



ДРУГИ ПРОЈЕКАТ DENVER MOBILITY



Flink



Електронски факултет у Нишу
Вештачка интелигенција и машинско учење

Професор: Др. Драган Х. Стојановић
Студент: Петковић Петар

САДРЖАЈ

01

Подешавање окружења

02

Flink апликација


















































02

Spark Апликација

04

Упис података у
Cassandra базу
података

Списак контејнера и docker compose

| | | | | | | | | | | |
|--------------------------|---|---------------------------------------|--|---------------|--|----------------|---------|---|---|---|
| <input type="checkbox"/> |  | big-data-systems-mobi | | Running (9/9) | | 33 seconds ago | 231.1% |  |  |  |
| <input type="checkbox"/> |  | cassandra-1 | cassandra:latest | Running | 9042:9042  | 35 seconds ago | 214.84% |  |  |  |
| <input type="checkbox"/> |  | jobmanager-1 | flink:1.16.0 | Running | 8081:8081  | 35 seconds ago | 7.03% |  |  |  |
| <input type="checkbox"/> |  | kafka-1 | wurstmeister/kafka:2.13-2.7.0 | Running | 9091:9091  | 33 seconds ago | 4.02% |  |  |  |
| <input type="checkbox"/> |  | producer-1 | big-data-systems-mobility-stre | Running | | 33 seconds ago | 0.74% |  |  |  |
| <input type="checkbox"/> |  | taskmanager-1 | flink:1.16.0 | Running | | 34 seconds ago | 3.92% |  |  |  |
| <input type="checkbox"/> |  | zookeeper-1 | wurstmeister/zookeeper:lates | Running | 2181:2181  | 35 seconds ago | 0.17% |  |  |  |
| <input type="checkbox"/> |  | spark-master | bde2020/spark-master:3.1.2-t | Running | 7077:7077  Show all ports (2 | 35 seconds ago | 0.12% |  |  |  |
| <input type="checkbox"/> |  | spark-worker-1 | bde2020/spark-worker:3.1.2-h | Running | 8071:8071  | 34 seconds ago | 0.13% |  |  |  |
| <input type="checkbox"/> |  | spark-worker-2 | bde2020/spark-worker:3.1.2-h | Running | 8072:8071  | 34 seconds ago | 0.13% |  |  |  |

```

1 version: "3.9"
2 networks:
3   bde:
4     external: true
5
6 services:
7   kafka:
8     image: wurstmeister/kafka:2.13-2.7.0
9     depends_on:
10      - zookeeper
11     ports:
12      - "9091:9091"
13     expose:
14      - "9092"
15     environment:
16       KAFKA_ADVERTISED_LISTENERS: INTERNAL://kafka:9092,EXTERNAL://localhost:9091
17       KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT,EXTERNAL:PLAINTEXT
18       KAFKA_LISTENERS: INTERNAL://0.0.0.0:9092,EXTERNAL://0.0.0.0:9091
19       KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
20       KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
21     volumes:
22      - /var/run/docker.sock:/var/run/docker.sock
23     networks:
24      - bde
25
26   zookeeper:
27     image: wurstmeister/zookeeper:latest
28     ports:
29      - "2181:2181"
30     networks:
31      - bde
32
33   producer:
34     build:
35       context: .
36       dockerfile: producer/Dockerfile
37     depends_on:
38      - kafka
39     environment:
40       SCRIPT: producer/producer.py
41       DATA: data/denverVehiclesCleaned.csv
42       KAFKA_HOST: kafka:9092
43       KAFKA_TOPIC: vehicles_topic
44       KAFKA_INTERVAL: 1
45     networks:
46      - bde
47

```

```

cassandra:
  image: cassandra:latest
  ports:
    - 9042:9042
  volumes:
    - ~/apps/cassandra:/var/lib/cassandra
  environment:
    - CASSANDRA_CLUSTER_NAME=cloudinfra
  networks:
    - bde

spark-master:
  image: bde2020/spark-master:3.1.2-hadoop3.2
  container_name: spark-master
  ports:
    - "8070:8070"
    - "7077:7077"
  environment:
    - INIT_DAEMON_STEP=setup_spark
    - SPARK_MASTER_PORT=7077
    - SPARK_MASTER_WEBUI_PORT=8070
  networks:
    - bde

spark-worker-1:
  image: bde2020/spark-worker:3.1.2-hadoop3.2
  container_name: spark-worker-1
  depends_on:
    - spark-master
  ports:
    - "8071:8071"
  environment:
    - "SPARK_MASTER=spark://spark-master:7077"
    - SPARK_WORKER_WEBUI_PORT=8071
  networks:
    - bde

spark-worker-2:
  image: bde2020/spark-worker:3.1.2-hadoop3.2
  container_name: spark-worker-2
  depends_on:
    - spark-master
  ports:
    - "8072:8071"

```






```





spark-worker-2:
  image: bde2020/spark-worker:3.1.2-hadoop3.2
  container_name: spark-worker-2
  depends_on:
    - spark-master
  ports:
    - "8072:8071"
  environment:
    - "SPARK_MASTER=spark://spark-master:7077"
    - SPARK_WORKER_WEBUI_PORT=8071
  networks:
    - bde






jobmanager:
  image: flink:1.16.0
  expose:
    - "6123"
  ports:
    - "8081:8081"
  command: jobmanager
  environment:
    - JOB_MANAGER_RPC_ADDRESS=jobmanager
  networks:
    - bde

taskmanager:
  image: flink:1.16.0
  expose:
    - "6121"
    - "6122"
  depends_on:
    - jobmanager
  command: taskmanager
  links:
    - "jobmanager:jobmanager"
  environment:
    - JOB_MANAGER_RPC_ADDRESS=jobmanager
  networks:
    - bde

```

| | | | |
|---|------------------|------------------|------|
|  data | 20.6.2023. 13:17 | File folder | |
|  flink_streaming | 26.9.2023. 00:54 | File folder | |
|  producer | 11.5.2023. 19:44 | File folder | |
|  spark_project | 23.9.2023. 20:06 | File folder | |
|  docker-compose | 15.9.2023. 18:19 | Yaml Source File | 3 KB |

| | | | |
|---|------------------|--------------------|------|
|  consumer_spark | 23.9.2023. 19:11 | Python Source File | 7 KB |
|  Dockerfile | 23.9.2023. 19:03 | File | 1 KB |
|  requirements | 6.9.2023. 18:36 | Text Document | 1 KB |
|  start-streaming | 23.9.2023. 19:53 | Windows Batch File | 1 KB |

| | | | |
|--|------------------|-----------------------|-------|
|  .idea | 25.9.2023. 15:20 | File folder | |
|  src | 20.6.2023. 13:03 | File folder | |
|  target | 26.9.2023. 00:54 | File folder | |
|  .gitignore | 20.6.2023. 13:02 | Git Ignore Source ... | 1 KB |
|  pom.xml | 15.9.2023. 18:18 | xmlfile | 10 KB |

ПРОЈЕКАТ



INICIJALIZACIJA SESIJE

```
1 usage  * Petar *
```

```
def initialize_spark_session():
```

```
    conf = SparkConf()
```

```
    conf.setMaster("spark://spark-master:7077")
```

```
    conf.set("spark.cassandra.connection.host", "cassandra")
```

```
    conf.set("spark.cassandra.connection.port", "9042")
```

```
    spark = SparkSession.builder.config(conf=conf).appName("Projekat2").getOrCreate()
```

```
    spark.sparkContext.setLogLevel("ERROR")
```

```
    return spark
```

PARSIRANJE VREDNOSTI

```
def parse_kafka_values(spark):
    data_frame = (
        spark.readStream.format("kafka")
        .option("kafka.bootstrap.servers", "kafka:9092")
        .option("subscribe", "vehicles_topic")
        .option("startingOffsets", "latest")
        .load()
    )

    schema = StructType(
        [
            StructField("latitude", StringType()),
            StructField("longitude", StringType()),
            StructField("speed_kmh", StringType()),
            StructField("id", StringType()),
            StructField("timestamp", StringType()),
            StructField("acceleration", StringType()),
            StructField("type", StringType()),
            StructField("distance", StringType()),
            StructField("odometer", StringType()),
            StructField("pos", StringType()),
        ]
    )

    parsed_values = data_frame.select(
        "timestamp", from_json(col("value").cast("string"), schema).alias("parsed_values")
    )
```

```
parsed_values = data_frame.select(
    "timestamp", from_json(col("value").cast("string"), schema).alias("parsed_values")
)

df_org = parsed_values.selectExpr(
    "timestamp",
    "parsed_values.latitude AS latitude",
    "parsed_values.longitude AS longitude",
    "parsed_values.speed_kmh AS speed_kmh",
    "parsed_values.id AS id",
    "parsed_values.acceleration AS acceleration",
    "parsed_values.type AS type",
    "parsed_values.distance AS distance",
    "parsed_values.odometer AS odometer",
    "parsed_values.pos AS pos",
)

df_org = df_org.withColumn("pos", col("pos").cast("double"))
df_org = df_org.withColumn("latitude", col("latitude").cast("double"))
df_org = df_org.withColumn("longitude", col("longitude").cast("double"))
df_org = df_org.filter(df_org.speed_kmh ≤ 120)

return df_org
```


RACUNANJE STATISTIKE

```
def calculate_statistics(df, long1=None, long2=None, lat1=None, lat2=None):
    if long1 is not None and long2 is not None and lat1 is not None and lat2 is not None:
        df_ret = df.where(
            (df.longitude < long1)
            & (df.longitude > long2)
            & (df.latitude < lat1)
            & (df.latitude > lat2)
        ).groupBy(window(df.timestamp, "10 seconds", "10 seconds")).agg(
            mean(df.speed_kmh).alias("mean_speed"),
            min(df.speed_kmh).alias("min_speed"),
            max(df.speed_kmh).alias("max_speed"),
            count(df.speed_kmh).alias("count_speed"),
        )
        return df_ret
    else:
        return None
```

```
def find_top_n_locations(df, N=5, num_decimal_places=3):
    df_rounded_locations = df.select(
        round("latitude", num_decimal_places).alias("latitude"),
        round("longitude", num_decimal_places).alias("longitude"),
    )

    df_with_window = df_rounded_locations.groupBy(
        window(df.timestamp, "10 seconds", "10 seconds"), "latitude", "longitude"
    ).agg(count("latitude").alias("freq"))

    top_n_locations = df_with_window.orderBy("freq", ascending=False).limit(N)
    return top_n_locations
```

UPIS U CASSANDRU

```
def write_statistics_to_cassandra(writeDF, _):  
    writeDF.write \  
        .format("org.apache.spark.sql.cassandra") \  
        .mode('append') \  
        .options(table="statistika", keyspace="locationsdb") \  
        .save()  
    writeDF.show()  
  
1 usage  * Petar *  
  
def write_top_n_locations_to_cassandra(writeDF, _):  
    writeDF.write \  
        .format("org.apache.spark.sql.cassandra") \  
        .mode('append') \  
        .options(table="top_locations", keyspace="locationsdb") \  
        .save()  
    writeDF.show()
```

GLAVNA FUNKCIJA

```
def execution(df):  
    long1, long2, lat1, lat2 = float(sys.argv[1]), float(sys.argv[2]), float(sys.argv[3]), float(sys.argv[4])  
  
    print("Statističke vrednosti za slučaj kada su prosleđene samo širina i dužina")  
    df_statistics = calculate_statistics(df, long1=long1, long2=long2, lat1=lat1, lat2=lat2)  
    df_statistics_cassandra = df_statistics.selectExpr(  
        "window.start as start",  
        "window.end as end",  
        "mean_speed",  
        "min_speed",  
        "max_speed",  
        "count_speed"  
    )  
  
    print("Pronalazak top N lokacija")  
    df_top_n_locations = find_top_n_locations(df)  
    df_top_n_locations_cassandra = df_top_n_locations.selectExpr(  
        "window.start as start",  
        "window.end as end",  
        "latitude",  
        "longitude",  
        "freq"  
    )  
  
    query1 = (df_statistics_cassandra.writeStream  
        .foreachBatch(write_statistics_to_cassandra)  
        .outputMode("update")  
        .start())  
    query2 = (df_top_n_locations_cassandra.writeStream  
        .foreachBatch(write_top_n_locations_to_cassandra)  
        .outputMode("complete")  
        .start())  
  
    query1.awaitTermination()  
    query2.awaitTermination()
```

MAIN

```
if __name__ == '__main__':
    num_arguments = len(sys.argv)
    if num_arguments < 2:
        print("Usage: main.py <input folder> ")
        exit(-1)
    elif num_arguments != 5:
        print("Invalid number of arguments")
        exit(-1)
    else:
        if all(is_float(sys.argv[i]) for i in range(1, 5)):
            df_spark = initialize_spark_session()
            dataframe = parse_kafka_values(df_spark)
            execution(dataframe)
        else:
            print("Invalid arguments")
            exit(-1)
```

SPARK STREAMING APP

```
FROM bde2020/spark-python-template:3.1.2-hadoop3.2

ENV CASSANDRA_HOST cassandra-node
ENV CASSANDRA_PORT 9042
ENV KAFKA_HOST=kafka:9092
ENV KAFKA_TOPIC=vehicles_topic
ENV KAFKA_CONSUMER_GROUP=Spark-Group
ENV SPARK_APPLICATION_PYTHON_LOCATION /app/consumer_spark.py
ENV SPARK_APPLICATION_ARGS "39.7545 -104.9733 39.7507 -104.9685"
ENV SPARK_SUBMIT_ARGS --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.1.2,com.datastax.spark:spark-cassandra-connector_2.12:3.2.0
```

```
#!/bin/bash
```



```
docker build --rm -t bde/spark-app .
```

```
docker run --name SparkStreaming --net bde -p 4040:4040 bde/spark-app
```

BATCH OBRADA U KONZOLI

| | | | | | | |
|---------------------|---------------------|--------------------|------------|-----------|-----------|-------------|
| | | | | | | |
| | start | end | mean_speed | min_speed | max_speed | count_speed |
| 2023-09-23 18:13:30 | 2023-09-23 18:13:40 | 43.275 | 39.89 | 46.66 | 2 | |
| 2023-09-23 18:13:00 | 2023-09-23 18:13:10 | 46.34999999999994 | 46.15 | 46.55 | 2 | |
| 2023-09-23 18:13:20 | 2023-09-23 18:13:30 | 45.4 | 45.4 | 45.4 | 1 | |
| 2023-09-23 18:13:50 | 2023-09-23 18:14:00 | 39.38 | 39.38 | 39.38 | 1 | |
| | | | | | | |
| | start | end | mean_speed | min_speed | max_speed | count_speed |
| 2023-09-23 18:14:10 | 2023-09-23 18:14:20 | 40.1 | 40.1 | 40.1 | 1 | |
| | | | | | | |
| | start | end | mean_speed | min_speed | max_speed | count_speed |
| 2023-09-23 18:14:20 | 2023-09-23 18:14:30 | 7.27 | 7.27 | 7.27 | 1 | |
| 2023-09-23 18:14:10 | 2023-09-23 18:14:20 | 30.576666666666668 | 17.39 | 40.1 | 3 | |
| | | | | | | |
| | start | end | mean_speed | min_speed | max_speed | count_speed |
| 2023-09-23 18:14:20 | 2023-09-23 18:14:30 | 43.629999999999995 | 7.27 | 79.99 | 2 | |

CASSANDRA TABELE

```
# cqlsh
Connected to cloudinfra at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.3 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> use locationsdb
... ;
cqlsh:locationsdb> select * from top_locations
... ;
```

| start | end | latitude | longitude | freq |
|---------------------------------|---------------------------------|----------|-----------|------|
| 2023-09-14 12:06:20.000000+0000 | 2023-09-14 12:06:30.000000+0000 | 39.73 | -104.973 | 2 |
| 2023-09-14 12:02:20.000000+0000 | 2023-09-14 12:02:30.000000+0000 | 39.735 | -105.026 | 1 |
| 2023-09-14 12:02:20.000000+0000 | 2023-09-14 12:02:30.000000+0000 | 39.751 | -104.976 | 2 |
| 2023-09-14 12:05:40.000000+0000 | 2023-09-14 12:05:50.000000+0000 | 39.73 | -104.973 | 2 |
| 2023-09-14 12:00:40.000000+0000 | 2023-09-14 12:00:50.000000+0000 | 39.727 | -104.962 | 1 |
| 2023-09-14 12:03:40.000000+0000 | 2023-09-14 12:03:50.000000+0000 | 39.73 | -104.973 | 2 |
| 2023-09-14 12:00:20.000000+0000 | 2023-09-14 12:00:30.000000+0000 | 39.748 | -105.024 | 1 |
| 2023-09-14 12:00:30.000000+0000 | 2023-09-14 12:00:40.000000+0000 | 39.737 | -104.983 | 1 |
| 2023-09-14 12:00:30.000000+0000 | 2023-09-14 12:00:40.000000+0000 | 39.751 | -104.96 | 1 |
| 2023-09-14 12:00:50.000000+0000 | 2023-09-14 12:01:00.000000+0000 | 39.751 | -104.976 | 2 |
| 2023-09-14 12:02:00.000000+0000 | 2023-09-14 12:02:10.000000+0000 | 39.73 | -104.973 | 2 |
| 2023-09-14 12:01:40.000000+0000 | 2023-09-14 12:01:50.000000+0000 | 39.727 | -105.025 | 1 |
| 2023-09-14 12:02:10.000000+0000 | 2023-09-14 12:02:20.000000+0000 | 39.737 | -104.998 | 1 |
| 2023-09-14 12:01:10.000000+0000 | 2023-09-14 12:01:20.000000+0000 | 39.731 | -105.025 | 1 |
| 2023-09-14 12:02:30.000000+0000 | 2023-09-14 12:02:40.000000+0000 | 39.73 | -104.973 | 2 |
| 2023-09-14 12:05:00.000000+0000 | 2023-09-14 12:05:10.000000+0000 | 39.73 | -104.973 | 2 |
| 2023-09-14 12:01:00.000000+0000 | 2023-09-14 12:01:10.000000+0000 | 39.732 | -104.971 | 1 |
| 2023-09-14 12:01:00.000000+0000 | 2023-09-14 12:01:10.000000+0000 | 39.748 | -104.98 | 1 |
| 2023-09-14 12:04:20.000000+0000 | 2023-09-14 12:04:30.000000+0000 | 39.73 | -104.973 | 2 |
| 2023-09-14 12:00:00.000000+0000 | 2023-09-14 12:00:10.000000+0000 | 39.729 | -104.973 | 1 |

```
(20 rows)
cqlsh:locationsdb> select * from statistika
... ;
```

| start | end | count_speed | max_speed | mean_speed | min_speed |
|---------------------------------|---------------------------------|-------------|-----------|------------|-----------|
| 2023-09-14 12:06:20.000000+0000 | 2023-09-14 12:06:30.000000+0000 | 6 | 5.54 | 24.095 | 0.0 |
| 2023-09-14 12:06:30.000000+0000 | 2023-09-14 12:06:40.000000+0000 | 2 | 61.81 | 50.365 | 38.92 |
| 2023-09-14 12:03:50.000000+0000 | 2023-09-14 12:04:00.000000+0000 | 2 | 37.4 | 34.36 | 31.32 |
| 2023-09-14 12:04:50.000000+0000 | 2023-09-14 12:05:00.000000+0000 | 2 | 17.89 | 8.945 | 0.0 |
| 2023-09-14 12:02:20.000000+0000 | 2023-09-14 12:02:30.000000+0000 | 1 | 21.1 | 21.1 | 21.1 |
| 2023-09-14 12:01:20.000000+0000 | 2023-09-14 12:01:30.000000+0000 | 2 | 7.2 | 22.895 | 38.59 |
| 2023-09-14 12:05:40.000000+0000 | 2023-09-14 12:05:50.000000+0000 | 6 | 53.78 | 35.02 | 0.14 |
| 2023-09-14 12:00:40.000000+0000 | 2023-09-14 12:00:50.000000+0000 | 2 | 53.86 | 45.415 | 36.97 |
| 2023-09-14 12:03:10.000000+0000 | 2023-09-14 12:03:20.000000+0000 | 4 | 52.2 | 37.2525 | 24.66 |
| 2023-09-14 12:03:40.000000+0000 | 2023-09-14 12:03:50.000000+0000 | 5 | 49.21 | 34.726 | 21.71 |
| 2023-09-14 12:02:50.000000+0000 | 2023-09-14 12:03:00.000000+0000 | 1 | 27.97 | 27.97 | 27.97 |
| 2023-09-14 12:05:30.000000+0000 | 2023-09-14 12:05:40.000000+0000 | 2 | 3.24 | 11.79 | 20.34 |
| 2023-09-14 12:00:30.000000+0000 | 2023-09-14 12:00:40.000000+0000 | 5 | 48.74 | 33.998 | 20.74 |
| 2023-09-14 12:06:10.000000+0000 | 2023-09-14 12:06:20.000000+0000 | 1 | 0.0 | 0 | 0.0 |
| 2023-09-14 12:05:10.000000+0000 | 2023-09-14 12:05:20.000000+0000 | 1 | 40.21 | 40.21 | 40.21 |
| 2023-09-14 12:00:50.000000+0000 | 2023-09-14 12:01:00.000000+0000 | 1 | 0.0 | 0 | 0.0 |
| 2023-09-14 12:02:00.000000+0000 | 2023-09-14 12:02:10.000000+0000 | 5 | 50.29 | 35.028 | 18.36 |
| 2023-09-14 12:01:30.000000+0000 | 2023-09-14 12:01:40.000000+0000 | 6 | 48.02 | 31.16833 | 15.37 |
| 2023-09-14 12:04:30.000000+0000 | 2023-09-14 12:04:40.000000+0000 | 2 | 38.45 | 37.64 | 36.83 |
| 2023-09-14 12:01:40.000000+0000 | 2023-09-14 12:01:50.000000+0000 | 1 | 13.36 | 13.36 | 13.36 |

ПРОЈЕКАТ



Flink

DATA STREAM JOB

```
public class DataStreamJob {

    1 usage new *
    private static void validateInput(Double lat1, Double lat2, Double long1, Double long2) {...}
    1 usage new *
    public static DataStream<Vehicle> ConvertJsonToVehicle(DataStream<String> jsonStream) {...}
    1 usage new *
    public static DataStream<String> StreamConsumer(String inputTopic, String server, StreamExecutionEnvironment environment) throws Exception {...}
    1 usage new *
    public static FlinkKafkaConsumer<String> createStringConsumerForTopic(String topic, String kafkaAddress) {...}
    1 usage new *
    private static SingleOutputStreamOperator processAverageAggregate(DataStream<Vehicle> vehicleStream, Double lat1, Double lat2, Double long1, Double long2) {...}
    1 usage new *
    private static SingleOutputStreamOperator processTopNLocations(DataStream<Vehicle> vehicleStream) {...}
    1 usage new *
    private static void sinkToDatabase(SingleOutputStreamOperator averageAggregateStream, SingleOutputStreamOperator topNLocationsStream) throws Exception {...}
    1 usage
    private static final String KAFKA_TOPIC = "vehicles_topic";
    1 usage
    private static final String KAFKA_SERVER = "kafka:9092";
```

DATA STREAM JOB

```
public static void main(String[] args) {

    Double lat1 = Double.parseDouble(args[0]);
    Double lat2 = Double.parseDouble(args[1]);
    Double long1 = Double.parseDouble(args[2]);
    Double long2 = Double.parseDouble(args[3]);
    validateInput(lat1, lat2, long1, long2);

    // Konfiguracija okoline

    StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
    DataStream<String> dataStream = StreamConsumer(KAFKA_TOPIC, KAFKA_SERVER, env);
    DataStream<Vehicle> vehicleStream = ConvertJsonToVehicle(dataStream);

    // Sliding window operacije
    SingleOutputStreamOperator averageAggregateStream = processAverageAggregate(vehicleStream, lat1, lat2, long1, long2);
    averageAggregateStream.print();
    SingleOutputStreamOperator topNLocationsStream = processTopNLocations(vehicleStream);
    topNLocationsStream.print();

    // Upis u bazu
    sinkToDatabase(averageAggregateStream, topNLocationsStream);

    // Execute program, beginning computation.
    env.execute( jobName: "Flink Java API Skeleton");
}
```

AVERAGE AGGREGATE

```
1 usage  ± Petar *
public class AverageAggregate implements AggregateFunction<Vehicle, Tuple5<String, Double, Double, Double, Integer>,
    Tuple5<String, Double, Double, Double, Integer>> {

no usages  ± Petar

@Override
public Tuple5<String, Double, Double, Double, Integer> createAccumulator() {
    return new Tuple5<>("", 0.0, Double.MAX_VALUE, 0.0, 0);
}

± Petar *
@Override
public Tuple5<String, Double, Double, Double, Integer> add(Vehicle value, Tuple5<String, Double, Double, Double, Integer> accumulator) {
    String id = value.getId();
    Double speed = value.getSpeed_kmh();
    Double minSpeed = Math.min(value.getSpeed_kmh(), accumulator.f2);
    Double maxSpeed = Math.max(value.getSpeed_kmh(), accumulator.f3);
    Integer count = accumulator.f4 + 1;

    return new Tuple5<>(id, accumulator.f1 + speed, minSpeed, maxSpeed, count);
}

± Petar *
@Override
public Tuple5<String, Double, Double, Double, Integer> getResult(Tuple5<String, Double, Double, Double, Integer> acc) {
    return new Tuple5<>(acc.f0, acc.f2, acc.f3, calculateAverage(acc.f1, acc.f4), acc.f4);
}

± Petar *
@Override
public Tuple5<String, Double, Double, Double, Integer> merge(Tuple5<String, Double, Double, Double, Integer> acc1,
    Tuple5<String, Double, Double, Double, Integer> acc2) {
    return new Tuple5<>(acc1.f0, acc1.f1 + acc2.f1,
        Math.min(acc1.f2, acc2.f2), Math.max(acc1.f3, acc2.f3), acc1.f4 + acc2.f4);
}

// Metoda za računanje proseka
1 usage  new *
private double calculateAverage(double sum, int count) {
    return count == 0 ? 0 : sum / count;
}
```

TOP N LOCATIONS

```
1 usage  ± Petar *  
  
public class TopNLocationsAggregate implements AggregateFunction<Vehicle, Map<String, Integer>, Tuple1<String>> {  
    2 usages  
    private final int n;  
    1 usage  ± Petar  
    public TopNLocationsAggregate(int n) { this.n = n; }  
    no usages  ± Petar *  
    @Override  
    public Map<String, Integer> createAccumulator() { return new HashMap<>(); }  
    ± Petar *  
    @Override  
    public Map<String, Integer> add(Vehicle vehicle, Map<String, Integer> accumulator) {  
        double roundedLatitude = roundToTwoDecimals(vehicle.getLatitude());  
        double roundedLongitude = roundToTwoDecimals(vehicle.getLongitude());  
  
        String key = roundedLatitude + " " + roundedLongitude;  
        accumulator.merge(key, value: 1, Integer::sum);  
        return accumulator;  
    }  
    new *
```

TOP N LOCATIONS

```
@Override
public Tuple1<String> getResult(Map<String, Integer> accumulator) {
    List<Tuple2<String, Integer>> sortedLocations = accumulator.entrySet().stream() Stream<Entry<...>>
        .map(entry → new Tuple2<>(entry.getKey(), entry.getValue())) Stream<Tuple2<...>>
        .sorted((e1, e2) → e2.f1.compareTo(e1.f1))
        .collect(Collectors.toList());

    int size = Math.min(n, sortedLocations.size());
    List<Tuple2<String, Integer>> topLocations = sortedLocations.subList(0, size);

    String output = topLocations.toString();
    return new Tuple1<>(output);
}
```

⌵ Petar *

```
@Override
public Map<String, Integer> merge(Map<String, Integer> a, Map<String, Integer> b) {
    b.forEach((key, value) → a.merge(key, value, Integer::sum));
    return a;
}
```

2 usages new *

```
private double roundToTwoDecimals(Double value) {...}
```

CASSANDRA SERVICE

3 usages 1 Petar *

```
public final class CassandraService {
```

no usages

```
private static final Logger logger = Logger.getLogger(CassandraService.class);
```

1 usage 1 Petar *

```
public static void sinkToDatabase15(final DataStream<Tuple5<String, Double, Double, Double, Integer>> sinkFilteredStream) throws Exception {
```

```
    sinkFilteredStream.print();
```

```
    CassandraSink
```

```
        .addSink(sinkFilteredStream)
```

```
        .setHost("cassandra")
```

```
        .setQuery("INSERT INTO flink_keyspace.statistika(id, min_speed, max_speed, mean_speed, count) " +
```

```
            "values (?, ?, ?, ?, ?);")
```

```
        .build();
```

```
}
```

1 usage 1 Petar *

```
public static void sinkToDatabase30(final DataStream<Tuple1<String>> sinkFilteredStream) throws Exception {
```

```
    sinkFilteredStream.print();
```

```
    CassandraSink
```

```
        .addSink(sinkFilteredStream)
```

```
        .setHost("cassandra")
```

```
        .setQuery("INSERT INTO flink_keyspace.top_locations(latitude_longitude_count) values (?);")
```

```
        .build();
```

```
}
```

```
}
```

POJO

14 usages [⌵](#) Petar *

@Builder

@Getter

@Setter

@ToString

@EqualsAndHashCode

@AllArgsConstructor

@NoArgsConstructor

```
public class Vehicle {  
    private String timestamp;  
    private String id;  
    private String type;  
    private double latitude;  
    private double longitude;  
    private double speed_kmh;  
    private double acceleration;  
    private double distance;  
    private double odometer;  
    private double pos;  
}
```


TASK MANAGER LOGS

```
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -105.0,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -105.0,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.75 -105.0,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.75 -105.0,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -104.98,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -104.98,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.75 -105.0,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.75 -105.0,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.74 -104.97,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.74 -104.97,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -105.02,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -105.02,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -104.99,1]])
big-data-systems-mobility-streaming-analytics-taskmanager-1 | ([[39.73 -104.99,1]])
|
```

```
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus36,16.63,16.63,16.63,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus36,16.63,16.63,16.63,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus35,20.05,20.05,20.05,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus35,20.05,20.05,20.05,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus34,0.0,0.0,0.0,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus34,0.0,0.0,0.0,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus33,29.81,29.81,29.81,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus33,29.81,29.81,29.81,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus32,30.85,30.85,30.85,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus32,30.85,30.85,30.85,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus31,34.81,34.81,34.81,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus31,34.81,34.81,34.81,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus30,41.69,41.69,41.69,1)
big-data-systems-mobility-streaming-analytics-taskmanager-1 | (bus30,41.69,41.69,41.69,1)
```

UPIS U CASSANDRU

```
cqlsh:flink_keyspace> select * from top_locations;
```

latitude_longitude_count

| |
|------------------------|
| [(39.75 -184.97, 1),] |
| [(39.75 -184.98, 1),] |
| [(39.75 -184.99, 1),] |
| [(39.74 -184.96, 1),] |
| [(39.74 -185.02, 1),] |
| [(39.74 -185.0, 1),] |
| [(39.73 -185.01, 1),] |
| [(39.75 -184.96, 1),] |
| [(39.73 -184.98, 1),] |
| [(39.75 -185.0, 1),] |
| [(39.74 -184.99, 1),] |
| [(39.74 -184.97, 1),] |
| [(39.74 -185.01, 1),] |
| [(39.73 -184.97, 1),] |
| [(39.74 -184.98, 1),] |
| [(39.73 -184.99, 1),] |
| [(39.73 -185.03, 1),] |
| [(39.73 -185.02, 1),] |
| [(39.73 -185.0, 1),] |

```
cqlsh:flink_keyspace> select * from statistika;
```

| id | count | max_speed | mean_speed | min_speed |
|--------|-------|-----------|------------|-----------|
| bus35 | 1 | 0 | 0 | 0 |
| bus26 | 1 | 8.42 | 8.42 | 8.42 |
| bus34 | 1 | 3.13 | 3.13 | 3.13 |
| bus29 | 1 | 25.06 | 25.06 | 25.06 |
| bus30 | 1 | 21.89 | 21.89 | 21.89 |
| bus25 | 1 | 40.36 | 40.36 | 40.36 |
| bus9 | 1 | 46.8 | 46.8 | 46.8 |
| bus20 | 1 | 15.62 | 15.62 | 15.62 |
| veh101 | 1 | 11.34 | 11.34 | 11.34 |
| veh103 | 1 | 6.12 | 6.12 | 6.12 |
| veh10 | 1 | 50.44 | 50.44 | 50.44 |
| bus7 | 1 | 47.88 | 47.88 | 47.88 |
| bus21 | 1 | 49.9 | 49.9 | 49.9 |
| bus24 | 1 | 49.46 | 49.46 | 49.46 |
| bus32 | 1 | 13.21 | 13.21 | 13.21 |
| bus5 | 1 | 42.19 | 42.19 | 42.19 |
| bus28 | 1 | 26.17 | 26.17 | 26.17 |
| veh0 | 1 | 74.7 | 74.7 | 74.7 |
| bus3 | 1 | 5.54 | 5.54 | 5.54 |
| bus31 | 1 | 16.85 | 16.85 | 16.85 |
| bus22 | 1 | 52.2 | 52.2 | 52.2 |
| bus6 | 1 | 0.14 | 0.14 | 0.14 |
| bus33 | 1 | 9.68 | 9.68 | 9.68 |
| bus4 | 1 | 11.63 | 11.63 | 11.63 |
| veh1 | 1 | 50.8 | 50.8 | 50.8 |
| bus27 | 1 | 27.79 | 27.79 | 27.79 |
| bus23 | 1 | 54.07 | 54.07 | 54.07 |
| veh102 | 1 | 8.35 | 8.35 | 8.35 |
| bus8 | 1 | 31.07 | 31.07 | 31.07 |
| veh100 | 1 | 16.24 | 16.24 | 16.24 |

SNAPSHOT

```
[INFO] --- install:3.1.1:install (default-install) @ flink-java ---
[INFO] Installing C:\Fakultet\big-data-systems-mobility-streaming-analytics\flink_streaming\pom.xml to C:\Users\petko\.m2\repository\Petar\flink-java\1.0-SNAPSHOT\flink-java-1.0-SNAPSHOT.pom
[INFO] Installing C:\Fakultet\big-data-systems-mobility-streaming-analytics\flink_streaming\target\flink-java-1.0-SNAPSHOT.jar to C:\Users\petko\.m2\repository\Petar\flink-java\1.0-SNAPSHOT\flink-java-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.800 s
[INFO] Finished at: 2023-09-26T03:02:21+02:00
[INFO] -----
```

FLINK UI

Overview

Jobs

Running Jobs

Completed Jobs

Task Managers

Job Manager

Submit New Job

Flink Java API Skeleton

Cancel Job

| | | | | | |
|------------|----------------------------------|-----------|----------------------|---------|---------------------------------|
| Job ID | e7911fc0d89e32c8ba39b6c8285bac20 | Job State | <div>RUNNING</div> 3 | Actions | Job Manager Log |
| Start Time | 2023-09-26 02:36:21 | Duration | 6m 18s | | |

OverviewExceptionsTimeLineCheckpointsConfiguration

Source: Custom Source -> Map -> Map -> Filter
Parallelism: 1
Backpressure (max): 0%
Busy (max): 100%

map

SlidingProcessingTimeWindows -> (Sink: Print to Std. Out, Sink: Print to Std. Out, Sink: Cassandra Sink)
Parallelism: 1
Backpressure (max): 0%
Busy (max): 0%

filter

SlidingProcessingTimeWindows -> (Sink: Print to Std. Out, Sink: Print to Std. Out, Sink: Cassandra Sink)
Parallelism: 1
Backpressure (max): 0%
Busy (max): 0%

| Name | Status | Bytes Received | Records Received | Bytes Sent | Records Sent | Parallelism | Start Time | Duration | Tasks |
|--|--------------------|----------------|------------------|------------|--------------|-------------|---------------------|----------|--------------|
| Source: Custom Source -> Map -> Map -> Filter | <div>RUNNING</div> | 0 B | 0 | 64.0 KB | 364 | 1 | 2023-09-26 02:36:21 | 6m 17s | <div>1</div> |
| SlidingProcessingTimeWindows -> (Sink: Print to Std. Out, Sink: Print to Std. Out, Sink: Cassandra Sink) | <div>RUNNING</div> | 42.6 KB | 364 | 0 B | 0 | 1 | 2023-09-26 02:36:21 | 6m 17s | <div>1</div> |
| SlidingProcessingTimeWindows -> (Sink: Print to Std. Out, Sink: Print to Std. Out, Sink: Cassandra Sink) | <div>RUNNING</div> | 42.6 KB | 364 | 0 B | 0 | 1 | 2023-09-26 02:36:21 | 6m 17s | <div>1</div> |



ХВАЛА НА
ПАЖЊИ