Міністерство освіти і науки, молоді та спорту України

Національний Технічний Університет України

«Київський Політехнічний Інститут ім. Ігоря Сікорського»

Навчально-науковий комплекс

Кафедра системного проектування

Курсова робота

з дисципліни

“Паралельні обчислення”

«Створення інвертованого індексу»

Виконав:
студент ДА-71
Кузнєцов Олексій

Київ – 2020

Завдання

1. Вивчити теорію з глави 4 «Побудова індексу».
2. Розробити модель системи для побудови індексу в паралельному середовищі, заклавши можливість варіювати кількість потоків (від 1 до N).
3. Завантажити вхідні текстові дані згідно до варіанту.
4. Створити проект Github (або у будь-якому іншому репозиторії).
5. Створити програмну реалізацію моделі з п.2, правильно використовуючи систему контролю версій (бажано, щоб один commit був однією атомарною зміною в проекті); описувати кожен commit в одному стилі.
6. Виконати тестування розробленої системи.
7. Зробити графічне порівняння часу роботи системи, задаючи різну кількість потоків (від 1 до N).
8. Перевірити правильність результату роботи послідовного та паралельних рішень.
9. Зробити форматування коду в одному стилі.
10. Зробити файл з інструкцією як зібрати та запустити проект, додати його в репозиторій.
11. Зробити протокол з результатами роботи, що має містити інформацію з попередніх пунктів.

Теоретична основа та алгоритм роботи

Інвертований індекс (англ. Inverted index) - структура даних, в якій для кожного слова колекції документів у відповідному списку перераховані всі документи в колекції, в яких воно зустрілося. Інвертований індекс використовується для пошуку за текстами.

Приклад :

Нехай у нас буде колекція з 3 тексті T0 ="it is what it is", T1 ="what is it" і T2="it is a banana", тоді інвертований індекс буде виглядати наступним чином :

"a": {2}

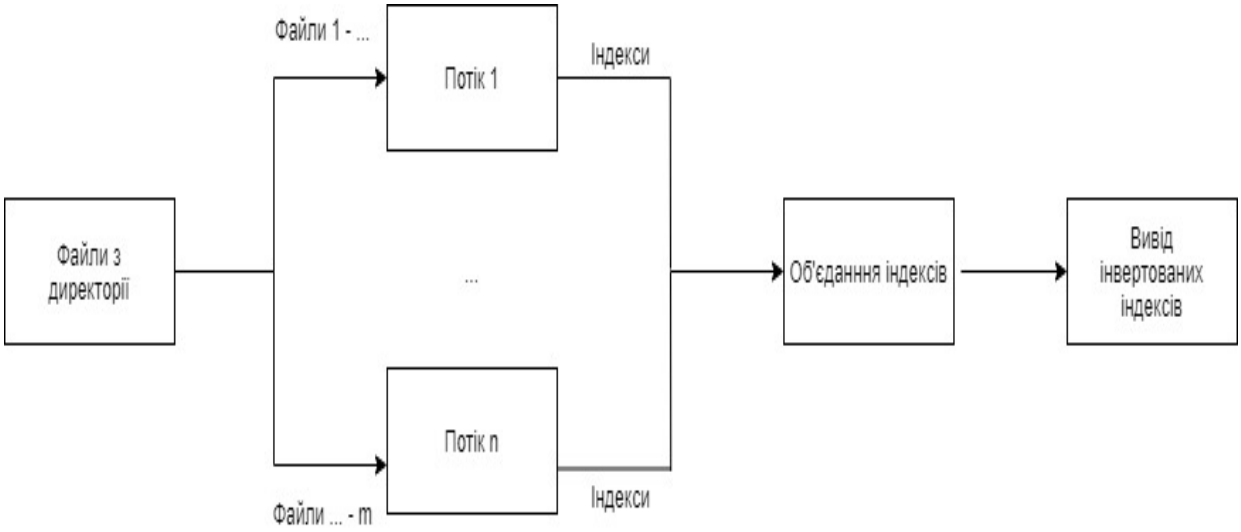
"banana": {2}

"is": {0, 1, 2}

"it": {0, 1, 2}

"what": {0, 1}

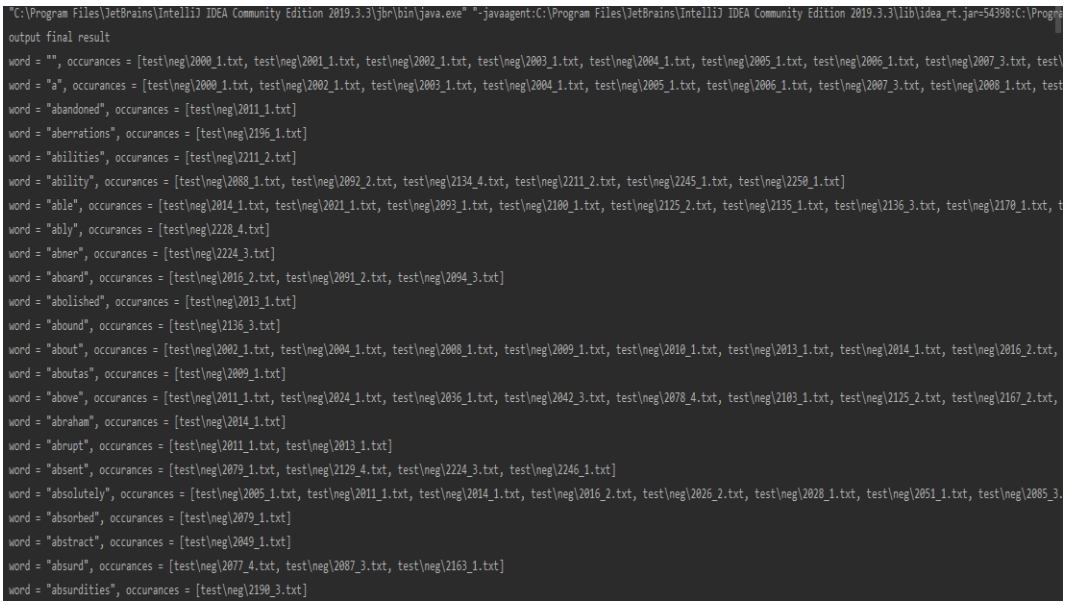
Модель роботи :



Модель роботи індексатора

Реалізація та тестування програми

Алгоритм побудови індексу реалізований у цій роботі мовою програмування JAVA. Були використані вбудовані бібліотеки цієї мови для роботи з паралелізмом та структурами даних. Приклад роботи програми продемонстрований на малюнку нижче.



Скріншот роботи програми

В процесі розробки програми була використана система контролю версій Git за допомогою веб-сервісу Github.

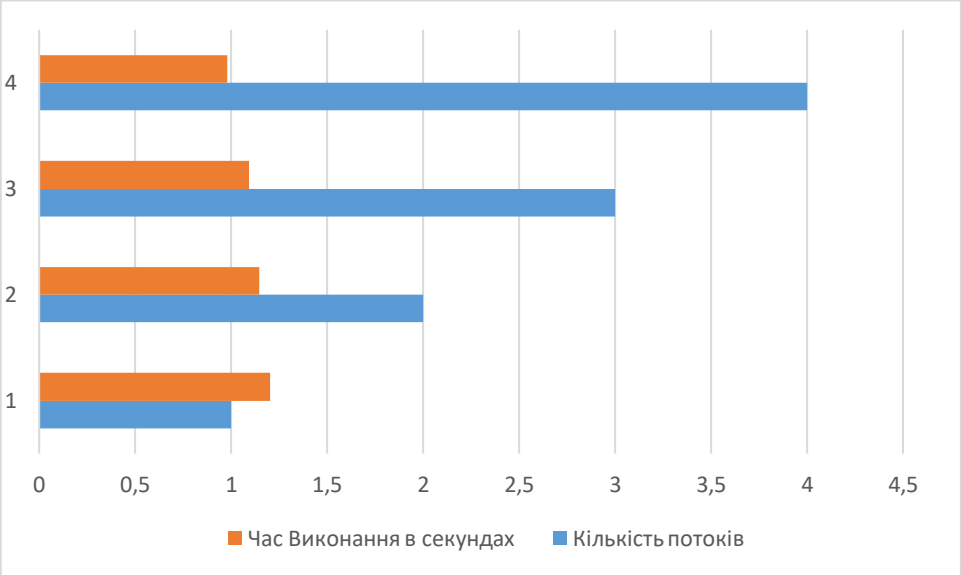
Сторінка проекту: https://github.com/Kinelan/cource_work. На ній знаходиться інструкція по компіляції та використанню проекту.

Перевіримо залежність роботи програми від кількості потоків, які ми виділимо на роботу програми. В якості тестових прикладів використаємо базу новин, в якій присутні наступні папки:

1. aclImdb\test\neg – N=250 файлів
2. aclImdb\test\pos – N=250 файлів
3. aclImdb\train\neg – N=250 файлів
4. aclImdb\train\pos – N=250 файлів
5. aclImdb\train\unsup – N=1000 файлів

Результати тестування ось такі :

Кількість потоків	Час виконання в нс	Час Виконання в секундах
1	1203752600	1,2037526
2	1145782000	1,145782
3	1092709500	1,0927095
4	981291400	0,9812914



Швидкодія програми

Як можемо побачити, завдяки використанню паралелізму, вдалося скоротити час виконання програми. Цей ефект буде більше замітний при використанні більшого об'єму інформації, тому що

алгоритм має найвищу ефективність при роботі з великими та надвеликими масивами даних.

Висновки

У ході виконання курсової роботи було досліджено алгоритм побудови інвертованого індексу. На етапі теоретичного ознайомлення була розроблена модель програми.

На етапі проектування я реалізував алгоритм та створив консольний додаток. Результатом розробки програми є закріплення навичок проектування паралельних алгоритмів та використання засобів мови програмування JAVA у паралельних додатках. Також була показана ефективність використання паралельних алгоритмів для збільшення швидкодії програми.