# VINF projekt- Parsovanie ocenení z wikipédie

Michal Sorát

Ústav informatiky, informačných systémov a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave

December 13, 2022

## 1 Úvod do problematiky

Tento projekt, rovnako ako aj predmet, sa zaoberá dôležitou oblasťou informatiky, ktorou je vyhľadávanie informácii. Na internete je enormné množstvo dát, ktoré je nutné vedieť prehľadávať a vedieť z nich vyberať tie dôležité pre nás. Následne ďalšou kľúčovou vlastnosťou procesu vyhľadávania a extrakcie dát je optimalizácia. Optimalizovať riešenie tak, aby bolo čo najviac efektívne, rýchle a presné. Konkrétne tento projekt sa zaoberá parsovaním ocenení z wikipédie (dátum získania, kategória, udalosť pri ktorej bola cena odovzdaná, typ ocenenia, miesto odovzdania, kto ju odovzdal či prebral...) a je napísaný v jazyku Python. Vyhľadávanie je vykonávané pomocou regulárnych výrazov doplnené o jednotlivé optimalizačné techniky akými sú indexovanie či paralelné spracovanie.

## 2 Existujúce riešenia problému

Najväčším a najobľúbenejším vyhľadávačom súčastnosti je bezpochyby Google, ktorý vlastní viac ako 80% podielu vyhľadávania na webe. Okrem toho zachytáva takmer 95% mobilnej návštevnosti [6].

Google neustále mapuje internet a ďalšie zdroje tak, aby používateľ bol spojený s najrelevantnejšími a najužitočnejšími informáciami. Index vyhľadávania možno prirovnať ku knižnici, ktorá je stále obohacovaná o nové dáta akými sú webové stránky, obrázky, knihy videá a mnoho ďalších [1].

Používateľ vyhľadáva informácie pomocou Googlu, ktorý prehľadáva v danom okamihu stovky miliárd webových stránok a ďalšieho obsahu, ktorý je uložený v indexe. Indexy boli vytvorené softvérmi, nazývanými prehľadávače, ktoré automaticky navštevujú verejne prístupné stránky a ukladajú informácie o tom čo nájdu do indexov.

Keď indexové prehľadávače nájdu webovú stránku, systémy vykreslia jej obsah rovnako ako prehliadač. Do úvahy berú všetky dôležité signály od kľúčových slov po aktuálnosť webu.

Index vyhľadávania Google obsahuje stovky miliárd webových stránok a zaberá viac ako 100 000 000 gigabajtov. Obsahuje záznam pre každé slovo zistené na každej webovej stránke, ktorú systém zindexoval. Pri indexovaní webovej stránky sú pridané záznamy pre všetky slová, ktoré obsahuje.

V skutočnosti Google využíva viacero indexov rôznych typov informácií, ktoré sú zhromaždované už spomenutým prehľadávaním, ale aj prostredníctvom partnerstiev, informačných kanálov či vlastnej encyklopédie faktov nazývanej znalostná databáza [2].

# 3 Popis riešenia, použitý softvér

## 3.1 Vývojové prostredie a programovací jazyk

Projekt bol zhotovený vo vývojovom prostredí PyCharm od spoločnosti JetBrains. Je napísaný v jazyku Python vo verzii 3.9.

## 3.2 Použité technológie a knižnice

### 3.2.1 Modul re

Modul re poskytuje sadu výkonných prostriedkov regulárnych výrazov, ktoré umožňujú rýchlo skontrolovať, či sa daný reťazec zhoduje s daným vzorom (pomocou funkcie match) alebo či takýto vzor obsahuje (pomocou funkcie search) [4]. V tomto projekte prevažuje najmä využívanie funkcie search.

### 3.2.2 Modul os

Modul os v jazyku Python poskytuje funkcie na vytvorenie a odstránenie priečinka či súboru, načítanie ich obsahov, či vykonávanie zmien [3]. V tomto projekte je využitý na zistenie existencie súborov obsahujúcich indexy pre vyhľadávanie.

### 3.2.3 Modul time

V projekte je použitý modul time na odmeranie času [5], ktorý bol potrebný na vykonanie vyhľadávania. Táto hodnota je vypísaná po každom vykonanom vyhľadávaní v sekundách.

### 3.2.4 Knižnica PySpark

PySpark je Python API pre Apache Spark, open source distribuovaný výpočtový rámec a súbor knižníc na spracovanie rozsiahlych dát v reálnom čase. V tomto projekte je použitý na pararelné spracovanie dát [7].

PySpark parallelize() je funkcia v SparkContext a používa sa na vytvorenie RDD z kolekcie zoznamov. Resilient Distributed Datasets (RDD) je základná dátová štruktúra PySpark-u. Je to nemenná distribuovaná kolekcia objektov. Každý súbor údajov v RDD je rozdelený na logické oddiely, ktoré môžu byť vypočítané na rôznych uzloch klastra.

# 4 Popis riešenia

## 4.1 Pseudokód

1. Otvorím si XML súbor z ktorého budem parsovať informácie
2. Prechádzam súbor po riadkoch až kým nenarazím na tag <page, značiaci začiatok nového bloku
3. Následne v každom riadku odstraňujem znak nového riadku
4. Takto upravený riadok vložím do premennej
5. Kroky 3. a 4. opakujem až kým nenájdem ukončovací znak bloku <page >
6. Najskôr skontrolujem regexom kategóriu bloku ktorý mám uložený v premennej z predchádzajúceho kroku
7. Ak sa zhoduje s vyhľadávanou kategóriou (t.j. ocenenia angl. award) pokračujem na krok číslo 8. inak na krok číslo 2.
8. Z premennej, kde je uložený aktuálny blok vytiahnem pomocou regexov všetky potrebné informácie a uložím ich do premennej triedy s názvom Trophy
9. Vytvorenú premennú triedy Trophy vložím do pola
10. Kroky 2. až 9. opakujem až kým sa nedostanem po koniec súboru, následne vypíšem výsledky

## 4.2 Zaujímavé funkcie programu

Indexy uložené v súbore, predstavujú usporiadaný zoznam čísiel, ktorý môže byť pri väčších súboroch príliš dlhý. A nakoľko je potreba v tomto zozname zistiť výskyt aktuálneho poradia prehľadávaného bloku je nutné indexy efektívne prehľadávať. Do funkcie s názvom `find_index_match` vstupuje pole všetkých indexov a poradie aktuálne prehľadávaného bloku. V rámci funkcie je uplatnené binárne vyhľadávanie, ktoré nájde/nenájde zhodu s časovou zložitosťou O(log n).

Funkcia `search` search postupuje podľa krokov opísaných v pseudokóde 4.1. Vstupným parametrom je názov prehľadávaného súboru. Ak existuje

súbor s indexami hodnoty z tohto súboru sú vložené do pola s názvom indexes. Neskôr pole indexes optimalizuje rýchlosť prehľadávania jednotlivých blokov.

Funkcia `create_indexes` funguje podobne ako samotné vyhľadávanie, no s tým rozdielom, že pri nájdení zhody kategórie bloku s vyhľadávanou kategóriou sa počítadlo poradia bloku zapíše do súboru obsahujúceho indexy pre daný súbor, ktorý prehľadávam.

# 5 Zhodnotenie riešenia

Každá funkcia v programe a jej vstupné parametre sú dôkladne zdokumentované použitím docstringov, ktoré slúžia na vytvorenie komentárov kódu v jazyku Python. Ak by som mal zhodnotiť subjektívne tento projekt, najväčším prekvapením pre mňa boli počty nájdených výsledkov. Vedel som, že každoročne sa koná veľa súťaží a je rozdaných mnoho ocenení v rozličných kategóriach, ale nepredpokladal som že výsledkov bude až toľko. Možno práve aj to mierne ovplyvnilo čas vykonania operácie vyhľadávania.

## 5.1 Zhodnotenie času

Pri použití indexovania sa čas skracoval približne na polovicu. Pri väčšom počte súborov bol čas vykonania mierne lepší vďaka pararelnému spracovaniu súborov. Jednotlivé súbory boli o veľkosti od 0,1 GB až po 1,5 GB. Jednotlivé časy rozdielnych veľkostí a počtov datasetov môžme vidieť v tabuľke 1.

| Veľkosť datasetu | Čas vykonania bez indexovania | Čas vykonania s indexovaním |
|---|---|---|
| 0,1 GB | 13 sekúnd | 4 sekundy |
| 0,3 GB | 30 sekúnd | 11 sekúnd |
| 1,05 GB | 84 sekúnd | 40 sekúnd |
| 3,6 GB | 264 sekúnd | 124 sekúnd |
| 7 GB | 434 sekúnd | 213 sekúnd |

Obr. 1: Čas vykonania operácie vyhľadávania nad rozdielne veľkými datasetmi.

## 5.2 Testovanie

Na testovanie správnosti fungovania programu bol použitý tzv. "runner"s názvom unittest. Unittest je súčasťou štandardnej knižnice jazyka Python od verzie 2.1. Testovanie je vykonávané v súbore `SearchTest.py`. Nachádzajú sa v ňom 2 unit testy na datasete s názvom `trophies.xml`. Prvý test je úspešný a testuje správnosť počtu nájdených výsledkov. Druhý test je neúspešný nakoľko porovnáva počet správny nájdených výsledkov s inou,

nesprávnou hodnotou. Obidva testy nás informujú o správnom fungovaní programu.

## 6 Spustenie projektu

Na spustenie projektu je potrebný jazyk Python 3 (v tomto riešení bol využitý Python verzie 3.9). Takisto je potreba inštalácie knižnice PySpark, či spustiť program v containeri pomocou Dockeru. Okrem toho je nutné mať vytvorený priečinok files, do ktorého umiestnite súbory na prehľadávanie popr. upraviť cestu k súborom v kóde. Posledným krokom je vložiť názvy prehľadávaných súborov do premennej v časti `main()` s názvom `fileNames`.

Pri spustení projektu sa ako prvé zobrazí používateľské rozhranie ktoré pozostáva z týchto častí:

- Stlačte 1 pre vytvorenie indexov súborov
- Stlačte 2 pre vyhľadávanie
- Stlačte 3 pre ukončenie programu

Jedným z možných použití programu je získanie dát pre štatistické úrady z oblasti športu, kinematografie, vedy a v podstate všetkých oblastí, kde sú odovzdávané ocenenia.

## Literatúra

[1] Ako vyhľadávanie Google funguje. URL: `https://www.google.com/search/howsearchworks/how-search-works/`.

[2] Ako vyhľadávanie Google usporadúva informácie. URL: `https://www.google.com/search/howsearchworks/how-search-works/organizing-information/`.

[3] os - Miscellaneous operating system interfaces. URL: `https://docs.python.org/3/library/os.html`.

[4] re - Regular expression operations. URL: `https://docs.python.org/3/library/re.html`.

[5] time - Time access and conversions. URL: `https://docs.python.org/3/library/time.html`.

[6] Caroline Forsey. The top 6 search engines, ranked by popularity. Oct. 19, 2022. URL: `https://blog.hubspot.com/marketing/top-search-engines`.

[7] Natassha Selvaraj. Pyspark Tutorial: Getting Started with Pyspark. Aug. 2022. URL: `https://www.datacamp.com/tutorial/pyspark-tutorial-getting-started-with-pyspark`.