Міністерство освіти і науки України
Національний технічний університет України „КПІ”
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки
інформації та управління

ЗВІТ

з лабораторної роботи № 3
дисципліни
“ТЕХНОЛОГІЇ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ В УМОВАХ ВЕЛИКИХ ДАНИХ”
на тему:

„Реалізація на основі Apache Hadoop”

Виконали:
студенти групи ІТ-01мн
Панасюк Станіслав
Лесогорський Кирило

Перевірив:
доц. Жереб К. А.

Київ – 2021

Зміст

1. Постановка задачі

Обрану задачу необхідно реалізувати, використовуючи Apache Hadoop. У якості задачі було обрано побудову системи пошуку схожих зображень. У ядрі системи лежатиме використання D-hash для знаходження хешу зображення. D-hash дозволяє точно та швидко шукати схожі зображення. Він стійкий до скейлінгу зображеня, але погано справляються з обрізаними та повернутими під кутом зображеннями. Тому цю техніку аугментовано за допомогою наступного прийому: при завантаженні зображення воно буде аугментовано за допомогою декількох фільтрів, при цьому для кожного фільтру буде згенеровано хеш і збережено у базу даних. При пошуку зображення буде використовуватись оператор XOR для знаходження зображень зі схожими хешами.

2. Обрані інструменти

Для виконання третьої лабораторної роботи буде використано стандартні інструменти Java із доданою до них бібліотекою Apache Hadoop. У нашому випадку третя та четверта лабораторні роботи взаємопов'язані, адже четверта робота, використовуючи Apache Spark, проводить підготовку даних, а вже третя робота працює із ними за рахунок написаної MapReduce програми.

3. Опис роботи програмного забезпечення

Код для підрахунків D-hash використовується ідентичний до попередніх робіт, тому у даній роботі розглянемо лише нові зміни.

Було написано цілком новий застосунок, який Apache Hadoop добре розуміє. Таким застосунком є MapReduce job, що читає попередньо підготовані через Apache Spark дані з Sequence File, обраховує хеші кожного зображення та зберігає їх у файл.

Застосунок складається з п'яти основних частин:

1. Мапери (це Map частина нашої MapReduce job). Вони реалізують інтерфейс Mapper та описують логіку, за якою будуть оброблятись окремі часточки даних (у нашому випадку це окремі зображення). Тут ми кожне зображення перетворюєм на його хеші;
2. Ред'юсери (це Reduce частина нашої MapReduce job). Вони реалізують інтерфейс Reducer та описують логіку, за якою усі підготовані часточки даних будуть агрегуватись згідно до ключа, який надає мапер із кожним обробленим значенням. У нашому випадку ключ – ім'я файла. Тут ми зберігаємо хеші згідно з ключами до файлу;

3. Утиліти – тут ми скопіювали ту саму логіку підрахунків D-hash;
4. Модель – у ній ми створили кастомний тип даних, який передаємо від мапера до ред'юсера, що дозволяє передати масив LongWritable даних (наших хешів);
5. Драйвер – основний клас, який відповідає за конфігурацію MapReduce job.

4. Отримані результати

В результаті виконання роботи було отримано наступні результати з процесінгу:

Рисунок 4.1 - Вихідний файл із ключами у вигляді імені зображення та значеннями у вигляді масиву хешів

5. Висновки

У ході лабораторної роботи було створено MapReduce додаток із використанням Apache Hadoop для досягнення ідентичної до першої та другої лабораторних робіт цілі.