Spark – *Structured Streaming cz.2*

Ten warsztat jest kontynuacją poprzedniego. Jednak tym razem nie będzie to tutorial. Czas na twórczość własną. To ona jest główną wartością dodaną w tym zestawie.

Na produkcje

1. W przypadku Scali utwórz za pomocą IDE wersję ostateczną Twojej aplikacji (plik jar).
   W przypadku Pythona utwórz skrypt który będzie pełnowymiarową ostateczną wersją Twojej aplikacji
   Uwaga! Dokonaj parametryzacji swojej aplikacji tak, aby można było określać nazwę tematu źródłowego oraz nazwę docelowej tabeli

2. Utwórz nowy temat Kafki o nazwie `my-kafka-input` oraz tabelę w bazie danych PostgreSQL `my_housestats`.

3. Uruchom wszystkie elementy układanki (producenta danych, Twoją aplikację).
   Sprawdź zawartość docelowej tabeli.

Obsługa awarii

4. Uruchom nadawanie na dłuższy czas (np. 5 minut).

```
java -cp /usr/lib/kafka/libs/*:datafaker-1.4.0.jar:KafkaFakerProducer.jar \
      KafkaFakerProducer ${CLUSTER_NAME}-w-0:9092 kafka-input json 2 300
```

   Ponownie uruchom Twoją aplikację. Przerwij po jakimś czasie jej działanie (np. po 1 minucie). Sprawdź zawartość docelowej tabeli.

5. Odczekaj jakiś czas (np. 1 minutę), a następnie uruchom aplikację ponownie. Czy zapamiętała ona stan i kontynuuje obliczenia? A może wylicza wszystko od nowa?

6. Jeśli Twoja aplikacja nie zachowuje swojego stanu podczas przetwarzania i nie wykorzystuje go w przypadku awarii, dokonaj korekty swojej aplikacji, aby była w stanie poprawnie obsługiwać awarie. Jeśli z jakiegoś powodu już teraz to robi, dokonaj analizy tego z jakiej przyczyny tak się dzieje. Koniecznie sprawdź swoje rozwiązanie.

Obsługa zdarzeń opóźnionych

7. Czy w obecnej wersji aplikacji problem zdarzeń opóźnionych lub spóźnionych istnieje?

8. Zmień kod aplikacji oraz strukturę tabeli tak, aby obliczane do tej pory statystyki odnosiły się do kolejnych minut. W tabeli wystarczy że dodasz kolumnę `start_time`, która będzie początkiem minuty, dla której składowane są obliczenia. Uruchom aplikację i spraw jej działanie.

9. Czy teraz problem zdarzeń opóźnionych czy spóźnionych istnieje? Rozwiąż go uwzględniając źródłową wersję naszego generatora danych.

Monitorowanie działania

10. Ponownie uruchom nadawanie na dłuższy czas (np. 10 minut). Uruchom swoją aplikację.

11. Korzystając z metod obiektu `StreamingQuery` zdobądź informację na temat ostatniego mikrobatcha dotyczącą:
    a. liczby krotek pobranych z wejścia
    b. liczby krotek zapisanych na wyjście
    c. liczby krotek przetworzonych w ciągu sekundy
    d. liczby bajtów wykorzystywanej do zapamiętania stanu przetwarzania

12. Przejdź do interfejsu sieciowego naszej aplikacji Sparka (skorzystaj z interfejsu sieciowego *ResourceManagera YARN*). Znajdź wykres pokazujący jak zmieniał się czas wykonania każdego mikrobatcha.

To kończy nasze zadania. Jeśli udało Ci się wykonać je wszystkie… duże brawa!