Przetwarzanie i analiza dużych zbiorów danych 2020/21

Prowadzący: mgr inż. Rafał Woźniak środa, 11:45

Piotr Wardęcki	234128	234128@edu.p.lodz.pl
Paweł Galewicz	234053	234053@edu.p.lodz.pl
Bartosz Jurczewski	234067	234067@edu.p.lodz.pl

Zadanie 2: System rekomendacji

1. Cel

Zadanie polegało na napisaniu programu który implementuje algorytm "Osoby, które możesz znać". Algorytm działa na zasadzie, że jeżeli dwóch użytkowników ma wielu wspólnych znajomych, to program proponuje im znajomość. Do wykonania powyższego zadania mieliśmy użyć języka programowania *Python* oraz *Apache Spark*.

2. Wprowadzenie

Użytkownicy na których mieliśmy przeprowadzić badanie byli reprezentowani w postaci pliku tekstowego w którym to dane były ułożone w następującej sekwencji: <UŻYTKOWNIK><TABULATOR><ZNAJOMI>, gdzie <UŻYTKOWNIK> oznacza unikalny identyfikator użytkownika, a <ZNAJOMI> to oddzielone po przecinku identyfikatory znajomych użytkownika o identyfikatorze <UŻYTKOWNIK>.

3. Opis implementacji

Do wykonywania zadania niezbędna była instancji *Apache Spark*. Aby ograniczyć liczbę zainstalowanych środowisk skorzystaliśmy z odpowiedniego obrazu dla Dockera [1], który zawierał także *Jupyter Notebook*, *Python* oraz *Miniconda*. Dodatkowo aby ułatwić tworzenie środowiska do kolejnych zadań i między naszymi komputerami skorzystaliśmy z narzędzia *Docker Compose* (nasz plik [2]).

4. Apache Spark

Apache Spark posłużył nam do wczytania pliku oraz przeprowadzenia na nim wszystkich niezbędnych operacji iteracyjnych w celu wyszukania rekomendowanych znajomości dla użytkownika.Bardzo pomocna okazała się funkcja groupByKey, która polega na łączeniu danych w sposób key-value. Dla każdego klucza pobierana jest wartość w sposób iteracyjny. Dodatkowo połączyliśmy inne wbudowane metody Sparka z samodzielnie zaimplementowaną logiką, w celu osiągnięcia zamierzonego kryterium zadania.

5. Wyniki

Poniżej przedstawiamy rekomendacje dla określonych użytkowników. Dodatkowo wyniki dla użytkownika 11 zgadzają się z wzorcem z treści zadania.

Tabela 1. Rekomendacje dla wybranych użytkowników

ID użytkownika	Rekomendacje
924	439, 2409, 6995, 11860, 15416, 43748, 45881
8941	8943, 8944, 8940
8942	8939, 8940, 8943, 8944
9019	9022, 317, 9023
9020	9021, 9016, 9017, 9022, 317, 9023
9021	9020, 9016, 9017, 9022, 317, 9023
9022	9019, 9020, 9021, 317, 9016, 9017, 9023
9990	13134, 13478, 13877, 34299, 34485, 34642, 37941
9992	9987, 9989, 35667, 9991
9993	9991, 13134, 13478, 13877, 34299, 34485, 34642, 37941

6. Wnioski

- *Apache Spark* jest szczególnie przydatny do równoległego przetwarzania rozproszonych danych za pomocą algorytmów iteracyjnych.

- Konteneryzacja *Apache Spark* wraz z *Jupyter Notebook* pozwoliła na szybką konfigurację środowiska oraz jego proste odtworzenie na pozostałych komputerach członków zespołu.

Bibliografia

[1] *Jupyter Notebook Python, Spark Stack* `https://hub.docker.com/r/jupyter/pyspark-notebook`

[2] *Plik Docker Compose do zadania 2* `https://github.com/jurczewski/PiADZD/blob/master/zad2/docker-compose.yml`