

Sprawozdanie

Zestaw danych: Netflix-Prize-Data

Technologia: Apache Spark Structured Streaming

Producent; skrypty inicjujące i zasilający

Uruchamiamy cluster dataproc i przechodzimy do konsoli węzła master

```
gcloud dataproc clusters create ${CLUSTER_NAME} \
--enable-component-gateway --bucket ${BUCKET_NAME} --region ${REGION} --
--master-machine-type n1-standard-4 --master-boot-disk-size 50 \
--num-workers 2 --worker-machine-type n1-standard-2 --worker-boot-disk-
--image-version 2.1-debian11 --optional-components DOCKER,ZOOKEEPER \
--project ${PROJECT_ID} --max-age=2h \
--metadata "run-on-master=true" \
--initialization-actions \
gs://goog-dataproc-initialization-actions-${REGION}/kafka/kafka.sh
```

Pobieramy repozytorium ze skryptami i uruchamiamy skrypt inicjujący środowisko.
Alternatywnie możemy rozpakować archiwum z kodem źródłowym.

```
git clone https://github.com/MichalxPZ/Spark-Structured-Streaming---Net  
mv Spark-Structured-Streaming---Netflix-Prize-Data/* .  
rm -rf Spark-Structured-Streaming---Netflix-Prize-Data
```

Inicjalizujemy zmienne środowiskowe - możliwość edycji w pliku env.sh

Pobieramy dane, budujemy kody źródłowe oraz uruchamiamy kontener z postgresem.

Inicjalizujemy również schemat bazy danych i nadajemy odpowiednie uprawnienia.

```
source ./setup.sh
```

Polecenie uruchamiające producenta danych

```
source ./kafka_producer.sh
```

Konsument: skrypt odczytujący wyniki przetwarzania

Aplikacja posiada dwóch konsumentów, którzy odczytują dane z bazy danych lub tematu Kafki.

Domyślnie konsument nasłuchujący na temat kafki zbiera dane dotyczące anomalii przetwarzania.

```
source ./kafka_consumer.sh
```

```
2002-09-20 00:00:00,2002-09-21 00:00:00,K-Pax,36,3.2777777777777777
2002-05-27 00:00:00,2002-05-28 00:00:00,Independence Day,27,3.925925925925926
2002-06-27 00:00:00,2002-06-28 00:00:00,Braveheart,24,4.416666666666667
2002-07-27 00:00:00,2002-07-28 00:00:00,Casino: 10th Anniversary Edition,25,3.68
2002-05-17 00:00:00,2002-05-18 00:00:00,Memento,24,3.5416666666666665
2002-04-20 00:00:00,2002-04-21 00:00:00,Big,73,3.6986301369863015
2002-03-31 00:00:00,2002-04-01 00:00:00,Coming to America,89,3.168539325842697
2002-03-30 00:00:00,2002-03-31 00:00:00,Sleepless in Seattle,101,3.485148514851485
2002-05-05 00:00:00,2002-05-06 00:00:00,Jay and Silent Bob Strike Back,23,3.391304347826087
2002-03-30 00:00:00,2002-03-31 00:00:00,Swingers,20,3.4
2002-04-17 00:00:00,2002-04-18 00:00:00,The Waterboy,23,3.217391304347826
2002-08-26 00:00:00,2002-08-27 00:00:00,Double Jeopardy,31,3.3548387096774195
2002-04-30 00:00:00,2002-05-01 00:00:00,Traffic,41,3.682926829268293
2002-09-03 00:00:00,2002-09-04 00:00:00,Father of the Bride,47,3.2127659574468086
2002-04-05 00:00:00,2002-04-06 00:00:00,Cast Away,30,3.6333333333333333
2002-04-10 00:00:00,2002-04-11 00:00:00,Die Hard With a Vengeance,94,3.425531914893617
2002-07-20 00:00:00,2002-07-21 00:00:00,The Abyss,21,3.1904761904761907
2002-05-11 00:00:00,2002-05-12 00:00:00,The Score,22,3.5454545454545454
2002-05-31 00:00:00,2002-06-01 00:00:00,Nine to Five,21,3.0952380952380953
2002-09-20 00:00:00,2002-09-21 00:00:00,Big,97,3.752577319587629
2002-08-31 00:00:00,2002-09-01 00:00:00,Pearl Harbor,33,3.5454545454545454
2002-07-19 00:00:00,2002-07-20 00:00:00,The Matrix,65,4.292307692307692
2002-05-14 00:00:00,2002-05-15 00:00:00,What Women Want,41,3.2195121951219514
2002-06-26 00:00:00,2002-06-27 00:00:00,Forrest Gump,56,4.25
2002-07-13 00:00:00,2002-07-14 00:00:00,Rat Race,28,3.2857142857142856
2002-03-26 00:00:00,2002-03-27 00:00:00,Forever Young,22,3.5454545454545454
2002-05-06 00:00:00,2002-05-07 00:00:00,Gremlins,33,3.5454545454545454
2002-09-12 00:00:00,2002-09-13 00:00:00,Predator: Collector's Edition,24,3.7083333333333333
2002-06-07 00:00:00,2002-06-08 00:00:00,Notting Hill,20,3.85
2002-06-26 00:00:00,2002-06-27 00:00:00,Remember the Titans,43,3.9767441860465116
2002-06-01 00:00:00,2002-06-02 00:00:00,Fight Club,25,3.84
2002-08-21 00:00:00,2002-08-22 00:00:00,Kate & Leopold,27,3.4074074074074074
2002-05-22 00:00:00,2002-05-23 00:00:00,Legends of the Fall,46,3.347826086956522
2002-05-22 00:00:00,2002-05-23 00:00:00,The Man Who Wasn't There,29,3.103448275862069
2002-09-07 00:00:00,2002-09-08 00:00:00,Ferris Bueller's Day Off,99,3.8282828282828283
2002-06-28 00:00:00,2002-06-29 00:00:00,This Is Spinal Tap,21,4.0476190476190474
2002-05-13 00:00:00,2002-05-14 00:00:00,Memento,23,4.0
2002-04-14 00:00:00,2002-04-15 00:00:00,Father of the Bride,28,3.4285714285714284
2002-07-30 00:00:00,2002-07-31 00:00:00,Fifteen Minutes,25,3.32
2002-09-13 00:00:00,2002-09-14 00:00:00,Ocean's Eleven,38,3.789473684210526
2002-08-09 00:00:00,2002-08-10 00:00:00,Dr. Seuss' How the Grinch Stole Christmas,36,4.0277777777777778
2002-08-27 00:00:00,2002-08-28 00:00:00,Citizen Kane,22,4.090909090909091
2002-06-10 00:00:00,2002-06-11 00:00:00,Remember the Titans,23,4.0
2002-03-12 00:00:00,2002-03-13 00:00:00,The Manchurian Candidate,23,3.9130434782608696
2002-09-10 00:00:00,2002-09-11 00:00:00,An Officer and a Gentleman,23,3.608695652173913
2002-09-11 00:00:00,2002-09-12 00:00:00,K-Pax,28,3.607142857142857
2002-08-02 00:00:00,2002-08-03 00:00:00,The Perfect Storm,37,3.5675675675675675
2002-03-12 00:00:00,2002-03-13 00:00:00,The Perfect Storm,39,3.358974358974359
2002-09-25 00:00:00,2002-09-26 00:00:00,The Wizard of Oz: Collector's Edition,83,4.228915662650603
2002-08-29 00:00:00,2002-08-30 00:00:00,Miss Congeniality,48,3.25
2002-08-18 00:00:00,2002-08-19 00:00:00,The Time Machine,23,3.217391304347826
2002-08-05 00:00:00,2002-08-06 00:00:00,A Walk in the Clouds,25,3.0
2002-06-05 00:00:00,2002-06-06 00:00:00,Sexy Beast,20,3.25
2002-09-03 00:00:00,2002-09-04 00:00:00,The Shipping News,25,3.36
2002-06-14 00:00:00,2002-06-15 00:00:00,Spy Game,29,3.793103448275862
```

Konsument typu jdbc odczytuje dane zagregowane.

```
source ./jdbc_consumer.sh
```

```

===== NEW DATA =====
1970-JANUARY-12 - 1970-FEBRUARY-11 Creepy Crawlers(12725) 13 35 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Friend(9428) 3 14 2
1970-JANUARY-12 - 1970-FEBRUARY-11 Town Without Pity(439) 22 69 4
1970-JANUARY-12 - 1970-FEBRUARY-11 The Pigkeeper's Daughter / Sassy Sue(12029) 4 6 2
1970-JANUARY-12 - 1970-FEBRUARY-11 G.I. Blues(14788) 15 48 4
1970-JANUARY-12 - 1970-FEBRUARY-11 The Fighting Seabees(3701) 36 121 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Buffalo '66(2149) 62 207 5
1970-JANUARY-12 - 1970-FEBRUARY-11 The Killing of a Chinese Bookie(5371) 5 15 3
1970-JANUARY-12 - 1970-FEBRUARY-11 Steamboat Bill(15193) 6 23 4
1970-JANUARY-12 - 1970-FEBRUARY-11 The Adventures of Huck Finn(9323) 20 65 3
1970-JANUARY-12 - 1970-FEBRUARY-11 The Last Castle(14407) 605 2185 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Star Wars: Episode II: Attack of the Clones(8687) 110 455 4
1970-JANUARY-12 - 1970-FEBRUARY-11 Samurai Trilogy 2: Duel at Ichijoji Temple(16431) 14 51 3
1970-JANUARY-12 - 1970-FEBRUARY-11 Malcolm X(17319) 48 203 4
1970-JANUARY-12 - 1970-FEBRUARY-11 A Paradise Under the Stars(15227) 5 9 3
1970-JANUARY-12 - 1970-FEBRUARY-11 Poirot: Death in the Clouds(16103) 32 107 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Comedy Only in Da Hood(2681) 2 8 1
1970-JANUARY-12 - 1970-FEBRUARY-11 Alfred Hitchcock: Sabotage and The Lodger(11465) 9 25 2
1970-JANUARY-12 - 1970-FEBRUARY-11 K-9(2072) 5 17 3
1970-JANUARY-12 - 1970-FEBRUARY-11 Silsila(6088) 2 8 1
1970-JANUARY-12 - 1970-FEBRUARY-11 Blue Planet: IMAX(588) 29 93 4
1970-JANUARY-12 - 1970-FEBRUARY-11 Innerspace(4089) 776 2431 5
1970-JANUARY-12 - 1970-FEBRUARY-11 The Specialist(12741) 280 726 5
1970-JANUARY-12 - 1970-FEBRUARY-11 A Lesson Before Dying(1694) 10 32 2
1970-JANUARY-12 - 1970-FEBRUARY-11 Entrapment(6692) 1303 4243 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Phenomenon(7399) 640 2170 5
1970-JANUARY-12 - 1970-FEBRUARY-11 The Adventures of Buckaroo Banzai(12490) 200 651 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Panic Room(16390) 1946 6895 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Surviving the Game(12274) 40 116 5
1970-JANUARY-12 - 1970-FEBRUARY-11 The Cook(10860) 82 246 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Relax ... It's Just Sex(13086) 51 140 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Grave of the Fireflies(14210) 66 255 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Iris(15361) 696 2386 5
1970-JANUARY-12 - 1970-FEBRUARY-11 The Graduate(15922) 815 3173 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Vanished(12705) 5 9 2
1970-JANUARY-12 - 1970-FEBRUARY-11 That Old Feeling(17649) 6 16 2
1970-JANUARY-12 - 1970-FEBRUARY-11 Essence of Echoes(6288) 5 8 2
1970-JANUARY-12 - 1970-FEBRUARY-11 Chopper Chicks in Zombietown(7452) 4 10 2
1970-JANUARY-12 - 1970-FEBRUARY-11 Ravenous(3481) 13 35 4
1970-JANUARY-12 - 1970-FEBRUARY-11 Carnal Crimes(4654) 6 8 2
1970-JANUARY-12 - 1970-FEBRUARY-11 The Cosmic Man(7075) 5 14 3
1970-JANUARY-12 - 1970-FEBRUARY-11 The Sentinel(2641) 4 4 1
1970-JANUARY-12 - 1970-FEBRUARY-11 Substitute 4: Failure is Not an Option(403) 3 8 2
1970-JANUARY-12 - 1970-FEBRUARY-11 Mission Kashmir(495) 22 64 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Holiday in the Sun(2080) 29 105 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Nothing to Lose(2135) 128 417 5
1970-JANUARY-12 - 1970-FEBRUARY-11 The Pianist(6044) 9 22 4
1970-JANUARY-12 - 1970-FEBRUARY-11 Homeward Bound: The Incredible Journey(9757) 144 463 5
1970-JANUARY-12 - 1970-FEBRUARY-11 Jim Breuer: Heavy Metal Comedy(1225) 6 22 2
1970-JANUARY-12 - 1970-FEBRUARY-11 Cosmic Voyage: IMAX(3932) 20 59 5

```

Utrzymanie obrazu czasu rzeczywistego – transformacje

Aby uruchomić przetwarzanie strumieniowe, należy uruchomić skrypt

```
source ./processing_engine.sh
```

[illegible]

Ogólna idea przetwarzania danych

1. Pobieranie danych z Kafki

Dane są pobierane z tematu Kafki (`kafka_topic`), które są w formacie CSV z czterema kolumnami: `timestamp`, `movie_id`, `user_id` i `rating`. Te dane są przetwarzane strumieniowo przy użyciu Spark Structured Streaming. Strumień danych jest odczytywany za pomocą `SparkSession`, a wartości są parsowane i przekształcane na odpowiednie typy danych.

2. Agregowanie danych

Grupowanie i agregacja:

Dane są grupowane według 30-dniowego okna czasowego i movie_id. Agregaty obejmują liczbę ocen (rating_count), sumę ocen (rating_sum) oraz liczbę unikalnych ocen (unique_rating_count). Zapis do bazy danych:

Wyniki agregacji są zapisywane do bazy danych PostgreSQL. W zależności od trybu przetwarzania (A dla trybu "append" i C dla trybu "complete").

3. Wykrywanie anomalii

Watermark i grupowanie:

Dane są przetwarzane z 1-dniowym watermarkiem, co pozwala na tolerowanie opóźnień w strumieniu danych. Dane są grupowane według okna czasowego (zdefiniowanego przez `sliding_window_size_days`) i `movie_id`.

Wykrywanie anomalii:

Wyliczane są średnie oceny (`ratingMean`) i liczba ocen (`ratingCount`) dla każdego okna czasowego i `movie_id`. Anomalie są wykrywane, jeśli liczba ocen i średnia ocena przekraczają odpowiednie progi (`anomaly_rating_count_threshold` i `anomaly_rating_mean_threshold`).

Zapis anomalii do Kafki:

Zidentyfikowane anomalie są wysyłane z powrotem do tematu Kafki (`kafka_anomaly_topic`).

Utrzymanie obrazu czasu rzeczywistego – obsługa trybów

W zależności od parametru ustawionego w pliku `env.sh`, aplikacja działa w jednym z dwóch trybów: A lub C.

1. W pierwszym (`delay=A`) program ma dostarczać dane do obrazu czasu rzeczywistego z najmniejszym możliwym opóźnieniem, nawet jeśli

dostarczane wyniki nie są ostateczne i będzie trzeba je wielokrotnie aktualizować.

2. W drugim (delay=C) program ma dostarczać dane do obrazu czasu rzeczywistego najszybciej jak się da, ale tylko wyniki ostateczne, tak aby nie było potrzeby ich późniejszej aktualizacji.

```
def real_time_processing(ratingsDF, movies, jdbc_url, jdbc_user, jdbc_password, processing_mode):
    aggregatedRatingsDF = ratingsDF
    if processing_mode == "C":
        aggregatedRatingsDF = ratingsDF.withWatermark("timestamp", "1 days")

    aggregatedRatingsDF = aggregatedRatingsDF.groupBy(window("timestamp", "30 days"), "movie_id") \
        .agg(
            count("r").alias("rating_count"),
            sum("rating").alias("rating_sum"),
            approx_count_distinct("rating").alias("unique_rating_count")
        ) \
        .join(movies, aggregatedRatingsDF.movie_id == movies["_c0"], "inner") \
        .select(unix_timestamp(col("window.start")).alias("window_start"), "movie_id", col("_c2").alias("title"), "rating_count", "rating_sum", "unique_rating_count") \

    jdbcProperties = {
        "user": jdbc_user,
        "password": jdbc_password,
        "driver": "org.postgresql.Driver"
    }

    if processing_mode == "A":
        aggregatedRatingsQuery = aggregatedRatingsDF \
            .writeStream \
            .outputMode("update") \
            .foreachBatch(lambda batchDF, batchId: save_to_jdbc(batchDF, jdbc_url, jdbcProperties)) \
            .option("checkpointLocation", "/tmp/checkpoints/aggregatedRatings") \
            .trigger(processingTime="20 seconds") \
            .start()
    if processing_mode == "C":
        aggregatedRatingsQuery = aggregatedRatingsDF \
            .writeStream \
            .outputMode("append") \
            .foreachBatch(lambda batchDF, batchId: save_to_jdbc(batchDF, jdbc_url, jdbcProperties)) \
            .option("checkpointLocation", "/tmp/checkpoints/aggregatedRatings") \
            .trigger(processingTime="20 second") \
            .start()
```

Tryb A - anomalie nie występują (należy zmienić wartość w pliku env.sh)

```
export ANOMALY_PERIOD_LENGTH=30
export ANOMALY_RATING_COUNT=70
export ANOMALY_RATING_MEAN=4
```

Tryb C - anomalie występują stosunkowo często (należy zmienić wartość w pliku env.sh)

```
export ANOMALY_PERIOD_LENGTH=1
export ANOMALY_RATING_COUNT=2
export ANOMALY_RATING_MEAN=2
```

Wykrywanie anomalii

Aplikacja wykrywa anomalie w danych, jeśli liczba ocen i średnia ocena przekraczają odpowiednie progi.

Progi te są zdefiniowane w pliku env.sh.

```
def anomalies(ratingsDF, sliding_window_size_days, anomaly_rating_count_threshold, anomaly_rating_mean_threshold, kafka_bootstrap_servers, kafka_anomaly_topic, movies, groupId): 1usage  2Michał
watermarkedRatingsDF = ratingsDF.withWatermark("timestamp", "1 days")
anomaliesDF = watermarkedRatingsDF \
    .groupBy(window("timestamp", f"#{sliding_window_size_days} days", "1 day"), "movie_id") \
    .agg(
        count("*").alias("ratingCount"),
        mean("rating").alias("ratingMean")
    ) \
    .filter(f"ratingCount >= {anomaly_rating_count_threshold} AND ratingMean >= {anomaly_rating_mean_threshold}")

anomalies_joined = anomaliesDF.join(movies, anomaliesDF.movie_id == movies["_c0"], "inner") \
    .select(col("window_end").alias("window_end"), col("window_start").alias("window_start"), "movie_id", col("_c2").alias("title"), col("_c1").alias("Year"), "ratingCount", "ratingMean")

anomalies_formatted = anomalies_joined.select(concat(
    col("window_start"),
    lit(", "),
    col("window_end"),
    lit(", "),
    col("title"),
    lit(", "),
    col("ratingCount"),
    lit(", "),
    col("ratingMean"),
).alias("value"))

host_name = socket.gethostname()
anomaliesQuery = anomalies_formatted \
    .writeStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", f"{host_name}:9092") \
    .option("topic", kafka_anomaly_topic) \
    .option("checkpointLocation", "/tmp/checkpoints/anomalies") \
    .start()
```

Program przetwarzający strumienie danych; skrypt uruchamiający

Aby uruchomić przetwarzanie strumieniowe, należy uruchomić skrypt


```
source ./processing_engine.sh
```

```
source ./env.sh
```

```
$SPARK_HOME/bin/spark-submit \  
  --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.3.0,org.postgresql:postgresql:42.6.0 \  
  processing/spark-structured.py \  
  "$INPUT_FILE_PATH" \  
  "$KAFKA_BOOTSTRAP_SERVERS" \  
  "$KAFKA_DATA_TOPIC_NAME" \  
  "$KAFKA_GROUP_ID" \  
  "$JDBC_URL" \  
  "$JDBC_USERNAME" \  
  "$JDBC_PASSWORD" \  
  "$ANOMALY_PERIOD_LENGTH" \  
  "$ANOMALY_RATING_COUNT" \  
  "$ANOMALY_RATING_MEAN" \  
  "$KAFKA_ANOMALY_TOPIC_NAME" \  
  "$PROCESSING_TYPE"
```

Miejsce utrzymywania obrazów czasu rzeczywistego –

skrypt tworzący

Miejsce utrzymywania obrazów czasu rzeczywistego jest tworzone w skrypcie setup.sh

Jest to kontener z bazą PostgreSQL, który jest uruchamiany w klastrze dataproc.

Schematy bazy danych oraz tabele są tworzone w skrypcie setup.sql

```
DROP DATABASE IF EXISTS :db_name";
CREATE DATABASE :db_name WITH ENCODING 'UTF8';
DROP USER IF EXISTS :user";
CREATE USER :user WITH PASSWORD :password';
ALTER DATABASE :db_name OWNER TO :user";
ALTER SCHEMA public owner to :user";
GRANT ALL PRIVILEGES ON DATABASE :db_name TO :user";
GRANT USAGE ON SCHEMA public TO :user";
GRANT CREATE ON SCHEMA public TO :user";
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO :user";

\c :db_name";
DROP TABLE IF EXISTS movie_ratings;
CREATE TABLE IF NOT EXISTS movie_ratings (
    window_start BIGINT NOT NULL,
    movie_id VARCHAR(32) NOT NULL,
    title VARCHAR(128) NOT NULL,
    rating_count INTEGER NOT NULL,
    rating_sum INTEGER NOT NULL,
    unique_rating_count INTEGER NOT NULL,
    PRIMARY KEY (window_start, movie_id)
);
GRANT ALL PRIVILEGES ON TABLE movie_ratings TO :user";
ALTER TABLE movie_ratings OWNER TO :user";
```

Miejsce utrzymywania obrazów czasu rzeczywistego – cechy

PostgreSQL jest idealny do przechowywania zagregowanych danych w procesach Big Data dzięki skalowalności, która umożliwia obsługę dużych wolumenów danych, oraz zaawansowanym funkcjom analitycznym i agregacyjnym. Dzięki wsparciu dla równoległego przetwarzania zapytań i optymalizacji wydajności, PostgreSQL efektywnie zarządza skomplikowanymi analizami na dużych zbiorach danych. Dodatkowo, łatwa integracja z narzędziami Big Data i elastyczność w obsłudze różnych typów danych czynią go wszechstronnym rozwiązaniem.

Czyszczenie środowiska

Aby wyczyścić środowisko, należy uruchomić skrypt

```
source ./cleanup.sh
```