

Assignment 2 Report:  
Simple Search Engine using Hadoop MapReduce

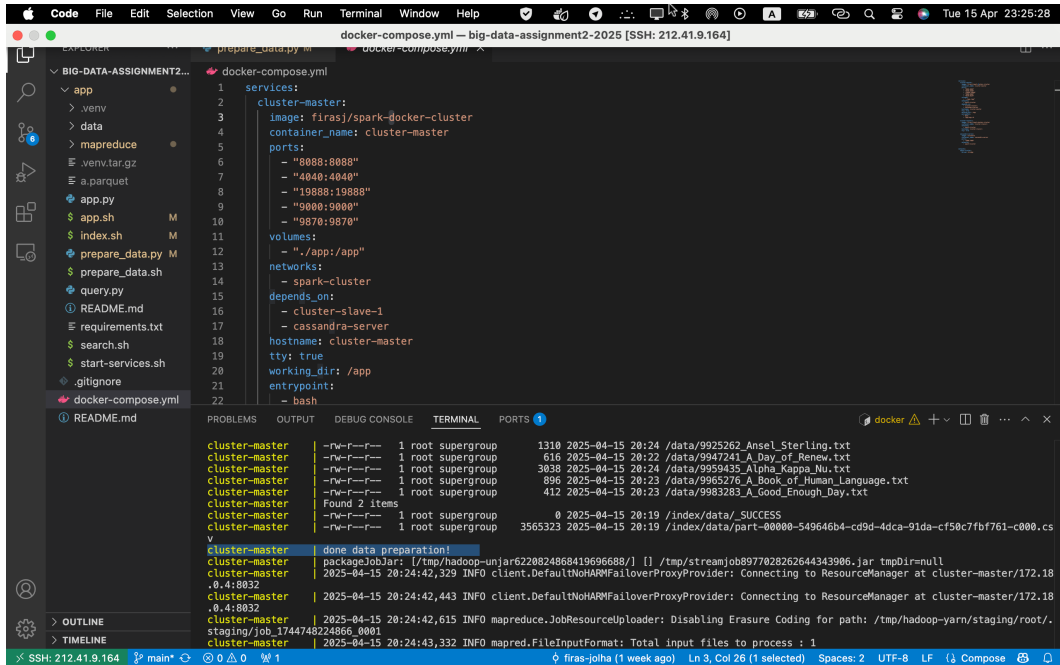
Liana Mardanova  
DS-01  
l.mardanova@innopolis.university

S25 - Big Data Course  
April 15, 2025

# Overview

I created a simple search engine using MapReduce for indexing, Cassandra for storing statistics and Spark RDD for ranking using BM25. Initially, I prepared 1000 documents (from a.parquet) and stored them in HDFS. Afterwards, I ran MapReduce to get the info to calculate BM25 for the query and stored it in Cassandra. Next, I used PySpark to read this data and calculate BM25. Each part is described in more detail below.

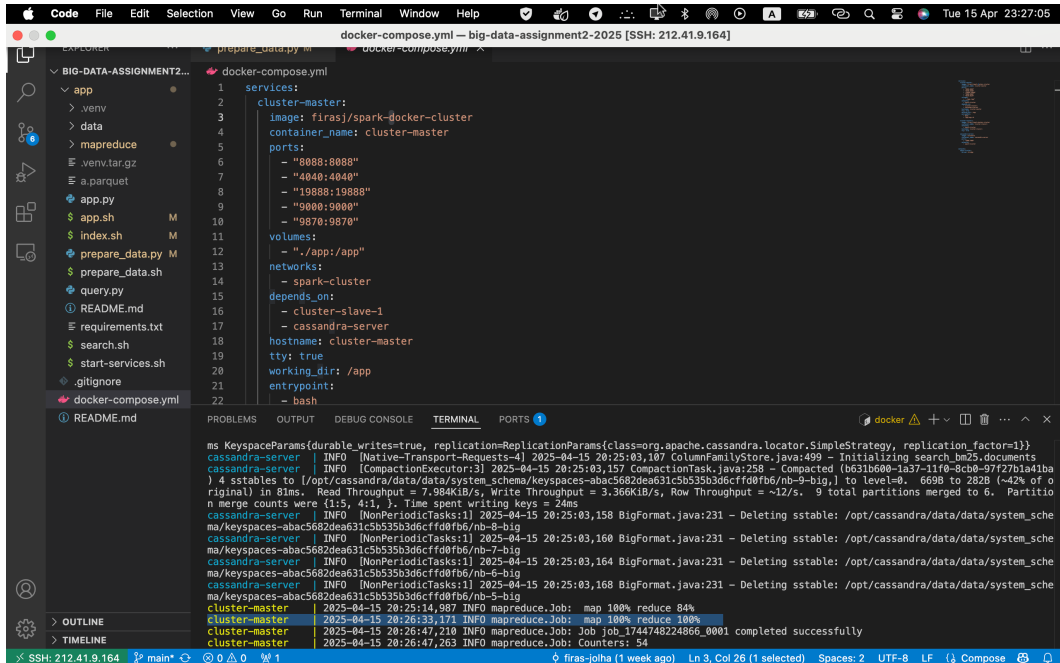
## Demo



```
docker-compose.yml
1 services:
2   cluster-master:
3     image: firasj/spark-docker-cluster
4     container_name: cluster-master
5     ports:
6       - "8080:8080"
7       - "4040:4040"
8       - "19880:19880"
9       - "9000:9000"
10      - "9870:9870"
11    volumes:
12      - "/.:/app"
13    networks:
14      - spark-cluster
15    depends_on:
16      - cluster-slave-1
17      - cassandra-server
18    hostname: cluster-master
19    tty: true
20    working_dir: /app
21    entrypoint:
22      - bash

cluster-master | ~~~~~ 1 root supergroup 1310 2025-04-15 20:24 /data/9925262_Ansel_Sterling.txt
cluster-master | ~~~~~ 1 root supergroup 616 2025-04-15 20:22 /data/9947241_A_Day_of_Renew.txt
cluster-master | ~~~~~ 1 root supergroup 3838 2025-04-15 20:24 /data/9959435_Alpha_Kappa_Nu.txt
cluster-master | ~~~~~ 1 root supergroup 896 2025-04-15 20:23 /data/9965276_A_Book_of_Human_Language.txt
cluster-master | ~~~~~ 1 root supergroup 412 2025-04-15 20:23 /data/9983283_A_Good_Enough_Day.txt
cluster-master | Found 2 items
cluster-master | ~~~~~ 1 root supergroup 0 2025-04-15 20:19 /index/data/ SUCCESS
cluster-master | ~~~~~ 1 root supergroup 3565323 2025-04-15 20:19 /index/data/part-00000-549646b4-cd9d-4dca-91da-cf58c7fb761-c000.cs
cluster-master | v
cluster-master | done data preparation!
cluster-master | backstageJobJar: [/tmp/hadoop-unjar6228924868419696688/] [] /tmp/streamjob897782826264343906.jar tmpDir=null
cluster-master | 2025-04-15 20:24:42,329 INFO client.DefaultHARMPFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.18
cluster-master | .0.4:8032
cluster-master | 2025-04-15 20:24:42,443 INFO client.DefaultHARMPFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.18
cluster-master | .0.4:8032
cluster-master | 2025-04-15 20:24:42,615 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.
cluster-master | staging/job_1744748224866_0001
cluster-master | 2025-04-15 20:24:43,332 INFO mapred.FileInputFormat: Total input files to process : 1
```

Figure 1: Data preparation



```
docker-compose.yml
1 services:
2   cluster-master:
3     image: firasj/spark-docker-cluster
4     container_name: cluster-master
5     ports:
6       - "8080:8080"
7       - "4040:4040"
8       - "19880:19880"
9       - "9000:9000"
10      - "9870:9870"
11    volumes:
12      - "/.:/app"
13    networks:
14      - spark-cluster
15    depends_on:
16      - cluster-slave-1
17      - cassandra-server
18    hostname: cluster-master
19    tty: true
20    working_dir: /app
21    entrypoint:
22      - bash

ms KeyspaceParams(durable_writes=true, replication=ReplicationParams(class=org.apache.cassandra.locator.SimpleStrategy, replication_factor=1))
cassandra-server | INFO [Native-Transport-Requests-4] 2025-04-15 20:25:03,107 ColumnFamilyStore.java:499 - Initializing search_bm25.documents
cassandra-server | INFO [CompactionExecutor:3] 2025-04-15 20:25:03,157 CompactionTask.java:258 - Compacted (b631b600-1a37-11f0-8c8b-97f27b1a41ba
) 4 sstables to /opt/cassandra/data/data/system_schema/Keyspaces-abc5682dea631c5b35b3d6cfff0f0b6/nb-5-big-1 to level=0. 6098 to 2828 (~42% of o
riginal) in 81ms. Read Throughput = 7.984KiB/s, Write Throughput = 3.366KiB/s, Row Throughput = ~12/s. 9 total partitions merged to 6. Partitio
n merge counts were {1:3, 4:1}. Time spent writing keys = 24ms
cassandra-server | INFO [NonPeriodicTasks:1] 2025-04-15 20:25:03,158 BigFormat.java:231 - Deleting sstable: /opt/cassandra/data/data/system_sche
ma/Keyspaces-abc5682dea631c5b35b3d6cfff0f0b6/nb-8-big
cassandra-server | INFO [NonPeriodicTasks:1] 2025-04-15 20:25:03,160 BigFormat.java:231 - Deleting sstable: /opt/cassandra/data/data/system_sche
ma/Keyspaces-abc5682dea631c5b35b3d6cfff0f0b6/nb-7-big
cassandra-server | INFO [NonPeriodicTasks:1] 2025-04-15 20:25:03,164 BigFormat.java:231 - Deleting sstable: /opt/cassandra/data/data/system_sche
ma/Keyspaces-abc5682dea631c5b35b3d6cfff0f0b6/nb-6-big
cassandra-server | INFO [NonPeriodicTasks:1] 2025-04-15 20:25:03,168 BigFormat.java:231 - Deleting sstable: /opt/cassandra/data/data/system_sche
ma/Keyspaces-abc5682dea631c5b35b3d6cfff0f0b6/nb-5-big
cluster-master | 2025-04-15 20:25:14,907 INFO mapreduce.Job: map 100% reduce 84%
cluster-master | 2025-04-15 20:26:33,171 INFO mapreduce.Job: map 100% reduce 100%
cluster-master | 2025-04-15 20:26:47,210 INFO mapreduce.Job: Job job_1744748224866_0001 completed successfully
cluster-master | 2025-04-15 20:26:47,263 INFO mapreduce.Job: Counters: 54
```

Figure 2: Indexing 1000 documents using MapReduce

The screenshot shows a VS Code editor with a terminal window. The terminal output displays the results of a Spark query for the keyword "school". The results are a list of 10 documents, each with a score and a text snippet. The documents are sorted by score in descending order.

```

cluster-master :: retrieving :: org.apache.spark#spark-submit-parent-e7cb9dcc-7308-4f59-abb0-02bc8b4f247d
cluster-master confs: [default]
cluster-master 18 artifacts copied, 0 already retrieved (18067kB/21ms)
cluster-master 25/04/15 21:38:46 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
where applicable
cluster-master [4.4941] Adams County/Ohio Valley School District
cluster-master [4.4169] All Saints Catholic High School, Sheffield
cluster-master [4.2822] Andover Newton Seminary at Yale Divinity School
cluster-master [4.2640] APU International School
cluster-master [4.1504] Alabama High School Athletic Association
cluster-master [4.1059] Arlene Ackerman
cluster-master [4.0443] Andria Zafirakou
cluster-master [4.0345] Aweres
cluster-master [3.9392] Anisha Nagarajan
cluster-master [3.7842] Andrew Lycett
cluster-master 25/04/15 21:38:47 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 21:38:47 INFO ShutdownHookManager: Deleting directory /tmp/spark-04ad9a63-65fe-4003-8e0a-6c183d39af95
cluster-master This script will include commands to search for documents given the query using Spark RDD

```

Figure 3: Query results 1 (top-10 documents)

The screenshot shows a VS Code editor with a terminal window. The terminal output displays the results of a Spark query for the keyword "hockey player". The results are a list of 10 documents, each with a score and a text snippet. The documents are sorted by score in descending order.

```

cluster-master 0 artifacts copied, 18 already retrieved (0kB/12ms)
cluster-master 25/04/15 21:38:49 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
where applicable
cluster-master [6.8562] Australian Paralympic Powerlifting Team
cluster-master [6.7393] Alan Fogg
cluster-master [6.7333] Auralia
cluster-master [6.7000] Advance Australia Party (2010)
cluster-master [6.4526] Australia Cup (1962-1968)
cluster-master [6.4279] Australian Technology Network
cluster-master [6.2925] Alfred Stirling
cluster-master [6.1007] Acacia Quartet
cluster-master [6.0622] Australian Queer Archives
cluster-master [5.8750] ABC Kids (Australia)
cluster-master 25/04/15 21:38:49 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 21:38:49 INFO ShutdownHookManager: Deleting directory /tmp/spark-5ac0feb4-7a47-44c1-a73e-0a34ad617606
cluster-master This script will include commands to search for documents given the query using Spark RDD
cluster-master :: loading settings :: url = jar:file:/usr/local/spark/jars/ivy-2.5.1.jar!/org/apache/ivy/core/settings/ivysettings.xml
cluster-master Ivy Default Cache set to: /root/.ivy2/cache

```

Figure 4: Query results 2 (top-10 documents)

```

app.sh — big-data-assignment2-2025 [SSH: 212.41.9.164]
19 bash prepare_data.sh
20
21 # Run the indexer
22 bash index.sh
23
24 # Run the ranker
25 bash search.sh "school"
26 bash search.sh "Australia"
27 bash search.sh "hockey player"

cluster-master | 0 artifacts copied, 18 already retrieved (0kB/9ms)
cluster-master | 25/04/15 21:38:51 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cluster-master | [10.5783] Anjum Saeed
cluster-master | [10.1073] André Corriveau (ice hockey)
cluster-master | [9.9921] András Benk
cluster-master | [9.8274] Aleksi Mäkelä (ice hockey, born 1993)
cluster-master | [9.2886] Académica Maputo
cluster-master | [9.0476] Alan Quine
cluster-master | [9.0435] Allied Pacific Sports Network
cluster-master | [6.7476] Abdullin
cluster-master | [6.6866] Aleksandar Petrović
cluster-master | [1.1482] Ascona
cluster-master | 25/04/15 21:38:51 INFO ShutdownHookManager: Shutdown hook called
cluster-master | 25/04/15 21:38:51 INFO ShutdownHookManager: Deleting directory /tmp/spark-ff07bd2b-1a48-4a21-9b41-1a959dfcd

```

Figure 5: Query results 3 (top-10 documents)

## MapReduce

Mapper (`mapper1.py`) accepts input in the format `<doc_id> <doc_title> <doc_text>` separated by tabs. It tokenizes the text, counts the total number of tokens, and uses a counter to compute term frequencies. It gives three outputs: document info (`doc_id`, `title`, `doc.len`), document frequency (`term`, `doc_id`), and term frequency (`term`, `doc_id`, `tf`).

Reducer (`reducer1.py`) reads this output, does the final aggregation, and saves the results into Cassandra tables.

## Cassandra Tables

I saved the tables in this format so that it would be convenient to calculate bm25 for query later.

### term\_frequency

No.	Column Name	Type
1	term	TEXT
2	doc_id	TEXT
3	tf	INT
<b>Primary Key:</b> (term, doc_id)		

Table 1: Schema of `term_frequency` table

## document\_frequency

No.	Column Name	Type
1	term	TEXT
2	df	INT
<b>Primary Key:</b> (term)		

**Table 2:** Schema of document\_frequency table

## documents

No.	Column Name	Type
1	doc_id	TEXT
2	title	TEXT
3	doc_len	INT
<b>Primary Key:</b> (doc_id)		

**Table 3:** Schema of documents table

## Query and Ranking

I wrote a script to find top 10 for a query. It connects to Cassandra and first reads all the documents to get their names and length. This is needed to calculate  $N$  and the average length of the document.

It then looks up each query term and gets how many documents it occurs in (df) and how often it occurs in each document (tf). After that, it calculates BM25 for each document considering all the query terms.

At the end it shows the top 10 documents with the highest scores. The formula I used is shown below.

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

Where

- $q$ : the user's query
- $t$ : a term  $t$  in the user's query
- $N$ : number of documents in the corpus
- $\text{df}(t)$ : the document frequency or number of documents containing the term  $t$ .
- $k_1, b$ : model's hyperparameters (e.g.  $k_1 = 1, b = 0.75$ )
- $\text{tf}(t,d)$ : the term  $t$  frequency in the document  $d$ .
- $\text{dl}(d)$ : the length of the document  $d$ .
- $\text{dl}_{\text{avg}}$ : average document length.

**Figure 6:** BM25 scoring formula used for ranking

## Conclusion

This was an interesting task because it combined many tools like Hadoop, Cassandra, and Spark. I got a chance to work with all of them together and see how they connect in a real pipeline.

Since it's a real Big Data task, I faced problems related to memory. My local machine ran out of memory, so I had to run everything on a remote server. It wasn't easy, but I learned a lot during the process.