Kafka, Stream Processing – projekt

Ogólny opis projektu

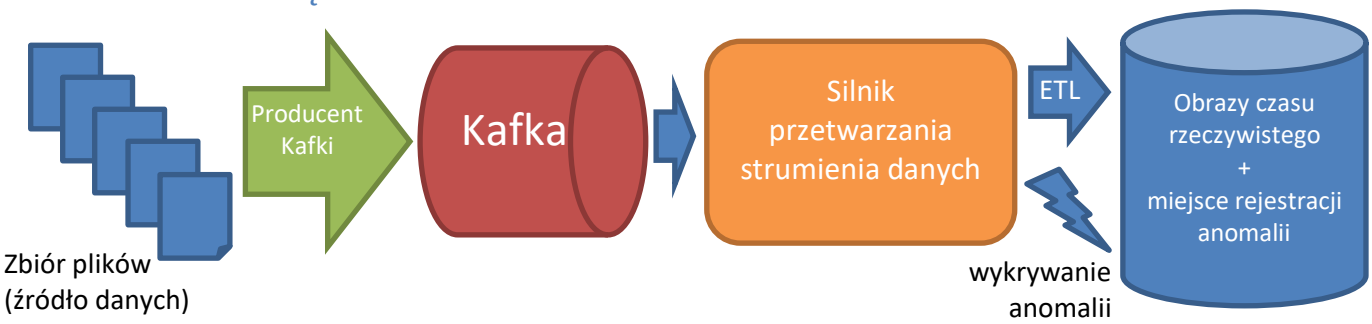
W ramach projektu należy samodzielnie zaimplementować rozwiązanie dokonujące przetwarzania strumieni danych w oparciu o:

- brokera wiadomości Kafka oraz
- określony silnik przetwarzania strumieni danych wykorzystywany w środowiskach Big Data, a także
- wybrane miejsce docelowe.

Dostępne silniki (*poziomy API*) przetwarzania strumieni danych:

- Spark Structured Streaming
- Kafka Streaming
- Flink (*DataStream API*)

Architektura rozwiązania



Opis

Dane źródłowe w naszym rozwiązaniu będą miały postać zbioru plików (do 100) dostępnych w jednym z katalogów.

Producent Kafki (zaimplementowany w ramach jednego z zestawów zadań) będzie odczytywał zawartość kolejnych plików z tego zbioru i wysyłał je, linia po linii, do brokera Kafki symulując w ten sposób zachodzenie zdarzeń w świecie rzeczywistym.

Twoim zadaniem będzie implementacja rozwiązania, które będzie:

- odczytywało dane z serwera Kafki
- utrzymywało na podstawie tych danych wyniki pewnych obliczeń (agregacji) – obraz czasu rzeczywistego
- reagowało na zachodzące "anomalie" rejestrując ich wystąpienia

Ponadto konieczne będzie wybranie oraz wykorzystanie właściwego (ze względu na własności) miejsca przechowywania obrazów czasu rzeczywistego oraz miejsca rejestracji anomalii. W obu przypadkach może to być to samo narzędzie/miejsce.
Uwzględnij fakt, że na platformie *Dataproc* dostępna środowisko Docker, co to daje praktycznie nieograniczone możliwości. Niestety nie każda z platform przetwarzania strumieni danych posiada złącza (*connector*) do każdego możliwego miejsca docelowego. Ważnym też są własności złącza.

Aplikacja ma działać w dwóch trybach w zależności od parametru programu `delay`.

- 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.
- 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.

Ostateczna wersja programu ma mieć postać pliku jar.

Oprócz samego programu należy dostarczyć

1. Skrypt tworzący źródłowe tematy Kafki i resetujący środowisko
   Resetowanie środowiska powinno umożliwiać ponownie uruchomienie programu "od nowa".
   Należy zatem zadbać o usunięcie i ponowne utworzenie tematów Kafki, miejsc składowania punktów kontrolnych oraz danych wykorzystywanych do wznawiania aplikacji *Kafka Streams* po awarii.
2. Skrypt i ewentualnie program zasilający źródłowe tematy Kafki
3. Program przetwarzania strumieni danych utrzymujący obraz czasu rzeczywistego oraz wyliczający anomalie
4. Skrypt uruchamiający program przetwarzania strumieni danych
5. Skrypt przygotowujący miejsce docelowe dla obrazu czasu rzeczywistego oraz dostarczania alertów po wykryciu anomalii
6. Skrypt odczytujący wyniki z miejsca docelowego
7. Sprawozdanie

Zbiory danych

Wszystkie zbiory danych pobieramy ze strony
`http://www.cs.put.poznan.pl/kjankiewicz/bigdata/stream_project` niezależnie od ich oryginalnego źródła pochodzenia.

Kilka wskazówek

1. Nie ładuj wejściowych danych bezpośrednio na klaster. Załaduj dane jeden raz na zasobnik (*bucket*), a następnie, za każdym razem kiedy będzie taka potrzeba, kopiuj je z zasobnika na klaster (`hadoop fs -copyToLocal gs://`) ewentualnie przetwarzaj bezpośrednio z zasobnika
2. Nie twórz rozwiązań bezpośrednio na GCP. Postaraj się w miarę możliwości tworzyć Twoje rozwiązania lokalnie. Oszczędzaj zasoby.
3. Nie uruchamiaj początkowych wersji programów na pełnym zbiorze danych. Postaraj się sprawdzić swoje rozwiązania na próbce danych, najlepiej tak przygotowanej aby znać oczekiwane wyniki i móc je porównać. Dopiero kiedy Twój program będzie gotowy, przetestuj go na pełnym wolumenie danych.
4. Twórz kolejne wersje programów zwiększając stopniowo używane w nim konstrukcje. Rozpocznij od przepisywania danych z tematu wynikowego do ujścia. Jeśli to działa, wprowadzaj kolejno poszczególne transformacje cały czas mając wszystko pod kontrolą.
5. Rozpocznij tworzenie Twojego rozwiązania od zasilania wejściowego tematu "z konsoli", mając pod kontrolą dostarczanie każdej wiadomości, obserwując po każdej z nich to co dostajesz na wyjściu.

Punktacja projektu

Kryterium	Poziom 0 – 0%	Poziom 1 – 75%	Poziom 2 – 100%	Liczba punktów
Producent; skrypty inicjujące i zasilający	Brak, lub fundamentalne błędy uniemożliwiające działanie	Drobne błędy uniemożliwiające działanie, lub działanie jest możliwe ale niepoprawne	Ideał, spójny z resztą projektu	4
Utrzymanie obrazu czasu rzeczywistego – transformacje	Brak lub brak spójności z tematem projektu lub fundamentalne błędy uniemożliwiające działanie	Drobne błędy uniemożliwiające działanie, lub działanie jest możliwe ale niepoprawne	Ideał, spójny z tematem i resztą projektu	8
Utrzymanie obrazu czasu rzeczywistego – obsługa trybu A	Brak lub fundamentalne błędy	Drobne błędy	Ideał	4
Utrzymanie obrazu czasu rzeczywistego – obsługa trybu C	Brak lub fundamentalne błędy	Drobne błędy	Ideał	4
Wykrywanie anomalii	Brak lub brak spójności z tematem projektu lub fundamentalne błędy uniemożliwiające działanie	Drobne błędy uniemożliwiające działanie, lub działanie jest możliwe ale niepoprawne	Ideał, spójny z tematem i resztą projektu	8
Program przetw. strumienie danych; jar	Brak	Istnieje, występują problemy z jego użyciem	Ideał, spójny z resztą projektu	2
Program przetw. strumienie danych; skrypt uruchamiający	Brak	Istnieje, występują problemy z jego użyciem	Ideał, spójny z resztą projektu	2
Miejsce utrzymywania obrazów czasu rzeczywistego – skrypt tworzący	Brak lub fundamentalne błędy uniemożliwiające działanie	Istnieje, występują problemy z jego użyciem	Ideał, spójny z resztą projektu	2
Miejsce utrzymywania obrazów czasu rzeczywistego – cechy	Brak użycia	Istnieje, występują problemy z jego użyciem i/lub jego cechy nie są adekwatne	Ideał, spójny z resztą projektu	4
Konsument: skrypt odczytujący wyniki przetwarzania	Brak lub fundamentalne błędy uniemożliwiające działanie	Drobne błędy uniemożliwiające działanie, lub działanie jest możliwe ale niepoprawne	Ideał, spójny z resztą projektu	2
			Razem	40

Sprawozdanie

Celem sprawozdania jest ułatwienie oceny projektu. W związku z tym sprawozdanie powinno się składać z jednoznacznie wyodrębnionych i czytelnych punktów odnoszących się do kryteriów oceny.

Poniżej wymienione zostały kryteria – punkty sprawozdania – oraz ich sugerowana treść.

Producent; skrypty inicjujące i zasilający

- Jeśli potrzeba jest uruchomienia klastra w specyficzny sposób (zakładamy, że domyślny klaster oparty jest na obrazie 2.1-debian11 i oprócz domyślnych komponentów Hadoop, Spark, ma dodane: środowisko Docker, Apache Kafka i Flinka) to tu powinna się pojawić instrukcja jego uruchomienia
- Jeśli przed uruchomieniem skryptów należy wykonać dodatkowe działania, to tu należy podać ich kod i wyjaśnienie
- Wskazanie skryptu tworzącego źródłowe tematy Kafki i resetującego środowisko oraz sposób jego uruchomienia.
- Wskazanie będącego skryptem lub skryptami zasilającymi tematy Kafki i sposób uruchomienia
- Dobrze jest dodać zrzut ekranu pokazujący działanie tych skryptów

Utrzymanie obrazu czasu rzeczywistego – transformacje

- Fragmentu kodu programu odpowiadającego za wyliczenia obrazu czasu rzeczywistego od poziomu źródła
- Komentarze ułatwiające zrozumienie kodu oraz intencje autora

Utrzymanie obrazu czasu rzeczywistego – obsługa trybu A

- Fragmenty kodu obsługujące tryb A – najmniejsze możliwe opóźnienia, z aktualizacjami wyników obrazu czasu rzeczywistego
- Komentarze ułatwiające zrozumienie kodu oraz intencje autora

Utrzymanie obrazu czasu rzeczywistego – obsługa trybu C

- Fragmenty kodu obsługujące tryb C – najmniejsze możliwe opóźnienia przy uwzględnieniu braku aktualizacji wyników obrazu czasu rzeczywistego
- Komentarze ułatwiające zrozumienie kodu oraz intencje autora

Wykrywanie anomalii

- Fragmentu kodu programu odpowiadającego za wyliczenia anomalii od poziomu źródła
- Komentarze ułatwiające zrozumienie kodu oraz intencje autora

Program przetwarzający strumienie danych; skrypt uruchamiający

- Wskazanie skryptu uruchamiającego program przetwarzający strumień danych z opisem sposobu jego wykorzystania a także opisem jego parametrów.
- Podane dwa warianty uruchomienia programu
    - wariant 1
        - tryb obsługi obrazu czasu rzeczywistego A,
        - parametry wykrywania anomalii, które powodują, że anomalie nie występują
    - wariant 2
        - tryb obsługi obrazu czasu rzeczywistego C,
        - parametry wykrywania anomalii, które powodują, że anomalie występują stosunkowo często
- Można załączyć zrzuty ekranu przedstawiające sposoby uruchomienia i uzyskiwanie wyniki

Miejsce utrzymywania obrazów czasu rzeczywistego – skrypt tworzący

- Wskazanie skryptu tworzącego miejsce utrzymywania obrazu czasu rzeczywistego
- Można załączyć zrzuty ekranu przedstawiające procedurę tworzenia miejsca przechowywania obrazu czasu rzeczywistego

Miejsce utrzymywania obrazów czasu rzeczywistego – cechy

- Wyjaśnienia dotyczące powodów wyboru takiego, a nie innego miejsca utrzymywania obrazu czasu rzeczywistego uwzględniające charakter wymagań narzucanych przez systemy przetwarzania strumieni danych

Konsument: skrypt odczytujący wyniki przetwarzania

- Wskazanie skryptu pozwalającego na inspekcję zawartości wyników przetwarzania (zarówno obrazu czasu rzeczywistego jak i wyników wykrywania anomalii
- Można załączyć zrzuty ekranu przedstawiające uruchomienie i przykładowe wyniki