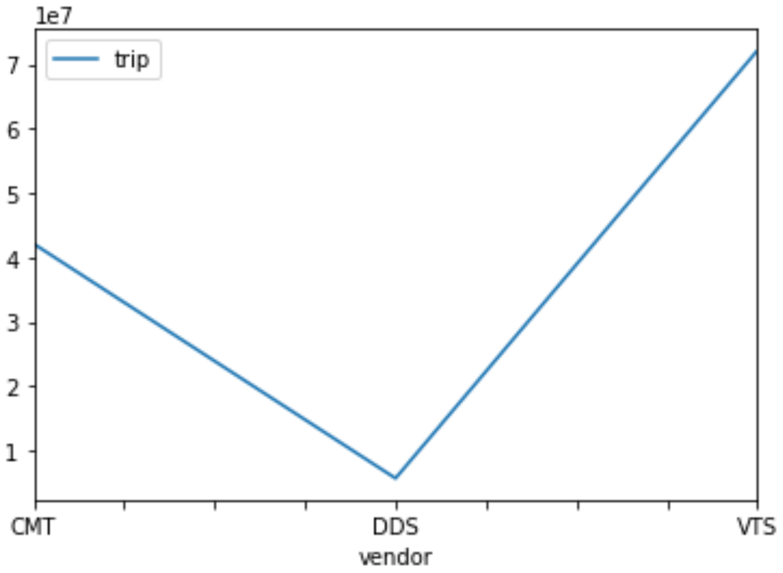
The mostly used payment type

Cash	**53873315**
Credit	**16813730**
No charge	**221254**
Dispute	**42517**

The table shows that cash was majorly used as payment type in the year 2010.

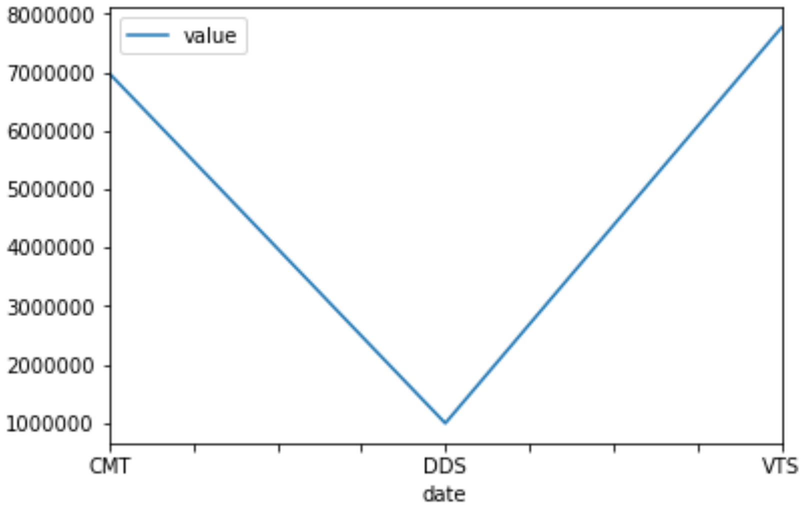
No. of trips took by each vendor in 2010

DDS	**5662497.0**
VTS	**72135578**
CMT	**42102840**



Total trip distance by each vendor by each month

DDS	**993839.433333**
VTS	**7763327.31159**
CMT	**6982015.08025**

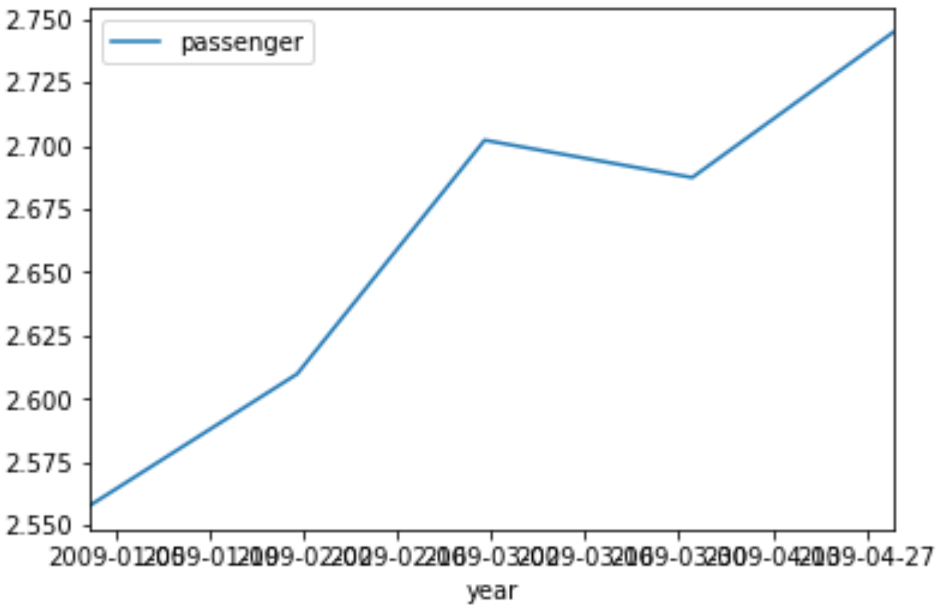


Total fare by each vendor Jan to May in 2010

DDS	**3524072.47583**
VTS	**28151799.4608**
CMT	**26130411.8033**

Average no. of passengers per month per day

2009-05	**2.74477415833**
2009-04	**2.68741308432**
2009-03	**2.70218479269**
2009-02	**2.60981118353**
2009-01	**2.55777528823**



Challenges faced and its solution:

1. **Creating a cluster:**
   - The master node's port number had to be removed from the hdfs-site.xml as this led to an error when I tried to SSH the master node and name node did not start because it was in the master node.

- The disks were found that they could be downloaded to the local HDD if the **“ActiveSAS” state** was enabled. But this led to an error when the clusters were created and had to be **unchecked**.

2. **Memory limitation:**
   As the vms have limited memory, It was challenging to download all data and pushed into cluster. Therefore, limited data is downloaded and pushed into cluster for data analysis.

Access to VMs:

Masterhadoop	ssh hadoop@ 13.78.179.87
Slave1	ssh hadoop@ 52.161.8.221
Slave2	ssh hadoop@ 52.161.29.205
Slave3	ssh Hadoop@ 52.161.8.140

Command to run:

```
hadoop jar /usr/local/Cellar/hadoop/2.8.0/libexec/share/hadoop/tools/lib/hadoop-streaming-2.8.0.jar -mapper map.py -reducer reducer.py -input /2010 -output /2010output -file map.py -file reducer.py
```

Conclusion:

Having worked in this project, I gained valued in-sights in creating vms and setting up a cluster to run MapReduce program to process datasets from NYC.