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 № 4

дисципліни

“ТЕХНОЛОГІЇ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ В УМОВАХ

ВЕЛИКИХ ДАНИХ”

на тему:

„Реалізація на основі Apache Spark”

Виконали:

студенти групи IT-01мн

Панасюк Станіслав

Лесогорський Кирило

Перевірив:

доц. Жереб К. А.

Київ – 2021

Зміст

[1. Постановка задачі 3](#_Toc91019724)

[2. Обрані інструменти 3](#_Toc91019725)

[3. Опис роботи програмного забезпечення 3](#_Toc91019726)

[4. Отримані результати 4](#_Toc91019727)

[5. Висновки 5](#_Toc91019728)

# Постановка задачі

Обрану задачу необхідно реалізувати, використовуючи Apache Spark. У якості задачі було обрано підготовку зображень до поглинення через Apache Hadoop задля вирішення тої ж задачі, що і у перших лабораторних роботах. 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 філтрів, 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.

# Обрані інструменти

Для виконання четвертої лабораторної роботи буде використано стандартні інструменти Java із доданою до них бібліотекою Apache Spark. У нашому випадку третя та четверта лабораторні роботи взаємопов’язані, адже четверта робота підготовує зображення, використовуючи Apache Spark, а вже третя робота працює із ними за рахунок написаної MapReduce програми.

# Опис роботи програмного забезпечення

У ході виконання даної лабораторної роботи було створено додаткову логіку у додатку із першою та другою лабораторними, що допомагає третій лабораторній ефективно опрацьовувати зображення.

Сама ж логіка працює у декілька кроків:

1. Налаштування JavaSparkContext. Бібліотека, яку ми використовуємо, дозволяє запустити Apache Spark без додаткового встановлення його на компьютері, чим ми і скористались. У реальному проєкті Spark був би запущений на віддаленому сервері та допомагав ефективно паралелити задачі між багатьма серверами;
2. Зчитування зображень у зручний для Apache Spark формат. Ми зчитуємо зображення, як бінарні файли та складаємо їх до JavaPairRDD, де ключом э ім’я зображення, а значенням – поток бінарних даних, що є тілом зображення;
3. Записуємо дані у зручний для Apache Hadoop формат. На цьому кроці ми, використовуючи SequenceFile.Writer записуємо зображення один за одним у єдиний SequenceFile, що складається з ключів та значень (у нашому випадку, з імен та бінарних тіл зображень);

# Отримані результати

В результаті виконання роботи було отримано наступні результати з процесінгу:

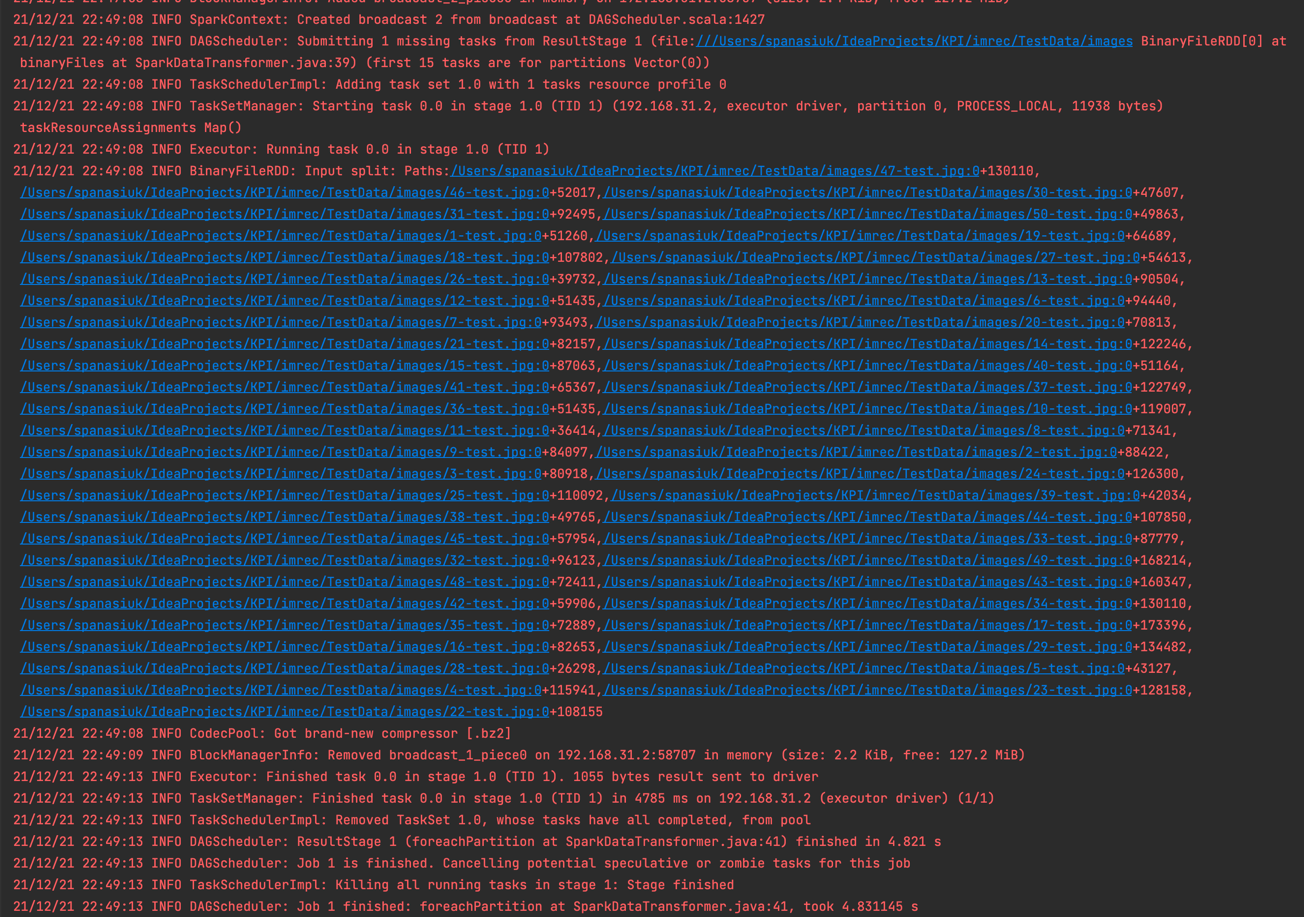


Рисунок 4.1 – Частина логів виконання задачі

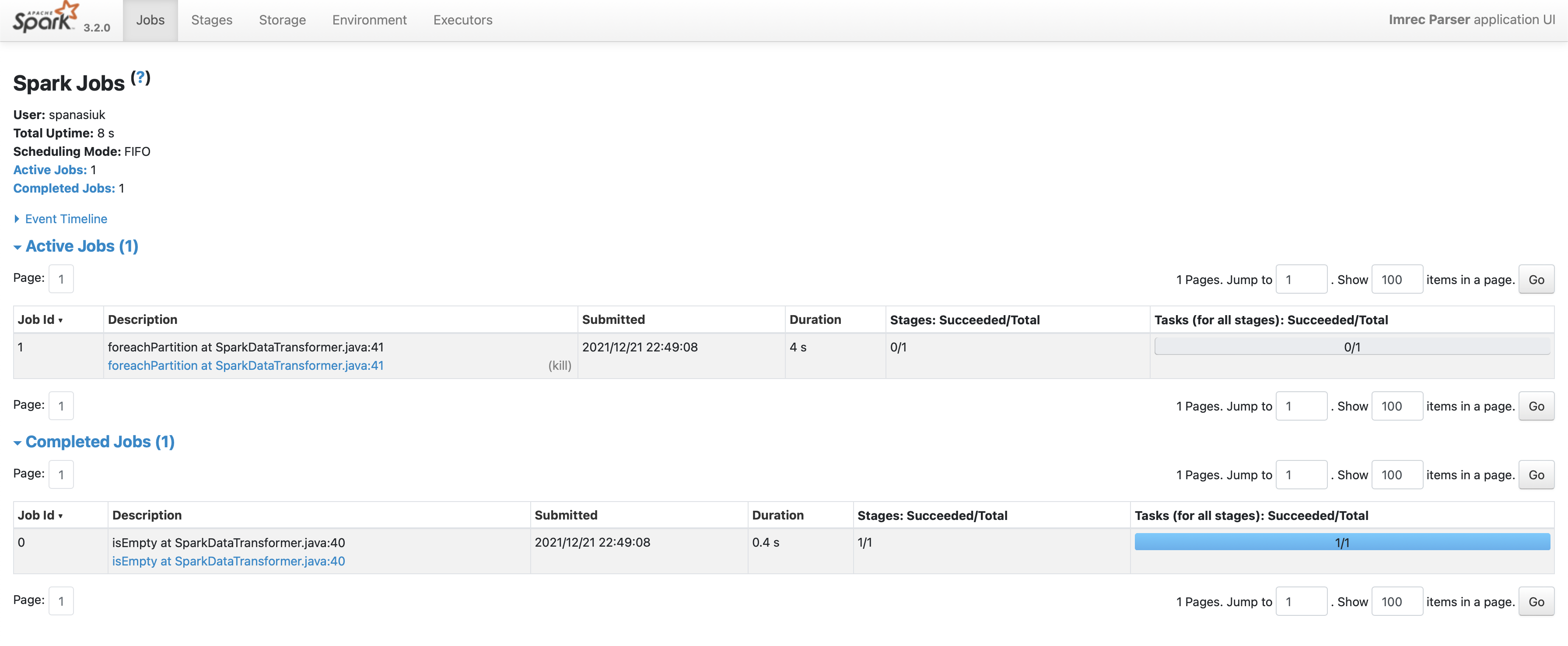


Рисунок 4.2 - UI Apache Spark, що показує стан поточних задач

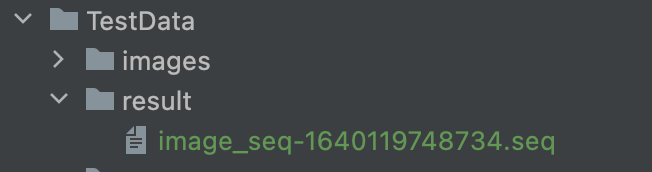


Рисунок 4.3 - Результат опрацювання зображень

# Висновки

У ході лабораторної роботи було створено додаток із використанням Apache Spark для підготовки даних з ціллю подальшого їх поглинення через Apache Hadoop