Departamento de Electrónica, Sistemas e Informática

Computer Systems Engineering

Big Data Processing

Autumn 2025

Final Project: Batch Processing

Date: October, 2025

Student Name: Axel Gallardo

Professor: Pablo Camarillo Ramirez

In [1]:
```
import findspark
findspark.init()

from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("Structured Streaming - Profiles (Kafka)") \
    .master("spark://spark-master:7077") \
    .config("spark.jars.packages", "org.apache.spark:spark-sql-kafka-0-10_2.13:4.0.0") \
    .config("spark.ui.port", "4040") \
    .getOrCreate()

sc = spark.sparkContext
sc.setLogLevel("INFO")

spark.conf.set("spark.sql.shuffle.partitions", "5")
```

```
WARNING: Using incubator modules: jdk.incubator.vector
:: loading settings :: url = jar:file:/opt/spark/jars/ivy-2.5.3.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /root/.ivy2.5.2/cache
The jars for the packages stored in: /root/.ivy2.5.2/jars
org.apache.spark#spark-sql-kafka-0-10_2.13 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-608bf916-80c3-4f0d-9da2-1ab90cde7b5f;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.13;4.0.0 in central
        found org.apache.spark#spark-token-provider-kafka-0-10_2.13;4.0.0 in central
        found org.apache.kafka#kafka-clients;3.9.0 in central
        found org.lz4#lz4-java;1.8.0 in central
        found org.xerial.snappy#snappy-java;1.1.10.7 in central
        found org.slf4j#slf4j-api;2.0.16 in central
        found org.apache.hadoop#hadoop-client-runtime;3.4.1 in central
        found org.apache.hadoop#hadoop-client-api;3.4.1 in central
        found com.google.code.findbugs#jsr305;3.0.0 in central
        found org.scala-lang.modules#scala-parallel-collections_2.13;1.2.0 in central
        found org.apache.commons#commons-pool2;2.12.0 in central
:: resolution report :: resolve 704ms :: artifacts dl 33ms
        :: modules in use:
        com.google.code.findbugs#jsr305;3.0.0 from central in [default]
        org.apache.commons#commons-pool2;2.12.0 from central in [default]
        org.apache.hadoop#hadoop-client-api;3.4.1 from central in [default]
        org.apache.hadoop#hadoop-client-runtime;3.4.1 from central in [default]
        org.apache.kafka#kafka-clients;3.9.0 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.13;4.0.0 from central in [default]
        org.apache.spark#spark-token-provider-kafka-0-10_2.13;4.0.0 from central in [default]
        org.lz4#lz4-java;1.8.0 from central in [default]
        org.scala-lang.modules#scala-parallel-collections_2.13;1.2.0 from central in [default]
        org.slf4j#slf4j-api;2.0.16 from central in [default]
        org.xerial.snappy#snappy-java;1.1.10.7 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   11  |   0   |   0   |   0   ||   11  |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-608bf916-80c3-4f0d-9da2-1ab90cde7b5f
        confs: [default]
        0 artifacts copied, 11 already retrieved (0kB/18ms)
25/11/16 00:31:45 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
```

In [2]:
```
kafka_df = spark.readStream \
            .format("kafka") \
            .option("kafka.bootstrap.servers", "kafka:9093") \
            .option("subscribe", "project-stream") \
            .option("startingOffsets", "earliest") \
            .option("failOnDataLoss", "false") \
            .load()
```

In [3]:
```
from pyspark.sql.functions import from_json, col
from pyspark.sql.types import StructField, StringType
from pcamarillor.spark_utils import SparkUtils

value_ts_df = kafka_df.select(
    kafka_df.value.cast("string").alias("value_str"),
    kafka_df.timestamp.alias("kafka_ts")
)

schema_columns = [
    ("id", "int"),
    ("first_name", "string"),
    ("last_name", "string"),
    ("email", "string"),
    ("gender", "string"),
    ("age", "int"),
    ("education", "string"),
]
profiles_schema = SparkUtils.generate_schema(schema_columns)

profiles_extracted_df = value_ts_df.withColumn(
    "profile", from_json(col("value_str"), profiles_schema)
).filter(col("profile").isNotNull())
```

In [4]:
```
from pyspark.sql.functions import (
    year, month, day,
    trim, lower, when, lit, concat_ws, col
)

profiles_flat_stream = (
    profiles_extracted_df
      .select(
          col("kafka_ts"),
          col("profile.id").alias("id"),
          col("profile.first_name").alias("first_name"),
          col("profile.last_name").alias("last_name"),
          col("profile.email").alias("email"),
          col("profile.gender").alias("gender"),
          col("profile.age").alias("age"),
          col("profile.education").alias("education"),
      )
)

profiles_clean_stream = (
    profiles_flat_stream
      .dropDuplicates(["id"])
      .withColumn("first_name",  trim(col("first_name")))
      .withColumn("last_name",   trim(col("last_name")))
      .withColumn("email",       lower(trim(col("email"))))
      .withColumn("gender",      trim(col("gender")))
      .withColumn("education",   trim(col("education")))
      .filter(col("age").isNotNull())
      .filter((col("age") >= 18) & (col("age") <= 65))
)

profiles_clean_v2_stream = profiles_clean_stream.dropna(
    subset=[c[0] for c in schema_columns if c[0] != "id"]
)

profiles_t1_stream = (
    profiles_clean_v2_stream
      .withColumn("full_name", concat_ws(" ", col("first_name"), col("last_name")))
      .withColumn(
          "education_numeric",
          when(lower(col("education")) == "no schooling", lit(0))
          .when(lower(col("education")) == "primary", lit(1))
          .when(lower(col("education")) == "middle school", lit(2))
          .when(lower(col("education")) == "high school", lit(3))
          .when(lower(col("education")) == "university", lit(4))
          .otherwise(lit(None))
      )
)

profiles_time_df = (
    profiles_t1_stream
      .withColumn("year",  year(col("kafka_ts")))
      .withColumn("month", month(col("kafka_ts")))
      .withColumn("day",   day(col("kafka_ts")))
)
```

In [5]:
```
!rm -rf /opt/spark/work-dir/data/profiles_output/
!rm -rf /opt/spark/work-dir/profiles_checkpoint/

!mkdir -p /opt/spark/work-dir/data/profiles_output/
!pwd
```

```
/opt/spark/work-dir/final_project/streaming_processing
```

In [6]:
```
query_files = profiles_time_df.select(
                    "id",
                    "first_name",
                    "last_name",
                    "email",
                    "gender",
                    "age",
                    "education",
                    "full_name",
                    "education_numeric",
                    "year",
                    "month",
                    "day"
                ).writeStream \
                 .trigger(processingTime="10 seconds") \
                 .partitionBy("education") \
                 .format("parquet") \
                 .option("header", "true") \
                 .option("path", "/opt/spark/work-dir/data/profiles_output/") \
                 .option("checkpointLocation", "/opt/spark/work-dir/profiles_checkpoint") \
                 .outputMode("append") \
                 .start()
```

```
25/11/16 00:32:01 WARN ResolveWriteToStream: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
```

In [7]:
```
!python3 /opt/spark/work-dir/lib/axelgallardo/producer_profiles.py
```

```
25/11/16 00:32:22 WARN ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 10000} milliseconds, but spent 20375 milliseconds
[Stage 9:==================================>                      (3 + 1) / 5]
Envio completado a topic 'project-stream'.
```

In [10]:
```
query_files.stop()
sc.stop()
```

```
25/11/16 00:33:30 WARN DAGScheduler: Failed to cancel job group 76b62ab0-f266-4103-aa60-ee8c58f8cc3d. Cannot find active jobs for it.
25/11/16 00:33:31 WARN DAGScheduler: Failed to cancel job group 76b62ab0-f266-4103-aa60-ee8c58f8cc3d. Cannot find active jobs for it.
```

In [ ]:
```

```