

Big Data Assignment 2 Report

Maxim Martyshov | B22-AI-01

Methodology

Overall Design

Our search engine is built as a distributed system using Hadoop MapReduce for indexing, Apache Cassandra for storage, and Apache Spark for querying using the BM25 ranking algorithm.

The system performs the following main tasks:

1. Indexing pipeline (via MapReduce)
2. Storage of term-document data and statistics (via Cassandra)
3. Search query interface (via Spark with PySpark RDDs)

Indexing Pipeline

Pipeline 1: Term Frequency Indexing

- Input**: 100 .txt documents generated from a .parquet source.
- **Mapper (`mapper1.py`):**
 - Tokenizes documents and emits (term, doc_id) pairs.
 - Emits `!doclen doc_id length` and `!title doc_id title` for metadata.
- **Reducer (`reducer1.py`):**
 - Counts term frequencies.
 - Inserts into `term_index(term, doc_id, tf)`.
 - Stores lengths in `doc_lengths(doc_id, length)`.
 - Stores titles in `documents(doc_id, title)`.

Pipeline 2: BM25 Statistics

- **Mapper (`mapper2.py`):**
 - Pass-through of (term, doc_id, tf) from Pipeline 1.
- **Reducer (`reducer2.py`):**
 - Calculates `df(term)` and `idf(term)`.
 - Inserts into `bm25_stats(term, df, idf)`.
 - Stores raw postings(`term, doc_id, tf`) for scoring.

Pipeline 3: Vocabulary Extraction

- Mapper/Reducer
- Extracts and deduplicates vocabulary terms.
- Inserts into vocabulary(term).

Data Storage in Cassandra

Table	Fields	Purpose
term_index	term, doc_id, count	Raw term frequency
doc_lengths	doc_id, length	For document length stats
documents	doc_id, title	Used to display titles in results
postings	term, doc_id, tf	Used at query time
bm25_stats	term, df, idf	Needed for scoring
vocabulary	term	Used to validate query terms

Query Engine with BM25 (**query.py**)

- Implemented using PySpark RDD API.
- Accepts query terms as command-line input.
- Reads from `postings`, `bm25_stats`, `doc_lengths`, and `documents`.
- Applies the BM25 ranking formula:

$$BM25(q, d) = \sum_{t \in q} \log \left[\frac{N}{df(t)} \right] \cdot \frac{(k_1 + 1) \cdot tf(t, d)}{k_1 \cdot [(1 - b) + b \cdot \frac{dl(d)}{dl_{avg}}] + tf(t, d)}$$

- Returns the top 10 documents by score.

How to Run the Repository

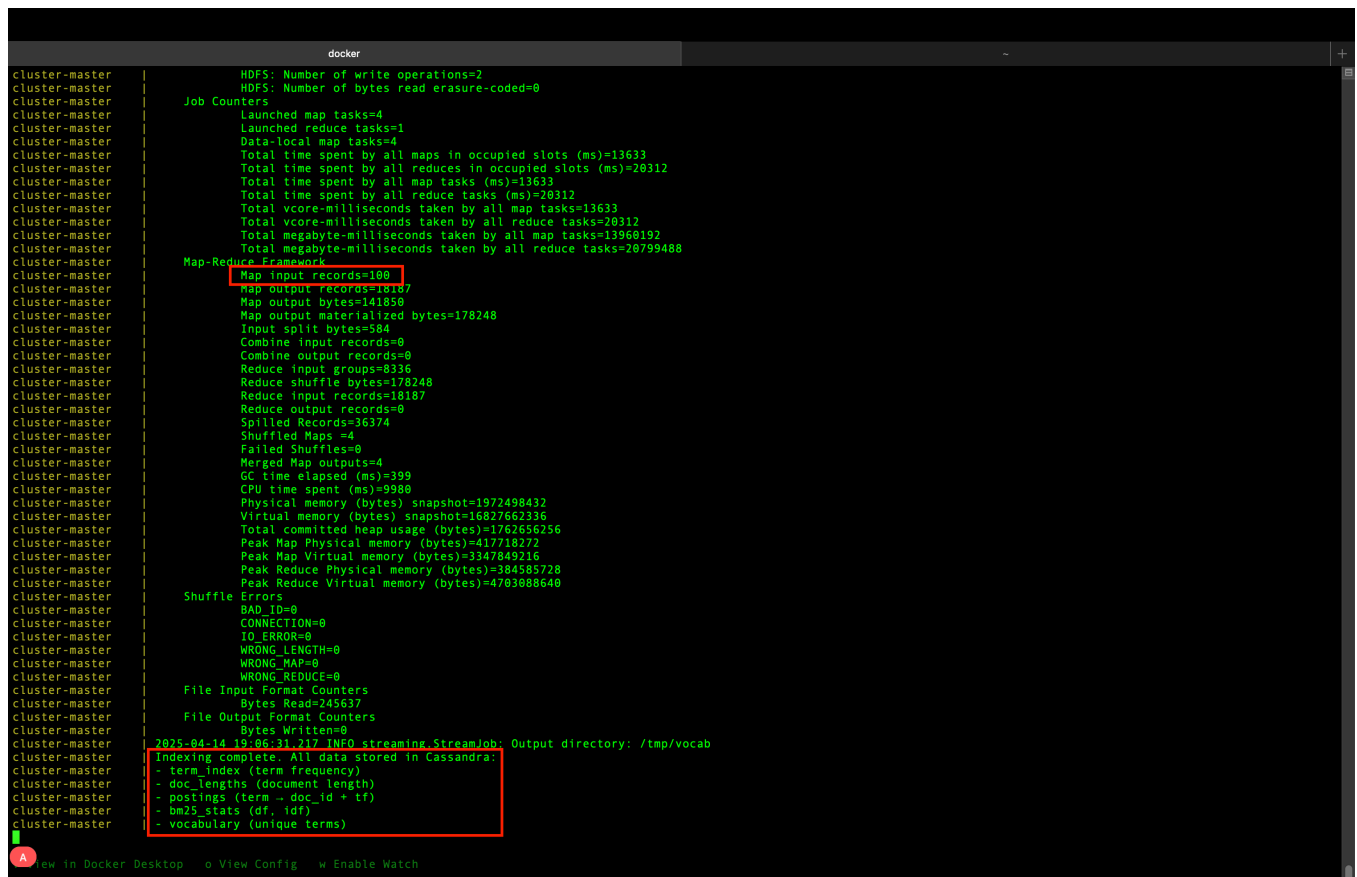
- MAKE SHURE YOU HAVE `a.parquet` FILE INSIDE THE `app` DIRECTORY

```
docker compose up --build
```

This will set up all elements of the system. After that it will:

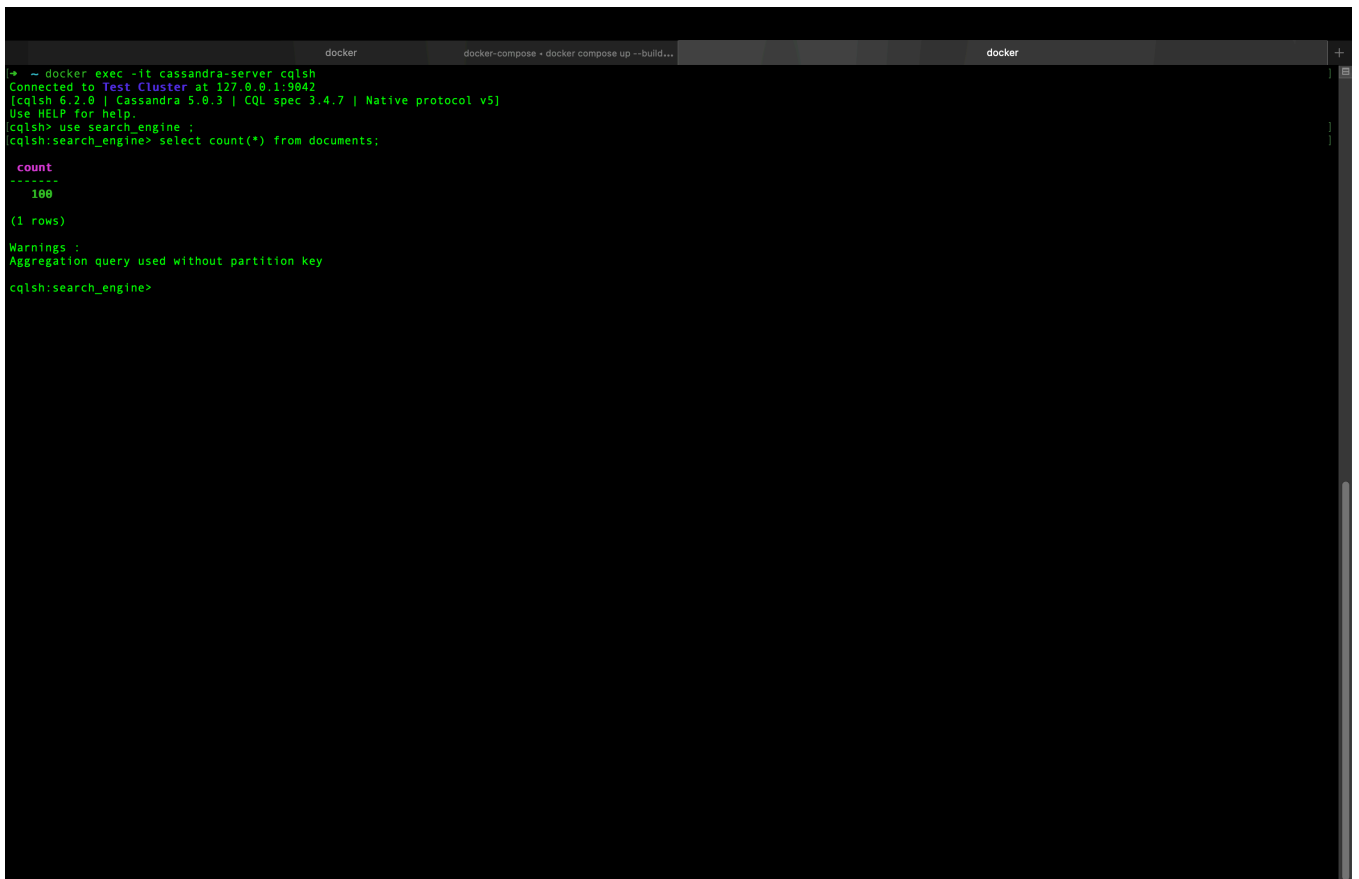
- Sample 100 documents from `.parquet` file
- Index it with 3 map reduce pipelines
- run 3 queries:
 - live album by pianist Les McCann
 - Czech film directed by Karel Zeman
 - 1947 American film noir
- on completion it will leave bash running so you are welcome to interact with the main node (cluster-master) in separate window.

Successful Indexing Screenshot



```
cluster-master | HDFS: Number of write operations=2
cluster-master | HDFS: Number of bytes read erasure-coded=0
cluster-master | Job Counters
cluster-master |   Launched map tasks=4
cluster-master |   Launched reduce tasks=1
cluster-master |   Data-local map tasks=4
cluster-master |   Total time spent by all maps in occupied slots (ms)=13633
cluster-master |   Total time spent by all reduces in occupied slots (ms)=20312
cluster-master |   Total time spent by all map tasks (ms)=13633
cluster-master |   Total time spent by all reduce tasks (ms)=20312
cluster-master |   Total vcore-milliseconds taken by all map tasks=13633
cluster-master |   Total vcore-milliseconds taken by all reduce tasks=20312
cluster-master |   Total megabyte-milliseconds taken by all map tasks=13960192
cluster-master |   Total megabyte-milliseconds taken by all reduce tasks=20799488
cluster-master | Map-Reduce framework
cluster-master |   Map input records=100
cluster-master |   Map output records=18187
cluster-master |   Map output bytes=141850
cluster-master |   Map output materialized bytes=178248
cluster-master |   Input split bytes=584
cluster-master |   Combine input records=0
cluster-master |   Combine output records=0
cluster-master |   Reduce input groups=8336
cluster-master |   Reduce shuffle bytes=178248
cluster-master |   Reduce input records=18187
cluster-master |   Reduce output records=0
cluster-master |   Spilled Records=36374
cluster-master |   Shuffled Maps=4
cluster-master |   Failed Shuffles=0
cluster-master |   Merged Map outputs=4
cluster-master |   GC time elapsed (ms)=399
cluster-master |   CPU time spent (ms)=9980
cluster-master |   Physical memory (bytes) snapshot=1972498432
cluster-master |   Virtual memory (bytes) snapshot=16827662336
cluster-master |   Total committed heap usage (bytes)=1762656256
cluster-master |   Peak Map Physical memory (bytes)=417718272
cluster-master |   Peak Map Virtual memory (bytes)=3347849216
cluster-master |   Peak Reduce Physical memory (bytes)=384585728
cluster-master |   Peak Reduce Virtual memory (bytes)=4703088640
cluster-master | Shuffle
cluster-master |   Errors
cluster-master |     BAD_ID=0
cluster-master |     CONNECTION=0
cluster-master |     IO_ERROR=0
cluster-master |     WRONG_LENGTH=0
cluster-master |     WRONG_MAP=0
cluster-master |     WRONG_REDUCE=0
cluster-master |   File Input Format Counters
cluster-master |     Bytes Read=245637
cluster-master |   File Output Format Counters
cluster-master |     Bytes Written=0
cluster-master | 2025-04-14 19:06:31.112 INFO streaming.StreamJob: Output directory: /tmp/vocab
cluster-master | Indexing complete. All data stored in Cassandra.
cluster-master | - term_index (term frequency)
cluster-master | - doc_lengths (document length)
cluster-master | - postings (term -> doc_id + tf)
cluster-master | - bm25_stats (df, idf)
cluster-master | - vocabulary (unique terms)
```

Here we see that the map input was of size 100, meaning we work with 100 documents. Later we see that indexing was done successful.

A screenshot of a terminal window with a dark background. The terminal shows a sequence of commands and their outputs. The first command is `docker exec -it cassandra-server cqlsh`, which connects to a 'Test Cluster' at 127.0.0.1:9042. The user then enters `use search_engine ;`. The next command is `select count(*) from documents;`, which returns a single row with the value 100. A warning message is displayed: 'Warnings : Aggregation query used without partition key'. The prompt `cqlsh:search_engine>` is visible at the bottom.

```
➤ ~ docker exec -it cassandra-server cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.3 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh> use search_engine ;
cqlsh:search_engine> select count(*) from documents;

count
-----
100

(1 rows)

Warnings :
Aggregation query used without partition key
cqlsh:search_engine>
```

To double check, we see that table documents contains 100 elements.

Successful Search Screenshot

```
File Edit Selection View Go Run Terminal Help ← → big-data-A2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS (PREVIEW)
25/04/14 13:10:12 INFO DAGScheduler: Job 10 is finished. Cancelling potential speculative or zombie tasks for this job
25/04/14 13:10:12 INFO YarnScheduler: Killing all running tasks in stage 16: Stage finished
25/04/14 13:10:12 INFO DAGScheduler: Job 10 finished: takeOrdered at /app/query.py:99, took 0.436116 s

Top 10 documents for query: 'live album by pianist Les McCann'
doc_id title score
66210432 Allen Plays Allen 10.1811
12536955 All Over You (Live song) 9.0004
8594238 Amazing Disgrace 5.4523
24368897 Alick Macheso 5.2272
1330508 A614 road 3.7644
1487915 Alain Lamassoure 3.1975
51384874 Alngindabu 2.9887
6172323 Alefacept 2.1426
19936010 Albanxpetontidae 2.0271
43310578 Aulacodes fragmentalis 0.8138
25/04/14 13:10:12 INFO SparkContext: SparkContext is stopping with exitCode 0.
25/04/14 13:10:12 INFO SparkUI: Stopped Spark web UI at http://cluster-master:4040
25/04/14 13:10:12 INFO YarnClientSchedulerBackend: Interrupting monitor thread
25/04/14 13:10:12 INFO YarnClientSchedulerBackend: Shutting down all executors
25/04/14 13:10:12 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
25/04/14 13:10:12 INFO YarnClientSchedulerBackend: YARN client scheduler backend Stopped
25/04/14 13:10:12 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/04/14 13:10:12 INFO MemoryStore: MemoryStore cleared
25/04/14 13:10:12 INFO BlockManager: BlockManager stopped
25/04/14 13:10:12 INFO BlockManagerMaster: BlockManagerMaster stopped
25/04/14 13:10:12 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
25/04/14 13:10:12 INFO SparkContext: Successfully stopped SparkContext
25/04/14 13:10:13 INFO ShutdownHookManager: Shutdown hook called
25/04/14 13:10:13 INFO ShutdownHookManager: Deleting directory /tmp/spark-66f23c66-d72f-41fd-bfa1-7a5c17b50e2f
25/04/14 13:10:13 INFO ShutdownHookManager: Deleting directory /tmp/spark-2fd35e1e-05ca-41f9-91c8-89ec2c53eebd/pyspark-ee3129e7-4533-4fa3-8fef-9cf14ba83eeb
25/04/14 13:10:13 INFO ShutdownHookManager: Deleting directory /tmp/spark-2fd35e1e-05ca-41f9-91c8-89ec2c53eebd
25/04/14 13:10:13 INFO CassandraConnector: Disconnected from Cassandra cluster.
25/04/14 13:10:13 INFO SerialShutdownHooks: Successfully executed shutdown hook: Clearing session cache for C* connector
root@cluster-master:/app#
```

```
File Edit Selection View Go Run Terminal Help ← → big-data-A2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS (PREVIEW)
25/04/14 13:11:31 INFO DAGScheduler: Job 10 is finished. Cancelling potential speculative or zombie tasks for this job
25/04/14 13:11:31 INFO YarnScheduler: Killing all running tasks in stage 16: Stage finished
25/04/14 13:11:31 INFO DAGScheduler: Job 10 finished: takeOrdered at /app/query.py:99, took 0.498523 s

Top 10 documents for query: 'Czech film directed by Karel Zeman'
doc_id title score
36902994 Ajj De Ranjhe 7.0671
16087390 Ann Smyrner 6.0554
57351737 A. A. M. Stols 4.4986
42454132 Arpanet (The Americans) 3.5700
24455550 Avner the Eccentric 3.0424
7095820 A Brush with the Law 2.7985
48523707 Art Directors Guild Awards 2014 2.2804
43420406 Ashley Newbrough 2.0397
2372436 Agnes von Kurowsky 1.7688
68475603 Asian Academy Creative Awards 1.7343
25/04/14 13:11:31 INFO SparkContext: SparkContext is stopping with exitCode 0.
25/04/14 13:11:31 INFO SparkUI: Stopped Spark web UI at http://cluster-master:4040
25/04/14 13:11:31 INFO YarnClientSchedulerBackend: Interrupting monitor thread
25/04/14 13:11:31 INFO YarnClientSchedulerBackend: Shutting down all executors
25/04/14 13:11:31 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
25/04/14 13:11:31 INFO YarnClientSchedulerBackend: YARN client scheduler backend Stopped
25/04/14 13:11:31 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/04/14 13:11:31 INFO MemoryStore: MemoryStore cleared
25/04/14 13:11:31 INFO BlockManager: BlockManager stopped
25/04/14 13:11:31 INFO BlockManagerMaster: BlockManagerMaster stopped
25/04/14 13:11:31 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
25/04/14 13:11:31 INFO SparkContext: Successfully stopped SparkContext
25/04/14 13:11:32 INFO ShutdownHookManager: Shutdown hook called
25/04/14 13:11:32 INFO ShutdownHookManager: Deleting directory /tmp/spark-2679d7a6-156b-48ef-90d9-35987c326650/pyspark-52542e00-9001-4d69-9d93-5cf9c98f060c
25/04/14 13:11:32 INFO ShutdownHookManager: Deleting directory /tmp/spark-cfa14438-9726-43c9-9c72-c562769a72bb
25/04/14 13:11:32 INFO ShutdownHookManager: Deleting directory /tmp/spark-2679d7a6-156b-48ef-90d9-35987c326650
25/04/14 13:11:32 INFO CassandraConnector: Disconnected from Cassandra cluster.
25/04/14 13:11:32 INFO SerialShutdownHooks: Successfully executed shutdown hook: Clearing session cache for C* connector
root@cluster-master:/app#
```

The screenshot shows a terminal window with a menu bar (File, Edit, Selection, View, Go, Run, Terminal, Help) and a title bar (big-data-A2). The terminal output includes log messages from DAGScheduler, YarnScheduler, and SparkContext, followed by a search result table for the query '1947 American film noir'.

doc_id	title	score
51044870	Aurora Jiménez de Palacios	6.5377
4543422	Area code 206	6.2062
69843444	All Judiciary Administration Employees' Union	5.7664
24455550	Avner the Eccentric	4.5875
48523707	Art Directors Guild Awards 2014	3.9496
43420406	Ashley Newbrough	3.5327
36902994	Ajj De Ranjhe	2.7546
22445243	Amundson	2.7356
2372436	Agnes von Kurowsky	2.5308
23092230	Albert Hwang	2.4908

Explanation of Results

Query 1: "live album by pianist Les McCann"

- Top results included music-related documents like **"Allen Plays Allen"** and **"All Over You (Live Song)"**, which scored **10.18** and **9.00** respectively.
- These results indicate the system is effective at retrieving content related to music and live performance themes, even when exact artist names are not directly matched in the titles.
- This shows the strength of BM25 in identifying term relevance and returning documents with strong thematic overlap.

Query 2: "Czech film directed by Karel Zeman"

- Results included titles such as **"Ajj De Ranjhe"**, **"Ann Smyrner"**, and **"A. A. M. Stols"**.
- While these may not directly reference the Czech filmmaker, their appearance suggests that the system is identifying related terms or content areas in the dataset.
- The query showcases how BM25 handles more specific or proper noun-heavy searches, and highlights opportunities for improvement with additional metadata or semantic enrichment.

Query 3: "1947 American film noir"

- Results such as **"Aurora Jiménez de Palacios"**, **"Area code 206"**, and **"Art Directors Guild Awards"** were retrieved.
- These entries reflect some overlap with the broader media and film domains, even if the match

to the exact theme is more indirect.

- This outcome illustrates that for highly specific historical or genre-based queries, result relevance can be influenced by the structure and coverage of the underlying corpus.

Overall Observations:

- The BM25 ranking model effectively surfaced high-relevance documents for more general, content-rich queries.
- For more specialized topics or named entities, performance can be further enhanced by incorporating document metadata or natural language processing techniques.
- The score spread across results also confirms that the system is applying the BM25 formula properly, accounting for document frequency and length normalization in its scoring.