In this project, the batch layer of the lamda architecture is prepared using Apache Spark's Dataframe and basic analytics performed on the M50 road network dataset

In [3]:

```
import pyspark
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.functions import col
spark = SparkSession.builder.appName("Assignment").getOrCreate()
```

In [4]:

```
#Reading Data from CSV
df = spark.read.options(header='True', inferSchema='True')\
.csv("hdfs://master-node:9000/assignment/per-vehicle-records-2021-01-31.csv")
```

In [5]:

```
df.show()
```

```
+-----+----+-----+---+----+------+------+-----------+-----------+----+--------+--------
----+----------------+-----+---------+------+-------+----+-----+------+-----------+----
----+------------+-------------+-----------+------------+
|cosit|year|month|day|hour|minute|second|millisecond|minuteofday|lane|lanename|straddlela
ne|straddlelanename|class|classname|length|headway| gap|speed|weight|temperature|duration
|validitycode|numberofaxles|axleweights|axlespacings|
+-----+----+-----+---+----+------+------+-----------+-----------+----+--------+--------
----+----------------+-----+---------+------+-------+----+-----+------+-----------+----
----+------------+-------------+-----------+------------+
|  998|2021|    1| 31|   2|    45|     0|          0|        165|   2|    Ch 2|
0|            null|    2|      CAR|   5.2|   1.07|1.13| 71.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     1|          0|        165|   2|    Ch 2|
0|            null|    5|  HGV_RIG|  11.1|    1.1|1.34| 69.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     1|          0|        165|   1|    Ch 1|
0|            null|    5|  HGV_RIG|  11.1|   1.17|1.11| 69.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     2|          0|        165|   1|    Ch 1|
0|            null|    2|      CAR|   5.3|    1.0|0.72| 70.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     3|          0|        165|   2|    Ch 2|
0|            null|    3|      LGV|   5.3|   1.01|0.72| 71.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     4|          0|        165|   1|    Ch 1|
0|            null|    2|      CAR|   5.2|   1.62|1.63| 70.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     5|          0|        165|   2|    Ch 2|
0|            null|    3|      LGV|   5.2|   1.64|1.63| 69.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     6|          0|        165|   1|    Ch 1|
0|            null|    5|  HGV_RIG|  11.4|   1.13|1.53| 70.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     7|          0|        165|   2|    Ch 2|
0|            null|    5|  HGV_RIG|  11.4|   1.39|1.83| 71.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     8|          0|        165|   1|    Ch 1|
0|            null|    5|  HGV_RIG|  11.1|   1.36|1.31| 69.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     8|          0|        165|   2|    Ch 2|
0|            null|    2|      CAR|   5.2|   1.57|1.22| 69.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|     9|          0|        165|   1|    Ch 1|
0|            null|    2|      CAR|   5.2|   1.16|0.92| 70.0|   0.0|        0.0|        0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    10|          0|        165|   2|    Ch 2|
0|            null|    5|  HGV RIG|  11.5|   1.34|1.63| 71.0|   0.0|        0.0|        0
```

```
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    11|          0|        165|   1|    Ch 1|
0|            null|    5|  HGV_RIG|  11.1|   1.72|1.93| 69.0|   0.0|        0.0|       0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    12|          0|        165|   2|    Ch 2|
0|            null|    2|      CAR|   5.2|   1.34|1.12| 71.0|   0.0|        0.0|       0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    13|          0|        165|   1|    Ch 1|
0|            null|    2|      CAR|   5.1|   1.17|0.92| 69.0|   0.0|        0.0|       0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    14|          0|        165|   1|    Ch 1|
0|            null|    2|      CAR|   5.1|   1.28|1.23| 69.0|   0.0|        0.0|       0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    14|          0|        165|   2|    Ch 2|
0|            null|    5|  HGV_RIG|  11.3|   1.07|1.34| 69.0|   0.0|        0.0|       0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    15|          0|        165|   1|    Ch 1|
0|            null|    2|      CAR|   5.1|   1.02|1.03| 69.0|   0.0|        0.0|       0
|           0|            0|       null|        null|
|  998|2021|    1| 31|   2|    45|    16|          0|        165|   2|    Ch 2|
0|            null|    5|  HGV_RIG|  11.5|   1.27|1.31| 71.0|   0.0|        0.0|       0
|           0|            0|       null|        null|
+-----+----+-----+---+----+------+------+-----------+-----------+----+--------+--------
----+----------------+-----+---------+------+-------+----+-----+------+-----------+----
----+------------+-------------+-----------+------------+
only showing top 20 rows
```

In [7]:

```
#usage of Irish road network in terms of percentage grouped by vehicle category
total = df.count() # Total count

category_percentage = df.groupBy("classname")\
.count()\
.withColumn('perc_of_count_total', (F.col('count') / total) * 100 )\
.show()
```

```
+---------+------+--------------------+
|classname| count| perc_of_count_total|
+---------+------+--------------------+
|      CAR|918254|   82.97585871619985|
|  HGV_ART| 33805|    3.05470915879608|
|      BUS| 10519|  0.9505246455073502|
|  HGV_RIG| 30866|  2.7891333499600597|
|     null|    50|0.004518132168016684|
|  CARAVAN|  5887|  0.5319648814622845|
|      LGV|104580|   9.450125242623697|
|    MBIKE|  2691| 0.24316587328265796|
+---------+------+--------------------+
```

In [9]:

```
#highest and lowest hourly fows on M50 - show the hours and total number of vehicle count
s

#M50 Junctions
m50=[1013,1012,1500,1501,1502,1508,1503,1509,1504,1505,1506,1507,15010,15011,15012]

#Hourly count in M50 Road
groupedData = df.select('hour').filter(df.cosit.isin(m50)).groupBy('hour').count()

#Ordering
hourlyCount = groupedData.orderBy('count')

#Lowest Hourly Flow

MinHour = hourlyCount.first()
print("Lowest Hourly Flow = ", MinHour)
```

```
#Highest Hourly Flow

MaxHour = groupedData.orderBy('count', ascending=False).first()
print("Highest Hourly Flow = " , MaxHour)
```

```
Lowest Hourly Flow =  Row(hour=3, count=585)
Highest Hourly Flow =  Row(hour=15, count=18711)
```

In [11]:

```
# Morning = 6.00 - noon (12.00)
#Evening = 15.00 - 21.00
morningHours = [6,7,8,9,10,11]
eveningHours = [15,16,17,18,19,20,21]
hourlyCount = df.select('hour').filter(df.cosit.isin(m50)).groupBy('hour').count()

print("Morning Rush hour Counts")
morningRushHourCount = hourlyCount.filter(hourlyCount.hour.isin(morningHours))\
.orderBy('hour')\
.show()

print("Evening Rush Hour Counts")
eveningRushHourCount = hourlyCount.filter(hourlyCount.hour.isin(eveningHours))\
.orderBy('hour')\
.show()
```

```
Morning Rush hour Counts
+----+-----+
|hour|count|
+----+-----+
|   6| 3944|
|   7| 6500|
|   8| 5530|
|   9| 6641|
|  10| 9088|
|  11|11947|
+----+-----+

Evening Rush Hour Counts
+----+-----+
|hour|count|
+----+-----+
|  15|18711|
|  16|17979|
|  17|16060|
|  18|12647|
|  19|10877|
|  20|10383|
|  21| 7136|
+----+-----+
```

In [13]:

```
#average speed between each junction on M50
columns= ["cosits","Junctions" ,"index"]
data = [(1013, "Junction 1 -  2" ,1),
        (1012, "Junction 2 -  3",2),
       (1500, "Junction 3 -  4",3),
       (1501, "Junction 4 -  5",4),
       (1502, "Junction 5 -  6",5),
       (1508, "Junction 6 -  7",6),
        (1503, "Junction 7 -  9",7),
        (1509, "Junction 9 - 10",8),
        (1504, "Junction 10 - 11",9),
        (1505, "Junction 11 - 12",10),
        (1506, "Junction 12 - 13",11),
        (1507, "Junction 13 - 14",12),
        (15010, "Junction 14 - 15",13),
        (15011, "Junction 15 - 16",14),
        (15012, "Junction 16 - 17",15)
       ]
```

```
rdd = spark.sparkContext.parallelize(data)
cositdf = rdd.toDF(columns)

#Average Speed in Cosit M50
avgSpeed = df.select("speed","cosit").filter(df.cosit.isin(m50))\
.groupBy('cosit')\
.avg("speed")

joined = cositdf.join(avgSpeed,cositdf.cosits ==  avgSpeed.cosit,"inner")
order = joined.orderBy("index").select("Junctions","avg(speed)")
order.show()
```

```
+----------------+------------------+
|       Junctions|        avg(speed)|
+----------------+------------------+
| Junction 1 -  2| 68.53492193919475|
| Junction 2 -  3| 86.61353856338961|
| Junction 3 -  4| 93.74959897337183|
| Junction 4 -  5|101.33084897730457|
| Junction 5 -  6|102.36304050088046|
| Junction 6 -  7| 98.64505637467477|
| Junction 7 -  9|102.18442775736273|
| Junction 9 - 10| 98.35261039422281|
|Junction 10 - 11|101.99216139028985|
|Junction 11 - 12| 99.69152287044645|
|Junction 12 - 13|102.79217719132893|
|Junction 13 - 14|102.74182687085913|
|Junction 14 - 15| 105.0165992764418|
|Junction 15 - 16|101.79879709487064|
|Junction 16 - 17|105.10443959243086|
+----------------+------------------+
```

In [15]:

```
#top 10 locations with highest number of counts of HGVs (class)

HGVcount = df.select('cosit','classname').filter(df.classname.like("HGV%")).groupBy('cos
it').count()

#Reading Cosit date

cositdf = spark.read.csv("hdfs://master-node:9000/assignment/cosit_data.csv")
#Heighest # of SVG result joining with cositDF

cositdata = HGVcount.join(cositdf,HGVcount.cosit ==  cositdf._c1,"inner")\
.orderBy('count',ascending=False)\
.limit(10)

#data.select(col("Name").alias("name"), col("askdaosdka").alias("age"))
cositdata.select(col("cosit"), col("_c2").alias("SiteName"), col("_c3").alias("Descripti
on"),col("count"))\
.show()
```

```
+-----+---------------+--------------------+-----+
|cosit|       SiteName|         Description|count|
+-----+---------------+--------------------+-----+
|  998|            998|         Test site 2|22367|
| 1508|TMU M50 015.0 S |M50 Between Jn06 ...| 1224|
| 1502|TMU M50 010.0 N |M50 Between Jn06 ...| 1186|
| 1503| TMU M50 020.0 N|M50 Between Jn07 ...|  962|
| 1501| TMU M50 005.0 N|M50 Between Jn05 ...|  923|
| 1070| TMU N07 001.0 E|N07 Between Jn01 ...|  881|
| 1071| TMU N07 005.0 E|N07 Between Jn02 ...|  827|
| 1073|       N07 E06.5|N7 Eastbound city...|  820|
|20071| TMU N07 020.0 E|N07 Between Jn07 ...|  797|
| 1072| TMU N07 000.0 W|N07 Between Jn01a...|  774|
+-----+---------------+--------------------+-----+
```