Big Data Assignment 2 Report

Arsenii Pavlov ars.pavlov@innopolis.university

Methodology

Data Preparation

1) Used a Parquet file containing Wikipedia articles. Selected 4000 documents using PySpark's sample() and limit() functions.
2) Each document is saved as <doc_id>_<doc_title>.txt in HDFS. Spaces in titles are replaced with _.

pipeline:

1) Read the Parquet file into a Spark DataFrame.
2) Extracted id, title, and text columns.
3) Saved documents to HDFS to /data using tab-separated format <doc_id>\t<doc_title>\t<doc_text>.

Indexer Tasks

The indexer uses two Hadoop MapReduce pipelines and save results in Cassandra.

1) **TF Calculation**

Mapper1

1) Read input documents from HDFS.
2) Tokenizes text cast to lowercase and splits on non-alphanumeric characters.
3) Convert to key-value pairs: <term>#<doc_id> to 1.

Reducer1

1) Aggregates counts for <term>#<doc_id> to compute TF.
2) Output: <term> <doc_id> <tf> to /tmp/index/pipeline1

2) **DF Calculation**

Mapper2

1) Reads TF data from Pipeline 1.
2) Convert term to 1 for each unique term-document pair.

Reducer2:

1) Sums counts to compute DF.
2) Output <term>\t<df> to /tmp/index/pipeline2

3) **Saving to Cassandra**

Tables:

vocabulary: words and DF values
inverted_index words and TF values
documents: map document IDs and titles

Loading Data: app.py reads HDFS outputs and inserts data into Cassandra using batch queries.

Ranker Tasks

ranker use PySpark to compute BM25 scores for queries.

BM25 Calculation

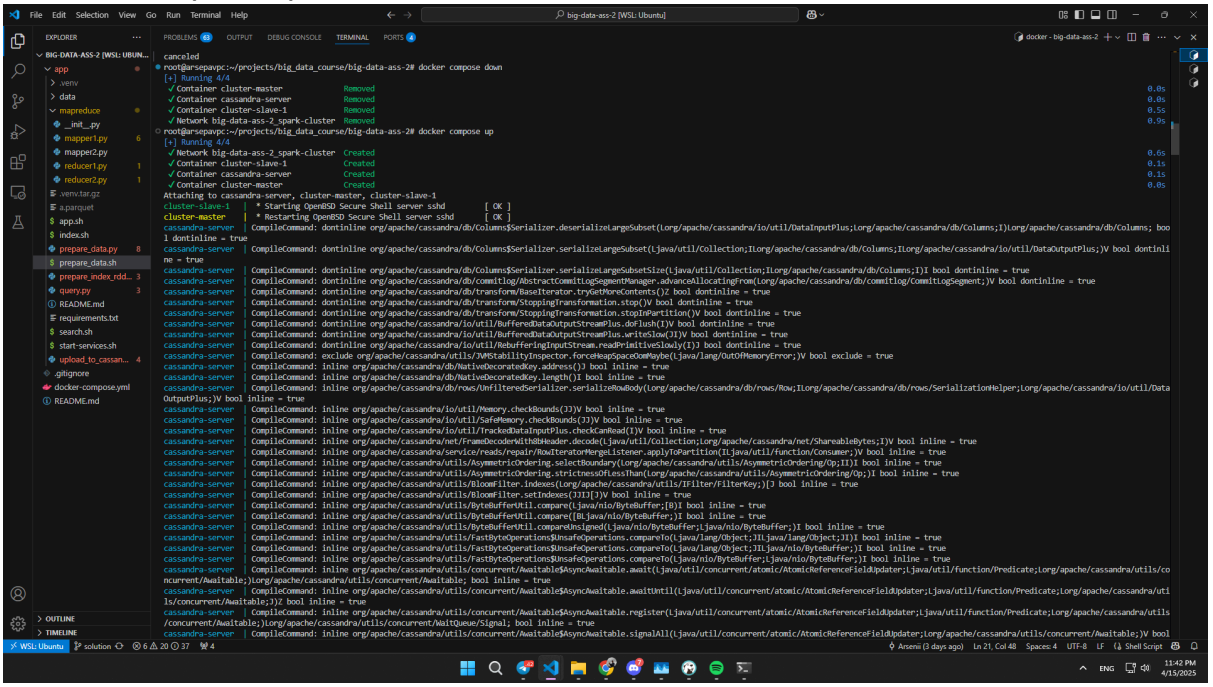
1) Fetch DF for query terms from vocabulary table.
2) Fetch TF and document lengths from inverted_index and documents.
3) Compute BM25 score using provided formula
4) Sum scores across all query terms.
5) Rank documents by score and return top 10.

Spark

1) Initialize Spark and Process Query
2) Compute BM25 Scores
3) Rank and Output Results

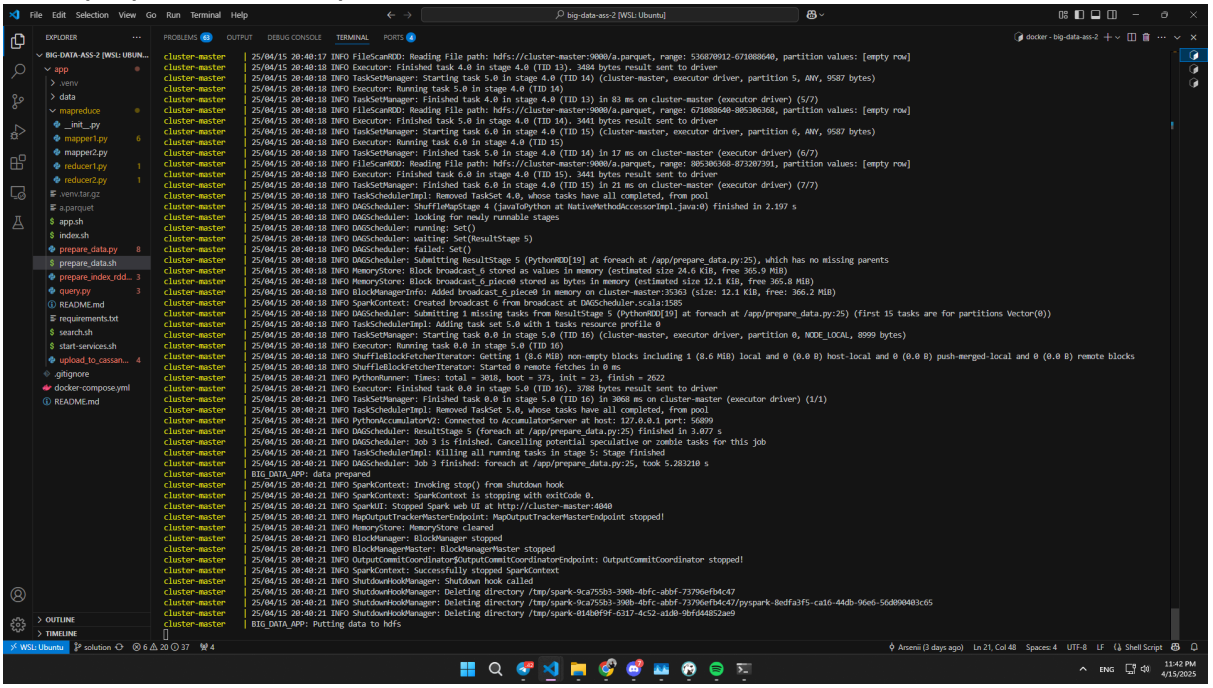
Demonstration

To start containers run
docker compose up

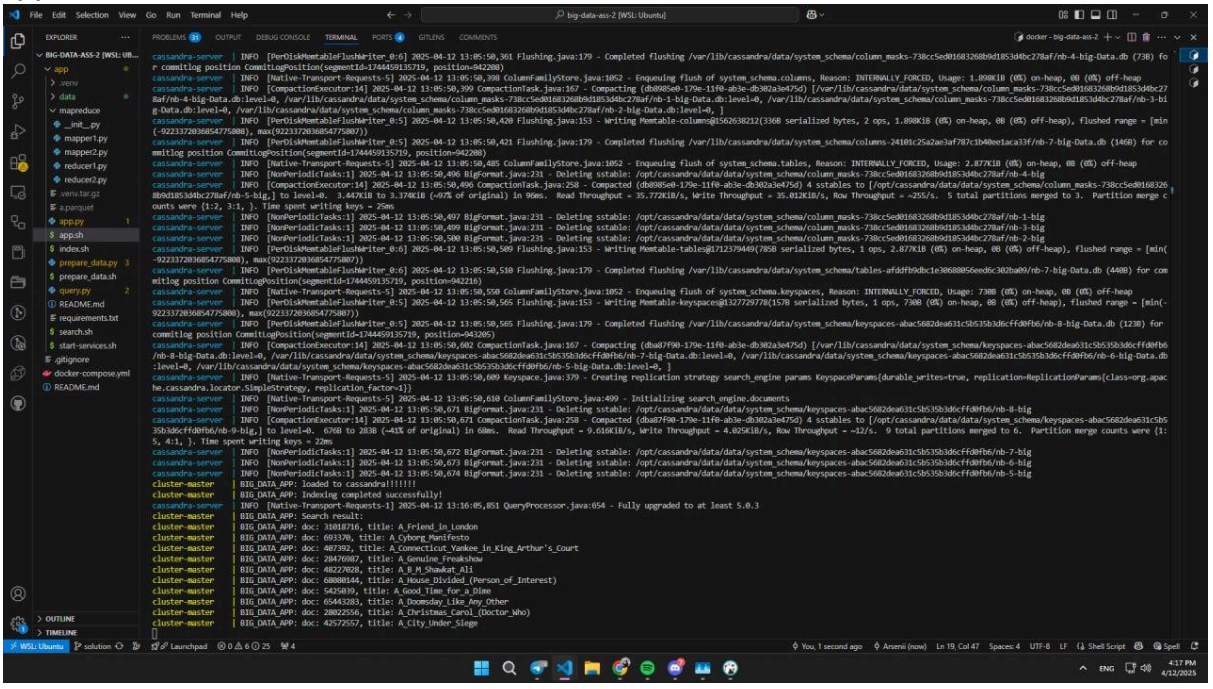


It will run all containers and run data preparation indexing and 1 query

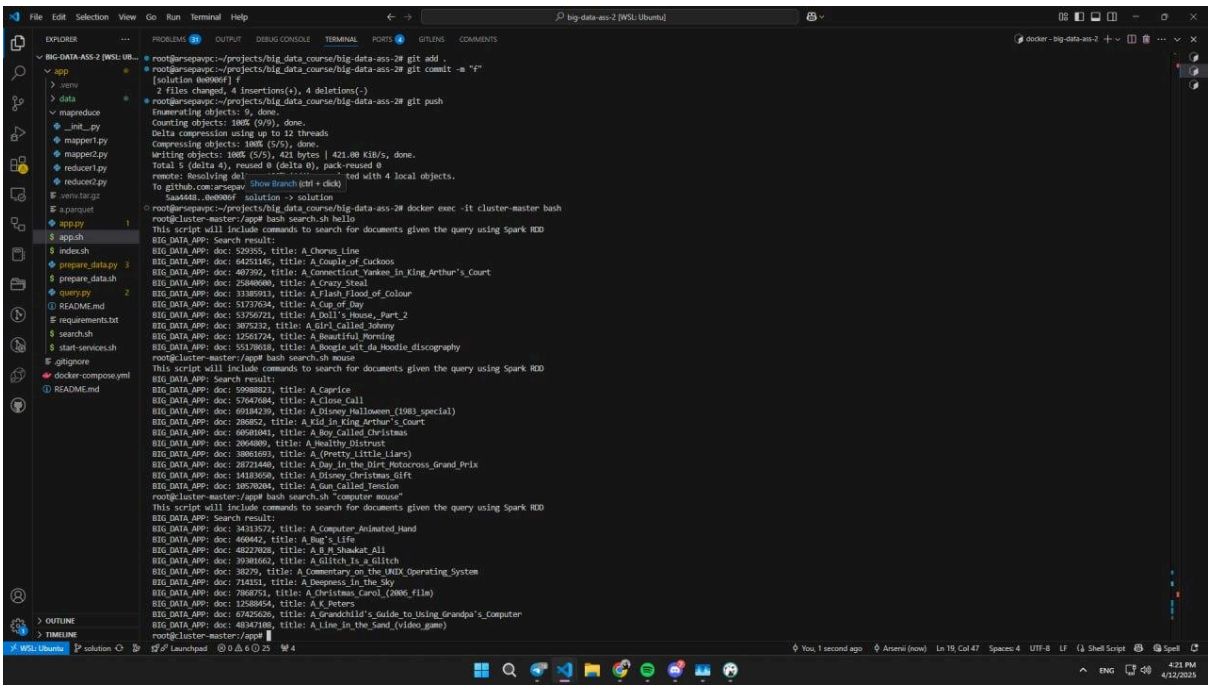
first it prepare data and put to hdfs



I disabled all logs except which contains BIG_DATA_APP so i van see in which my app is



Example with running 3 queries:



The results are actually good it can be seen in the example with computer mouse titles (they all connected to computer)