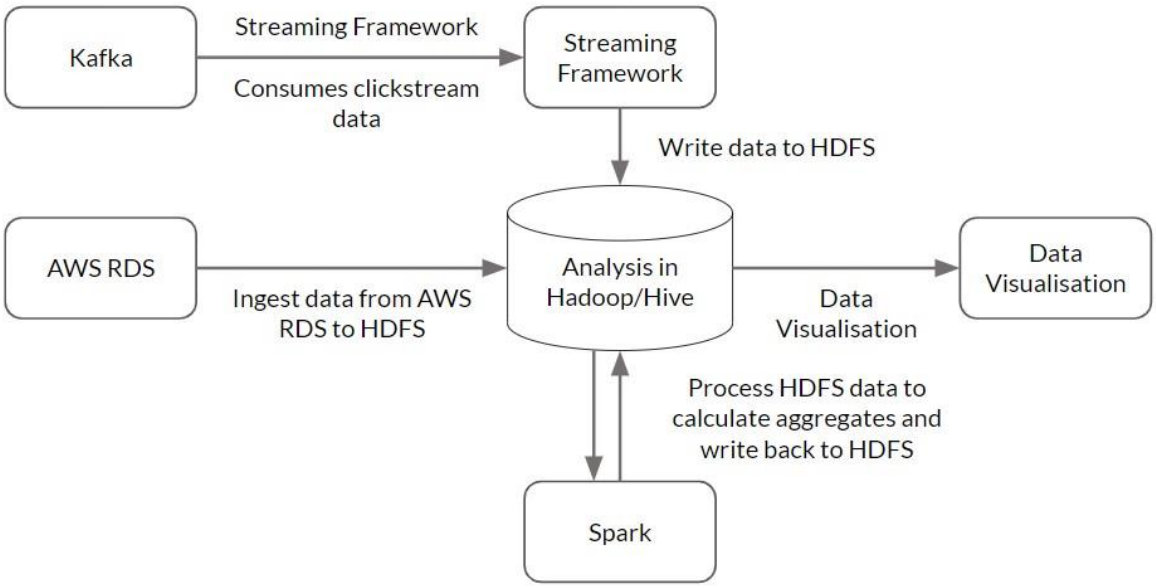
Logic For Final Submission

Overall Architecture:



For the mid submission, we carried out ingestion of stream and batch data from Kafka and RDS into HDFS. Then we created tables in a Hive database and loaded the final aggregated data into these tables.

For the final submission, we read data from the Hive tables to calculate the following metrics:

Task 5: Calculate the total different drivers for each customer
Task 6: Calculate the total rides taken by each customer
Task 7: Calculate the conversion ratio
Task 8: Count trips done on black cabs
Task 9: Calculate the total tip amount
Task 10: Count the ratings below 2 in a particular month
Task 11: Calculate the total iOS users

Hive Tables Creation and Loading Data into them:

```
hadoop@ip-172-31-33-247:~
[hadoop@ip-172-31-33-247 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> create database if not exists cab_booking ;
OK
Time taken: 0.91 seconds
hive> use cab_booking ;
OK
```

```
hadoop@ip-172-31-33-247:~
[hadoop@ip-172-31-33-247 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> create database if not exists cab_booking ;
OK
Time taken: 0.91 seconds
hive> use cab_booking ;
OK
Time taken: 0.073 seconds
hive> create table if not exists clickstream_data
    > (customer_id string ,app_version string, os_version string,lat string ,lon string ,page_id
    > string,button_id string , is_button_click varchar(3) ,is_page_view varchar(3) ,is_scroll_up
    > varchar(3) ,is_scroll_down varchar(3), click_timestamp string )
    > row format delimited fields terminated by ","
    > location '/home/hadoop/clickstream_data_flatten/';
OK
Time taken: 0.422 seconds
hive> create table if not exists booking_data
    > (booking_id string ,customer_id string ,driver_id string , customer_app_version string,
    > customer_phone_os_version string , pickup_lat double , pickup_lon double, drop_lat double,
    > drop_lon double, pickup_timestamp string , drop_timestamp string ,trip_fare int,
    > tip_amount int, currency_code string ,cab_color string, cab_registration_no string ,
    > customer_rating_by_driver int, rating_by_customer int ,passenger_count int )
    > row format delimited fields terminated by ","
    > location '/home/hadoop/booking_data_csv/';
OK
Time taken: 0.07 seconds
hive> create table if not exists datewise_data
    > (booking_date string , count int)
    > row format delimited fields terminated by ","
    > location '/home/hadoop/datewise_aggreation/';
OK
Time taken: 0.05 seconds
hive>
```

The final data is stored in tables “clickstream_data”, “booking_data”, “datewise_data” in the Hive database “cab_booking”

Tasks

The queries to perform Task 5 to Task 11 are provided in the file **queries.pdf.** To query the data, we use HiveQL.

<u>Task 5: Calculate the total different drivers for each customer</u>

- Use the database **cab_booking**
- From the table **booking_data**, select the column **customer_id**
- Group the **customer_id** values and count the entries in **driver_id** column using count(). This gives us the count of different driver ids for each customer
- Sort the result by **customer_id**

16.78s Database cab_booking Type text ?

```
select customer_id, count(driver_id) as driver_count from booking_data group by customer_id
order by customer_id;
```

Query execution screenshot:

```
INFO  : Map 1: 1/1      Reducer 2: 2/2  Reducer 3: 1/1
INFO  : Completed executing command(queryId=hive_20231108073103_ca9a9a62-3d67-4bbf-a27c-48cabe781648); Time taken:
14.952 seconds
INFO  : OK
```

application_1699426829632_0006

Query results screenshot:

Query History | Saved Queries | Query Builder | Results (100+)

	customer_id	driver_count
1	10022393	1
2	10058402	1
3	10339567	1
4	10435129	1
5	10555335	1
6	10592274	1
7	10614890	1
8	10678994	1
9	11264797	1
10	11353346	1
11	11418437	1
12	11438890	1
13	11454977	1
14	11479815	1
15	11518953	1
16	11580321	1
17	11596512	1
18	11608791	1

<u>Task 6: Calculate the total rides taken by each customer</u>

- Use the database **cab_booking**
- From the table **booking_data**, select the column **customer_id**
- Group the **customer_id** values and count the number of times a **customer_id** occurs using “count(*)”. This gives us the total no. of rides taken by each customer
- Sort the result by **customer_id**

6.57s Database cab_booking Type text

```
select customer_id, count(*) as total_ride from booking_data group by customer_id
order by customer_id;
```

Query execution screenshot:

```
INFO  : Map 1: 1/1      Reducer 2: 2/2  Reducer 3: 1/1
INFO  : Completed executing command(queryId=hive_20231023234925_386153ff-fcfc-4
6.532 seconds
INFO  : OK
```

application_1698100961546_0008

Query results screenshot:

Query History | Saved Queries | Query Builder | Results (100+)

	customer_id	total_ride
1	10022393	1
2	10058402	1
3	10339567	1
4	10435129	1
5	10555335	1
6	10592274	1

<u>Task 7: Find total visits by each customer on the booking page and total 'Book Now' button press. Also, calculate the conversion ratio.</u>

- Use the database **cab_booking**
- Use the table **clickstream_data**
- Calculate the number of times the button 'Book Now' is pressed. The Book Now **button_id** is 'fcba68aa-1231-11eb-adc1-0242ac120002' and the button is considered to be pressed when the value in **is_button_click =** 'Yes'
- Then calculate the total visits made by each customer on the booking page. The Booking **page_id** is 'e7bc5fb2-1231-11eb-adc1-0242ac120002'. The page is considered to be visited when **is_page_view** = 'Yes'
- Conversion ratio is defined as -> [(Total 'Book Now' Button Press) / (Total Visits made by customer on the booking page)]

- We calculate the conversion ratio, and round off the value to 4 decimal places

Database cab_booking Type text

```
select round(
(sum(case when button_id = "fcba68aa-1231-11eb-adc1-0242ac120002" and
is_button_click = 'Yes' then 1 end) /
sum(case when page_id = "e7bc5fb2-1231-11eb-adc1-0242ac120002" and
is_page_view = 'Yes' then 1 end)),4) as conversion_ratio
from clickstream_data;
```

Query execution screenshot:

```
INFO  : Map 1: 1/1      Reducer 2: 1/1
INFO  : Completed executing command(queryId=hive_20231023235717_6ebb548c-70a5-4138-88b2-c4d4bba8840b); Time taken:
7.0 seconds
INFO  : OK
```

Query results screenshot:

Query History | Saved Queries | Query Builder | Results (1+)

	conversion_ratio
1	0.9688

Task 8: Calculate the count of all trips done on Black cabs

- Use the database **cab_booking**
- Use the table **booking_data**
- Count all the number of trips taken by black cabs using “count(*)”. The condition for **cab_color** is applied using the “where” clause

Search data and saved documents...

Hive Add a name... Add a description...

6.64s Database cab_booking Type text

```
select count(*) as black_car_trips from booking_data where cab_color = 'black';
```

Query execution screenshot:

```
INFO  : Map 1: 1/1      Reducer 2: 1/1
INFO  : Completed executing command(queryId=hive_20231024000106_f7e433de-e536-4
5.894 seconds
INFO  : OK
```

application_1698100961546_0008

Query results screenshot:

Query History | Saved Queries | Query Builder | Results (1)

	black_car_trips
1	72

Task 9: Calculate the total tip amount on a given date to all drivers by customers

- Use the database **cab_booking**
- Use the table **booking_data**
- Select the column **pickup_timestamp** and convert its values to a date string of format 'YYYY-MM-dd' using "date_format()". Read the result in the variable **datewise**
- Group by date string and find the total tip amount on a particular date using the aggregate function "sum()"
- Sort the result by **datewise**

6.70s Database cab_booking Type text ?

```
select date_format(pickup_timestamp,'YYYY-MM-dd') as datewise, sum(tip_amount) as total_tip
from booking_data
group by date_format(pickup_timestamp,'YYYY-MM-dd')
order by datewise;
```

Query execution screenshot:

```
INFO  : Map 1: 1/1      Reducer 2: 2/2  Reducer 3: 1/1
INFO  : Completed executing command(queryId=hive_20231024002405_75c40540-b920-4
6.336 seconds
INFO  : OK
```

application_1698100961546_0009

Query results screenshot:

Query History | Saved Queries | Query Builder | Results (289)

	datewise	total_tip
1	2020-01-01	59
2	2020-01-02	95
3	2020-01-03	11
4	2020-01-04	123
5	2020-01-05	134
6	2020-01-06	189
7	2020-01-07	148
8	2020-01-08	111
9	2020-01-09	48
10	2020-01-10	77
11	2020-01-11	81
12	2020-01-12	109
13	2020-01-14	142
14	2020-01-15	338
15	2020-01-16	155
16	2020-01-17	296
17	2020-01-18	240

Task 10: Calculate the count of all the bookings with a rating below 2 in a particular month

- Use the database **cab_booking**
- Use the table **booking_data**
- Select the column **pickup_timestamp** and convert the dates to a string of format 'YYYY-MM' using "date_format()". Read the result in the variable **monthwise**
- Group by month string and find the count of all bookings with a **rating_by_customer** less than 2 using count(*). The condition for the customer rating is given by "where" clause
- Sort the result by **monthwise** to list the count of all bookings below 2 in a particular month

15.57s Database cab_booking Type text

```
select date_format(pickup_timestamp, 'YYYY-MM') as monthwise, count(*)
as total_bookings
from booking_data where rating_by_customer < 2
group by date_format(pickup_timestamp, 'YYYY-MM')
order by monthwise;
```

Query execution screenshot:

```
INFO  : Map 1: 1/1      Reducer 2: 2/2  Reducer 3: 1/1
INFO  : Completed executing command(queryId=hive_20231024002931_2c1f8283-428c-4599-9e2d-88daff9ffa93); Time taken:
14.258 seconds
INFO  : OK
```

application_1698100961546_0010

Query results screenshot:

Query History | Saved Queries | Query Builder | Results (10)

	monthwise	total_bookings
1	2020-01	26
2	2020-02	16
3	2020-03	16
4	2020-04	21
5	2020-05	21
6	2020-06	14
7	2020-07	20
8	2020-08	32
9	2020-09	21
10	2020-10	15

Task 11: Calculate the count of total iOS users

- Use the database **cab_booking**
- Use the table **clickstream_data**
- Count all the number of customers using iOS with “count(*)”. The condition for **os_version** is applied using “where” clause

6.43s Database cab_booking Type text ?

```
select count(*) AS TOTAL_USERS from clickstream_data where os_version = 'iOS';
```

Query execution screenshot:

```
INFO  : Map 1: 1/1      Reducer 2: 1/1
INFO  : Completed executing command(queryId=hive_20231024003227_86ab0e1f-49d6-4034-b400-b4a1ec594180); Time taken:
5.985 seconds
INFO  : OK
```

application_1698100961546_0010

Query results screenshot:

Query History | Saved Queries | Query Builder | Results (1)

	total_users
1	1515