Politechnika Warszawska
Wydział Elektryczny

Raport z projektu

Autorzy:
Aleksei Haidukevich, nr albumu 295233
Jakub Korczakowski, nr albumu 291079
Marharyta Kruk, nr albumu 295235
Maciej Leszczyński, nr albumu 291085
Piotr Rosa, nr albumu 291112

5 marca 2020

Spis treści

1 Etap 1 - Zbieranie danych

czas: 4.11.2019 - 15.12.2019

1.1 Opis

W pierwszym etapie projektu najważniejszym celem było utworzenie narzędzi do zbierania danych z Twittera i Reddita. Zostaną one użyte w celu stworzenia zbioru danych pozwalającego na naukę i testowanie algorytmów. Pobrane wiadomości i nagłówki będą składowane w bazie danych. Ten etap prac zawierał również przygotowanie infrastruktury zdolnej analizować i przechowywać zebrane dane.

1.2 Zadania

1. **Analiza dostępności API mediów społecznościowych.**
   Po przeanalizowaniu API zdecydowaliśmy się wykorzystać media społecznościowe **Twitter** i **Facebook**. API Twittera jest bardzo rozwinięte i pozwala na swobodny dostęp do tweetów, nawet w przypadku darmowej wersji konta. Zdecydowaliśmy się pobierać tweety wykorzystując bibliotekę Pythona - **Tweepy**. W przypadku Facebooka korzystamy z programu **FacePager**. Jest to program pozwalający pobierać wszystkie posty z wybranej strony i zapisać je do pliku z rozszerzeniem csv. Opcjonalna jest dodatkowa konwersja do JSON. Posty można pobierać z całego okresu istnienia strony.

2. **Analiza dostępnej infrasturktury do przetwarzania danych.**
   W docelowej aplikacji przetwarzającej dane w czasie rzeczywistym planujemy użyć:

   Apache Hadoop do pobierania tweetów w czasie rzeczywistym,

   Apache Spark do przetwarzania danych i uruchamiania modeli,

   MS SQL do przechowywania tweetów.

   Elementy architektury aplikacji prawdopodobnie ulegną jeszcze zmianie podczas dalszego rozwoju projektu.

3. **Analiza i wybór sposobu wdrożenia aplikacji (rozwiązania chmurowe).**
   Po analizie potrzeb naszego projektu zdecydowaliśmy się wykorzystać rozwiązanie chmurowe, ponieważ pozwala one na zapewnienie dostępu do zasobów projektowych(takich jak bazy danych czy maszyny wirtualne) dla wszystkich pracujących nad projektem.

   Wśród dostawców infrastruktury chmurowej można wyróżnić dwie firmy, które oferują darmowe środki dla studentów. Są to Amazon (AWS)

oraz Microsoft (Azure). Z powodu mniejszych możliwości oraz mniejszej ilości środków dostępnych na patformie AWS zdecydowaliśmy się wybrać platformę Azure.

W obrębie tej platformy oprócz klasycznych maszyn wirtualnych dostępne są rozwiązania docelowe przeznaczone do przetwarzania dużych ilości danych, należą do nich:

- HDInsight,
- Azure Databricks,
- Azure Data Lake Services.

Pomimo tego, że te usługi znacznie ułatwają budowę projektów nie zdecydowaliśmy się na ich użycie ze względu na wysoką cenę. Planujemy używać maszyn wirtualnych i za pomocą Dockera, a w przyszłości Kubertenesa zbudować infrastrukturę projektu.

4. **Stworzenie programu pobierającego dane z Twittera.**
Korzystając z biblioteki **Tweepy** udało nam się stworzyć skrypt pobierający wpisy z Twittera dotyczące wybranego słowa kluczowego. API Twittera pozwala na pobieranie naprawdę wielu szczegółów dotyczących wpisów, jednak do naszych celów nie potrzebujemy ich wszystkich. Zdecydowaliśmy się na pobieranie: ID tweeta, datę jego stworzenia, nazwę użytkownika oraz zawartość tweeta.
Pobrane dane przechowujemy w tabelach, osobno dla każdej firmy, z dodatkową kolumną Sentyment, która mówi o tym, czy przesłanie wiadomości jest negatywne, czy pozytywne.
Skrypt ten pobiera dane w interwałach czasowych, przy czym nie wszystkie tweety są zapisywane, a losowana jest jedynie część z nich. Wynika to z faktu, że często pojawia się więcej wpisów niż się spodziewaliśmy, a możemy nie poradzić sobie ze zbyt dużym zbiorem danych.

5. **Stworzenie programu pobierającego dane z Facebooka.**
Pobieranie danych z Facebooka przy użyciu programu FacePager pozwala na dostęp do danych niezależnie od dat powstania postów. Dzięki temu, uruchamiając program co określony czas dla wybranych stron, otrzymujemy pliki csv, z których dane przetwarzane są przy użyciu języka Python. Interesujące nas dane są analogiczne do tych w przypadku Twittera.

6. **Analiza dostępnych zbiorów opisanych klasami, pozwalających na testowanie algorytmu.**
Znaleziony został zbiór pozwalający na trenowanie oraz testowanie algorytmu. Zbiór zawiera 1600000 opisanych tweetów. Dla każdego tweetu został obliczony sentyment.

   Zbiór znajduje się pod adresem: adres.

2 Etap 2 - Przetwarzanie danych

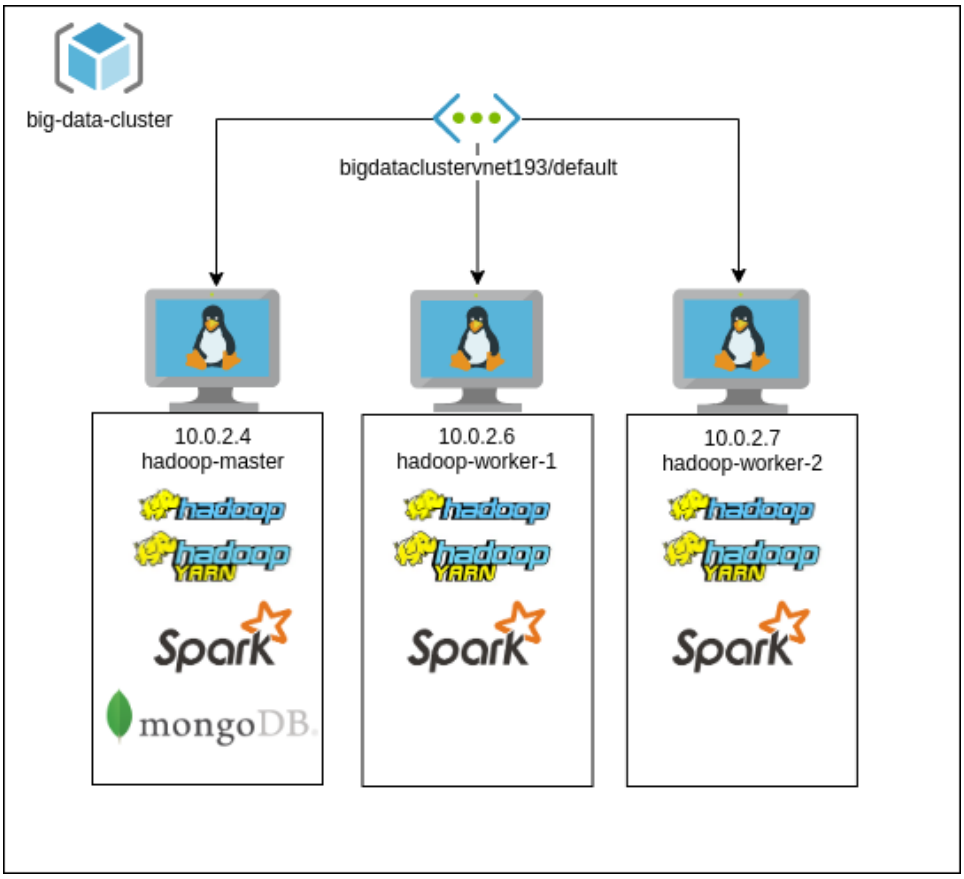
czas: 7.01.2020 - 31.03.2020

2.1 Opis

W drugim etapie do najważniejszych celów należą cele niezrealizowane poprawnie w pierwszym etapie, czyli budowa infrastruktury do przetwarzania danych oraz stworzenie programów pobierających dane. Kolejnymi celami będzie stworzenie programów przetwarzających dane i uruchomienie modeli na przygotowanej infrastrukturze.

2.2 Zadania

1. **Budowa infrastruktury do przetwarzania danych**



Rysunek 1: Diagram przygotowanej infrastruktury.

W celu utworzenia infrastruktury zdolnej przetwarzać dane zostały

utworzone 3 maszyny wirtualne w grupie zasobów **big-data-cluster** na platformie **Azure** w regionie **North Europe**. Są to odpowiednio:

hadoop-master Standard D2s v3 (2 vcpus, 8 GiB memory),

hadoop-worker-1 Standard B2s (2 vcpus, 4 GiB memory),

hadoop-worker-2 Standard B2s (2 vcpus, 4 GiB memory).

Ze względu na specyfikę platformy **Azure** wielkość i rodzaj maszyn może zostać przeskalowany w razie potrzeb. Wszystkie maszyny są dostępne pod publicznymi adresami IP poprzez protokół ssh. Dodatkowo maszyna **hadoop-master** posiada środowisko graficzne i można dostać się do niej również poprzez protokół RDP na porcie 3389.

Ilość maszyn jest zdeterminowana częściowo poprzez limit nałożony na subskrypcję studencką. Limit wynosi 6 vcpus na strefę regionalną maszyn.

2. **Instalacja narzędzi do przetwarzania danych**

Apache Hadoop Został zainstalowany na klastrze. Wartość replikacji plików została zmieniona na **2** ze względu na ilość dostępnych DataNode.

YARN Został skonfigurowany w klastrze.

Apache Spark Został zainstalowany na klastrze oraz skonfigurowany w trybach działania **standalone** oraz poprzez **YARN**. Został połączony ze środowiskiem **Hadoop**.

MongoDB Baza została zainstalowana na maszynie **hadoop-master**. Została skonfigurowana w sposób umożliwiający odwołanie się do niej z sieci lokalnej(poprzez ip **10.0.2.4**). Poprzez odpowiednie opcje konfiguracyjne możliwe jest połączenie z **Apache Spark**.

Rola maszyn w systemach.

	Apache Hadoop	YARN	Apache Spark
hadoop-master	NameNode	ResourceManager	Master
hadoop-worker-1	DataNode	NodeManager	Slave
hadoop-worker-2	DataNode	NodeManager	Slave