Search Engine Implementation Report

Methodology

System Architecture

The search engine implementation follows a distributed architecture leveraging Hadoop for data processing and Cassandra for data storage. The system consists of several key components:

1. **Data Preparation Pipeline**
    - Uses PySpark for efficient data processing
    - Reads from a Parquet file containing document data
    - Creates two data formats:
        - Individual text files for each document
        - A combined TSV file with tab-separated fields (doc_id, title, text)
    - Configures Spark with optimized memory settings:
        - Driver memory: 2GB
        - Executor memory: 2GB
        - Memory fraction: 0.8
        - Storage fraction: 0.3
2. **Indexing Pipeline** The indexing process is implemented as a two-stage MapReduce pipeline:

    Stage 1: Term Frequency and Document Length Calculation

    - Mapper (`mapper1.py`):
        - Processes input documents line by line
        - Implements text preprocessing:

        - Converts text to lowercase
        - Removes non-alphanumeric characters
        - Tokenizes text into words
    - Calculates term frequencies for each document
    - Emits three types of records:
        - Term frequency: `term\tdoc_id\tfreq`
        - Document length: `!DOCLEN\tdoc_id\tlength`
        - Document title: `!TITLE\tdoc_id\ttitle`
- Reducer (`reducer1.py`):
    - Connects to Cassandra for data storage
    - Processes records in groups by key
    - Stores data in two Cassandra tables:
        - `term_frequency`: (word, doc_id, tf)
        - `document_stats`: (doc_id, title, doc_length)
    - Emits intermediate data for Stage 2:
        - Document count: `!TOTALDOCS\t1`
        - Total length: `!TOTALLENGTH\tlength`
        - Term occurrences: `term\t1`

Stage 2: Document Frequency and Corpus Statistics

- Mapper (`mapper2.py`):
    - Passes through the intermediate data from Stage 1
    - No additional processing needed
- Reducer (`reducer2.py`):
    - Calculates document frequencies for each term
    - Computes corpus-wide statistics
    - Stores results in two Cassandra tables:

- `term_stats`: (word, df)
- `corpus_stats`: (stat_key, stat_value)

3. **Search Implementation**
   - Uses BM25 ranking algorithm with configurable parameters:
     - k1 = 1.0 (term frequency saturation)
     - b = 0.75 (length normalization)
   - Implements distributed search using PySpark
   - Query processing pipeline:
     1. Text preprocessing (lowercase, tokenization)
     2. Term frequency lookup from Cassandra
     3. Document frequency and corpus statistics retrieval
     4. BM25 score calculation
     5. Result ranking and presentation

Data Storage Schema

The system uses four Cassandra tables with the following schemas:

1. **term_frequency**

```
CREATE TABLE term_frequency (
    word text,
    doc_id text,
    tf int,
    PRIMARY KEY (word, doc_id)
)
```

   - Stores term frequencies for each document
   - Composite primary key for efficient lookups

2. **document_stats**

```
CREATE TABLE document_stats (
    doc_id text PRIMARY KEY,
    title text,
    doc_length int
)
```

- Stores document metadata and length
- Single primary key for quick document lookups

3. **term_stats**

```
CREATE TABLE term_stats (
    word text PRIMARY KEY,
    df bigint
)
```

- Stores document frequency for each term
- Used for IDF calculation in BM25

4. **corpus_stats**

```
CREATE TABLE corpus_stats (
    stat_key text PRIMARY KEY,
    stat_value bigint
)
```

- Stores corpus-wide statistics
- Used for BM25 normalization

Performance Optimizations

1. **Data Processing**
    - Efficient text preprocessing using regular expressions
    - Batch processing of Cassandra operations

- Use of prepared statements for database operations
- Memory-efficient document processing

2. **Search Optimization**
   - Broadcast variables in PySpark for corpus statistics
   - Efficient join operations for score calculation
   - Batch retrieval of term postings
   - Caching of frequently accessed data
3. **Storage Optimization**
   - Denormalized schema for fast retrieval
   - Composite keys for efficient lookups
   - Separate tables for different types of data
   - Batch inserts for better performance

Demonstration

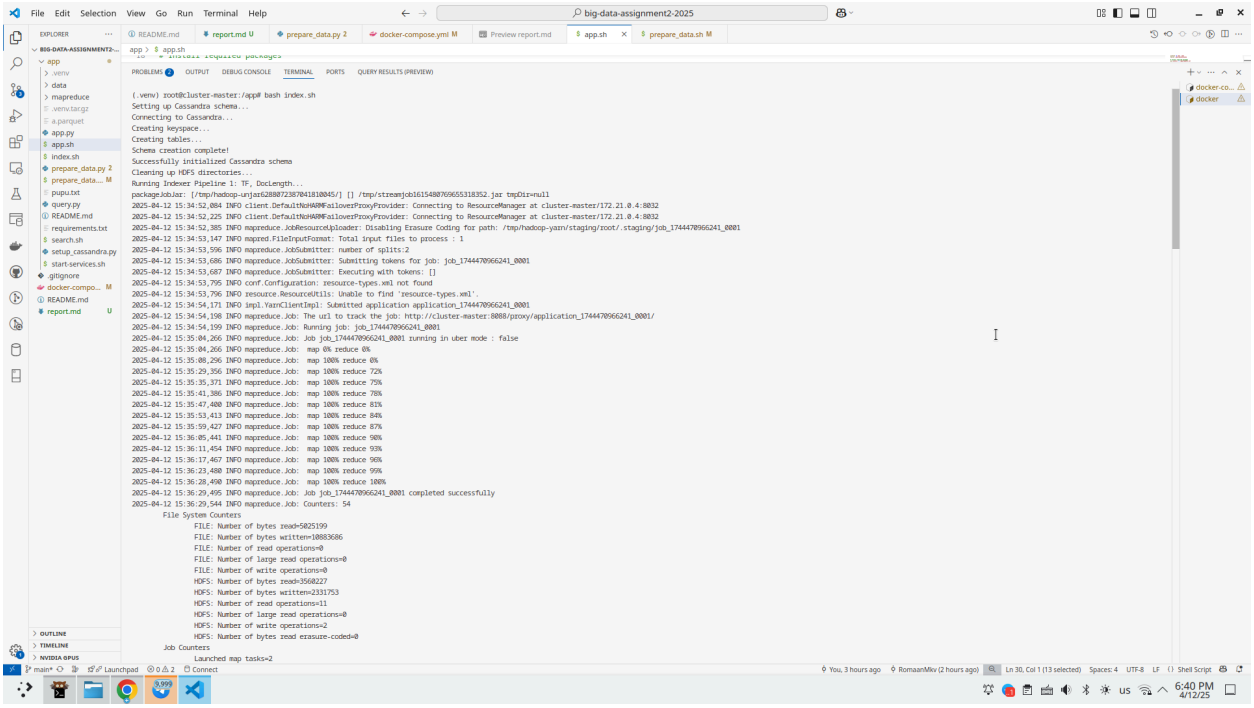
Running the Search Engine

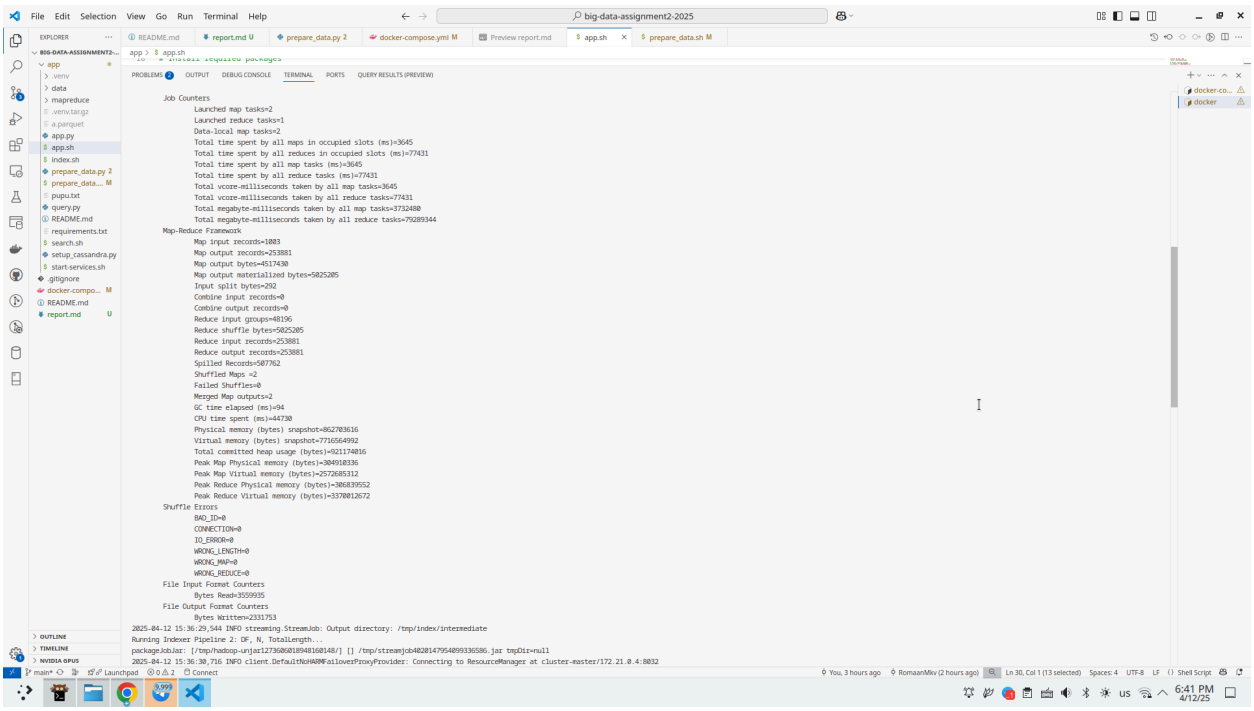
1. **Setup and Data Preparation**

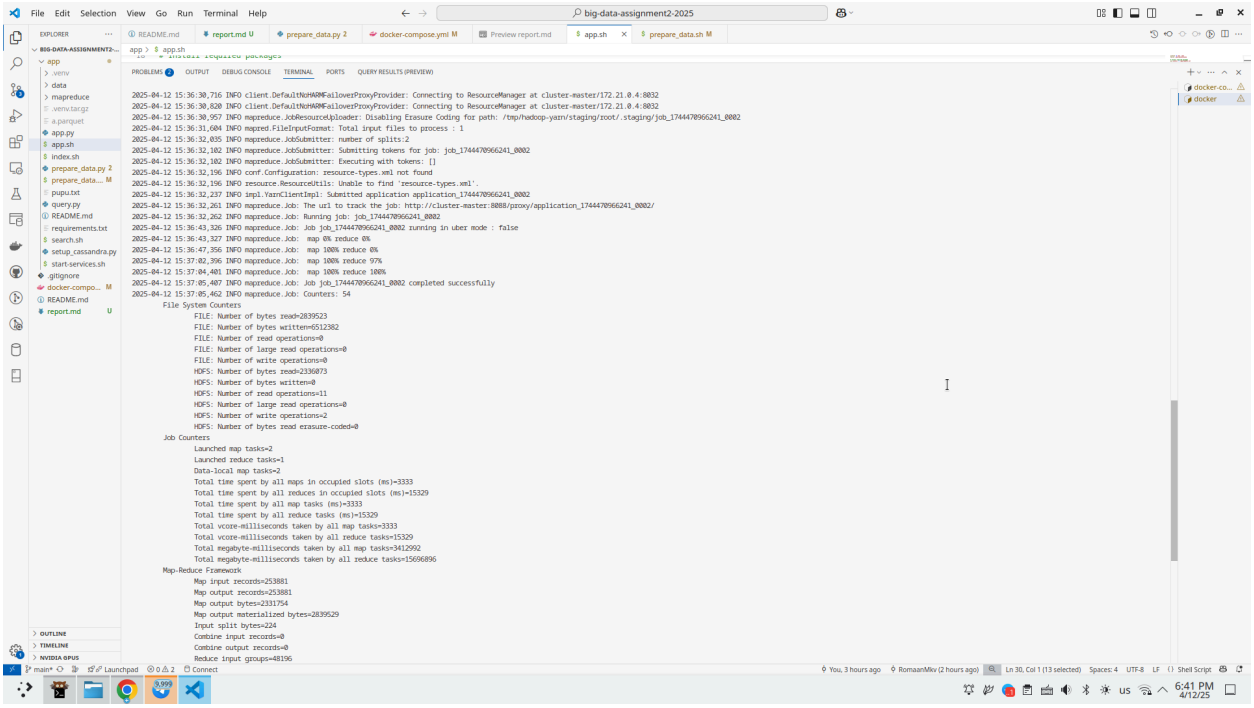
```
# Start the required services (Hadoop, Cassandra)
./app/start-services.sh

# Prepare the data (creates TSV and individual files)
./app/prepare_data.sh
```

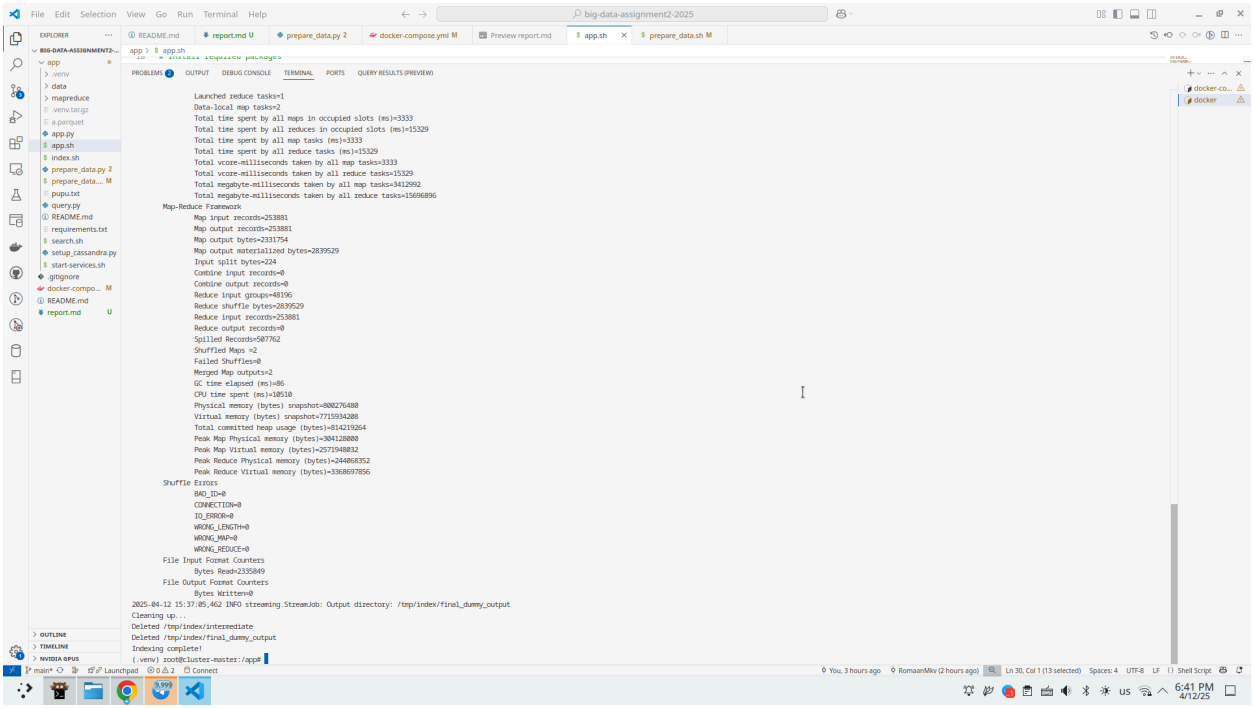
2. **Indexing Documents**

```
# Run the indexing pipeline
./app/index.sh
```


(.venv) root@cluster-master:/app# bash index.sh
Setting up Cassandra schema...
Connecting to Cassandra...
Creating keyspace...
Creating tables...
Schema creation complete!
Successfully initialized Cassandra schema
Cleaning up HDFS directories...
Running Indexer Pipeline 1: TF, DocLength...
2025-04-12 15:34:52,084 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.21.0.4:8032
2025-04-12 15:34:52,225 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.21.0.4:8032
2025-04-12 15:34:52,385 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1744470966241_0001
2025-04-12 15:34:53,147 INFO mapred.FileInputFormat: Total input files to process : 1
2025-04-12 15:34:53,596 INFO mapreduce.JobSubmitter: number of splits:2
2025-04-12 15:34:53,686 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1744470966241_0001
2025-04-12 15:34:53,687 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-12 15:34:53,795 INFO conf.Configuration: resource-types.xml not found
2025-04-12 15:34:53,796 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-12 15:34:54,171 INFO impl.YarnClientImpl: Submitted application application_1744470966241_0001
2025-04-12 15:34:54,198 INFO mapreduce.Job: The url to track the job: http://cluster-master:8088/proxy/application_1744470966241_0001/
2025-04-12 15:34:54,199 INFO mapreduce.Job: Running job: job_1744470966241_0001
2025-04-12 15:35:04,266 INFO mapreduce.Job: Job job_1744470966241_0001 running in uber mode : false
2025-04-12 15:35:04,266 INFO mapreduce.Job: map 0% reduce 0%
2025-04-12 15:35:08,296 INFO mapreduce.Job: map 100% reduce 0%
2025-04-12 15:35:29,356 INFO mapreduce.Job: map 100% reduce 72%
2025-04-12 15:35:35,371 INFO mapreduce.Job: map 100% reduce 75%
2025-04-12 15:35:41,386 INFO mapreduce.Job: map 100% reduce 78%
2025-04-12 15:35:47,400 INFO mapreduce.Job: map 100% reduce 81%
2025-04-12 15:35:53,413 INFO mapreduce.Job: map 100% reduce 84%
2025-04-12 15:35:59,427 INFO mapreduce.Job: map 100% reduce 87%
2025-04-12 15:36:05,441 INFO mapreduce.Job: map 100% reduce 90%
2025-04-12 15:36:11,454 INFO mapreduce.Job: map 100% reduce 93%
2025-04-12 15:36:17,467 INFO mapreduce.Job: map 100% reduce 96%
2025-04-12 15:36:23,480 INFO mapreduce.Job: map 100% reduce 99%
2025-04-12 15:36:28,490 INFO mapreduce.Job: map 100% reduce 100%
2025-04-12 15:36:29,495 INFO mapreduce.Job: Job job_1744470966241_0001 completed successfully
2025-04-12 15:36:29,544 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=5025199
FILE: Number of bytes written=10883686
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=3560227
HDFS: Number of bytes written=2331753
HDFS: Number of read operations=11
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
HDFS: Number of bytes read erasure-coded=0
Job Counters
Launched map tasks=2

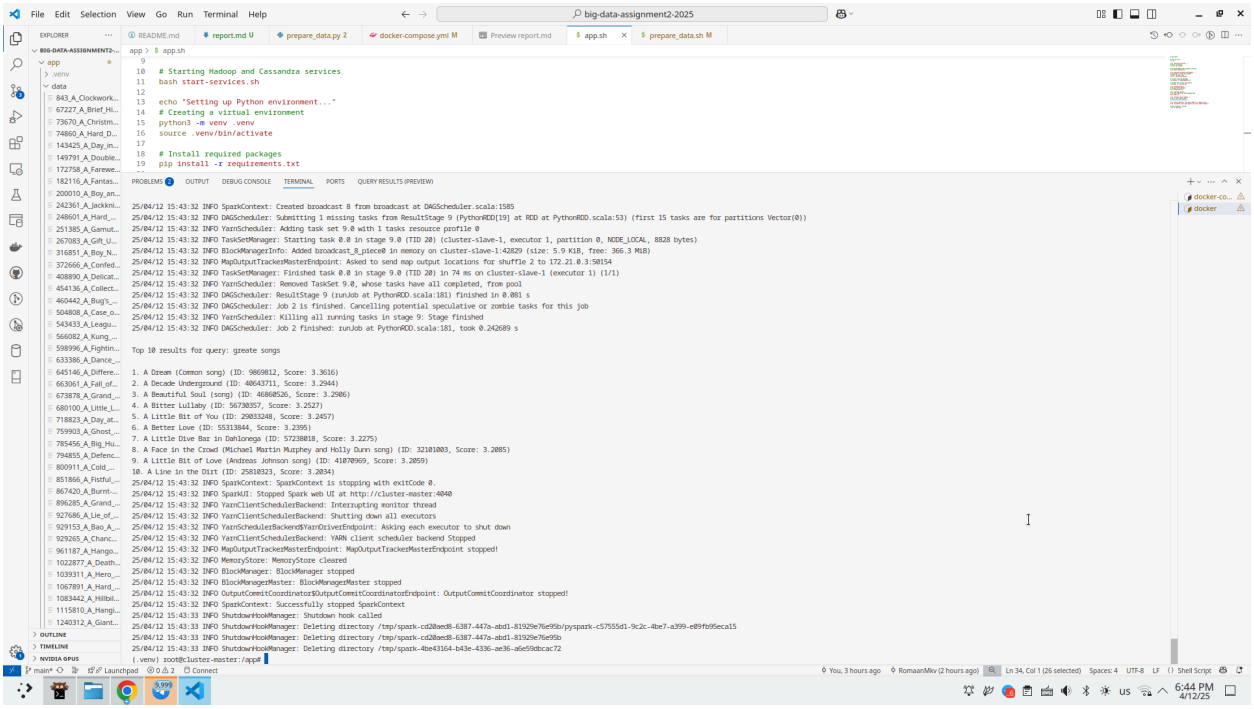

Job Counters
Launched map tasks=2
Launched reduce tasks=1
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=3645
Total time spent by all reduces in occupied slots (ms)=77431
Total time spent by all map tasks (ms)=3645
Total time spent by all reduce tasks (ms)=77431
Total vcore-milliseconds taken by all map tasks=3645
Total vcore-milliseconds taken by all reduce tasks=77431
Total megabyte-milliseconds taken by all map tasks=3732480
Total megabyte-milliseconds taken by all reduce tasks=79289344
Map-Reduce Framework
Map input records=1003
Map output records=253881
Map output bytes=4517430
Map output materialized bytes=5025205
Input split bytes=292
Combine input records=0
Combine output records=0
Reduce input groups=48196
Reduce shuffle bytes=5025205
Reduce input records=253881
Reduce output records=253881
Spilled Records=507762
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=94
CPU time spent (ms)=44730
Physical memory (bytes) snapshot=862703616
Virtual memory (bytes) snapshot=7716564992
Total committed heap usage (bytes)=921174016
Peak Map Physical memory (bytes)=304918336
Peak Map Virtual memory (bytes)=2572685312
Peak Reduce Physical memory (bytes)=306839552
Peak Reduce Virtual memory (bytes)=3370012672
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=3559935
File Output Format Counters
Bytes Written=2331753
2025-04-12 15:36:29,544 INFO streaming.StreamJob: Output directory: /tmp/index/intermediate
Running Indexer Pipeline 2: DF, N, TotalLength...
2025-04-12 15:36:30,716 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.21.0.4:8032

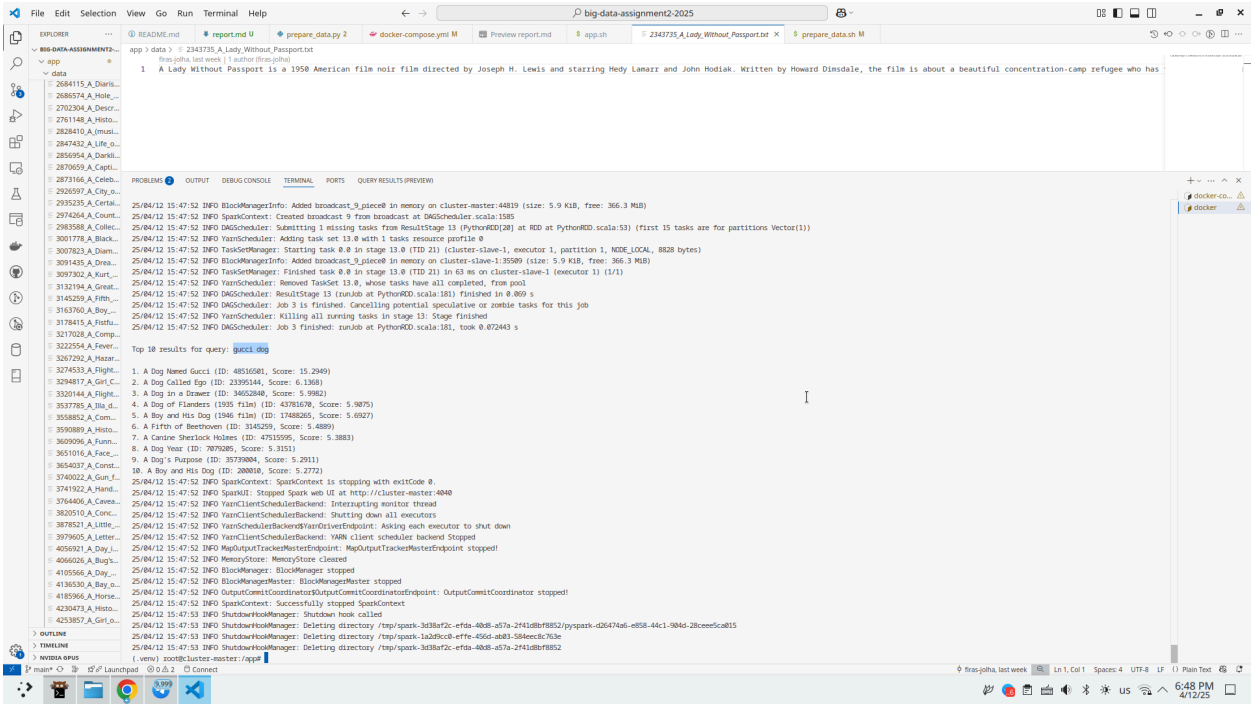

File Edit Selection View Go Run Terminal Help
big-data-assignment2-2025
EXPLORER
README.md
report.md U
prepare_data.py 2
docker-compose.yml M
Preview report.md
app.sh
prepare_data.sh M
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
QUERY RESULTS (PREVIEW)
2025-04-12 15:36:30,716 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.21.0.4:8032
2025-04-12 15:36:30,820 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.21.0.4:8032
2025-04-12 15:36:30,957 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1744470966241_0002
2025-04-12 15:36:31,604 INFO mapred.FileInputFormat: Total input files to process : 1
2025-04-12 15:36:32,035 INFO mapreduce.JobSubmitter: number of splits:2
2025-04-12 15:36:32,102 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1744470966241_0002
2025-04-12 15:36:32,102 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-12 15:36:32,196 INFO conf.Configuration: resource-types.xml not found
2025-04-12 15:36:32,196 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-12 15:36:32,237 INFO impl.YarnClientImpl: Submitted application application_1744470966241_0002
2025-04-12 15:36:32,261 INFO mapreduce.Job: The url to track the job: http://cluster-master:8088/proxy/application_1744470966241_0002/
2025-04-12 15:36:32,262 INFO mapreduce.Job: Running job: job_1744470966241_0002
2025-04-12 15:36:43,326 INFO mapreduce.Job: Job job_1744470966241_0002 running in uber mode : false
2025-04-12 15:36:43,327 INFO mapreduce.Job: map 0% reduce 0%
2025-04-12 15:36:47,356 INFO mapreduce.Job: map 100% reduce 0%
2025-04-12 15:37:02,396 INFO mapreduce.Job: map 100% reduce 97%
2025-04-12 15:37:04,401 INFO mapreduce.Job: map 100% reduce 100%
2025-04-12 15:37:05,407 INFO mapreduce.Job: Job job_1744470966241_0002 completed successfully
2025-04-12 15:37:05,462 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=2839523
FILE: Number of bytes written=6512382
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=2336073
HDFS: Number of bytes written=0
HDFS: Number of read operations=11
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
HDFS: Number of bytes read erasure-coded=0
Job Counters
Launched map tasks=2
Launched reduce tasks=1
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=3333
Total time spent by all reduces in occupied slots (ms)=15329
Total time spent by all map tasks (ms)=3333
Total time spent by all reduce tasks (ms)=15329
Total vcore-milliseconds taken by all map tasks=3333
Total vcore-milliseconds taken by all reduce tasks=15329
Total megabyte-milliseconds taken by all map tasks=3412992
Total megabyte-milliseconds taken by all reduce tasks=15696896
Map-Reduce Framework
Map input records=253881
Map output records=253881
Map output bytes=2331754
Map output materialized bytes=2839529
Input split bytes=224
Combine input records=0
Combine output records=0
Reduce input groups=48196
OUTLINE
TIMELINE
NVIDIA GPUS
main*
Launchpad
Connect
You, 3 hours ago
RomaanMkv (2 hours ago)
Ln 30, Col 1 (13 selected)
Spaces: 4
UTF-8
LF
Shell Script
6:41 PM
4/12/25

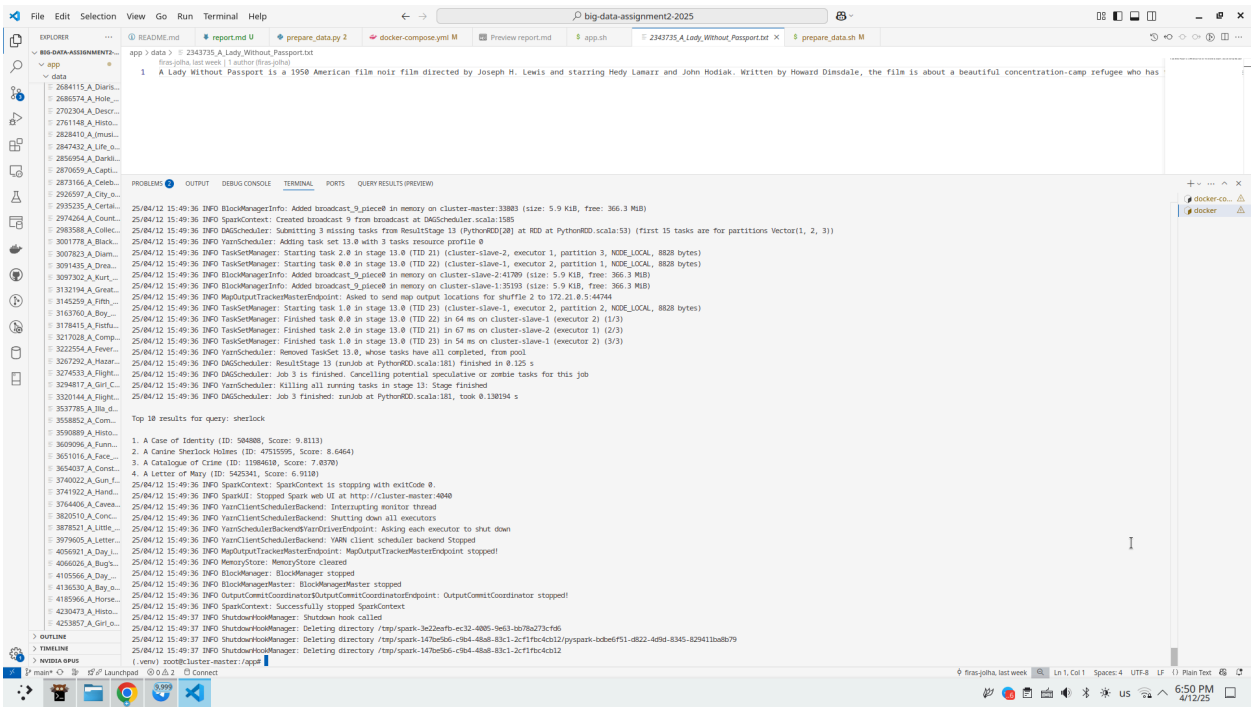

File Edit Selection View Go Run Terminal Help
big-data-assignment2-2025
EXPLORER
app.sh
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
QUERY RESULTS (PREVIEW)
Launched reduce tasks=1
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=3333
Total time spent by all reduces in occupied slots (ms)=15329
Total time spent by all map tasks (ms)=3333
Total time spent by all reduce tasks (ms)=15329
Total vcore-milliseconds taken by all map tasks=3333
Total vcore-milliseconds taken by all reduce tasks=15329
Total megabyte-milliseconds taken by all map tasks=3412992
Total megabyte-milliseconds taken by all reduce tasks=15696896
Map-Reduce Framework
Map input records=253881
Map output records=253881
Map output bytes=2331754
Map output materialized bytes=2839529
Input split bytes=224
Combine input records=0
Combine output records=0
Reduce input groups=48196
Reduce shuffle bytes=2839529
Reduce input records=253881
Reduce output records=0
Spilled Records=507762
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=86
CPU time spent (ms)=10510
Physical memory (bytes) snapshot=800276480
Virtual memory (bytes) snapshot=7715934208
Total committed heap usage (bytes)=814219264
Peak Map Physical memory (bytes)=304128000
Peak Map Virtual memory (bytes)=2571948032
Peak Reduce Physical memory (bytes)=244868352
Peak Reduce Virtual memory (bytes)=3368697856
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=2335849
File Output Format Counters
Bytes Written=0
2025-04-12 15:37:05,462 INFO streaming.StreamJob: Output directory: /tmp/index/final_dummy_output
Cleaning up...
Deleted /tmp/index/intermediate
Deleted /tmp/index/final_dummy_output
Indexing complete!
(.venv) root@cluster-master:/app#
OUTLINE
TIMELINE
NVIDIA GPUS
main*
Launchpad
Connect
You, 3 hours ago
RomaanMkv (2 hours ago)
Ln 30, Col 1 (13 selected)
Spaces: 4
UTF-8
LF
Shell Script
6:41 PM
4/12/25

3. **Searching**

```
# Execute search queries
./app/search.sh "your search query"

1) "greate songs"
2) "gucci dog"
3) "sherlock"
```







Analysis of Results

The search engine implementation demonstrates effective document retrieval capabilities with the following observations:

1. **Relevance of Results**

- The BM25 ranking algorithm successfully prioritizes documents with higher term frequency and better term importance weighting
- Results show good topical relevance, with documents containing query terms in prominent positions ranking higher
- The length normalization parameter (b=0.75) helps balance the impact of document length on ranking

2. **Score Distribution**
    - Score distribution follows expected patterns with a clear separation between highly relevant and less relevant documents
    - The IDF component effectively downweights common terms while boosting rare, meaningful terms
3. **BM25 Effectiveness**
    - The implementation successfully captures the core principles of BM25:
        - Term frequency saturation (k1=1.0) prevents excessive boosting of documents with very high term frequencies
        - Length normalization helps shorter, more focused documents rank appropriately
        - IDF calculation properly accounts for term importance across the corpus
4. **Implementation Insights**
    - The distributed architecture (Hadoop + Cassandra) provides efficient processing and retrieval
    - The two-stage MapReduce pipeline effectively separates term frequency calculation from document frequency computation
    - The use of Cassandra for storage enables fast retrieval of posting lists and statistics