Ivan Belstov i.beltsov@innopolis.university

Methodology

Data Preparation

There are 3 files responsible for data preparation: prepare_data.sh, prepare_data.py and prepare_index_data.py

prepare_data.sh transfers a.parquet from local to hdfs, starts prepare_data.py, copies newly made data folder to hdfs and starts prepare_index_data.py

prepare_data.py is unchanged from initial file

prepare_index_data.py reads files from hdfs data folder and creates a new partition using rdd

Indexer tasks

index.sh – Pipeline Runner Script that orchestrates the full indexing pipeline.

Actions:

- Accepts input path (local or HDFS) and uploads local files if needed
- Runs first MapReduce job to compute raw term frequencies
- Runs second MapReduce job to generate document stats and build the inverted index
- Triggers the Python script (app.py) to load results into Cassandra

MapReduce Stage 1

mapper1.py – Tokenizes document text, emits one entry per word occurrence.
Takes tab-separated lines with doc_id, title, text as input.
Returns lines with doc_id, term, 1, and title.

reducer1.py – Sums word counts per document to get term frequency.
Takes sorted output from mapper1.py.
Returns aggregated doc_id, term, tf, title.

MapReduce Stage 2

mapper2.py – Computes total document length and prepares structured input for indexing.
Takes output from reducer1 and returns:
DOCLEN lines with total doc length and title.
TERM lines with individual term frequencies per document.

reducer2.py – Groups all documents per term, computes document frequency (df) per term aand finalizes all index entries.
Takes output from mapper2.

Output:
VOCAB entries: term → document frequency (df).
INDEX entries: term → document ID, tf.
DOCLEN entries: document ID → length, title.

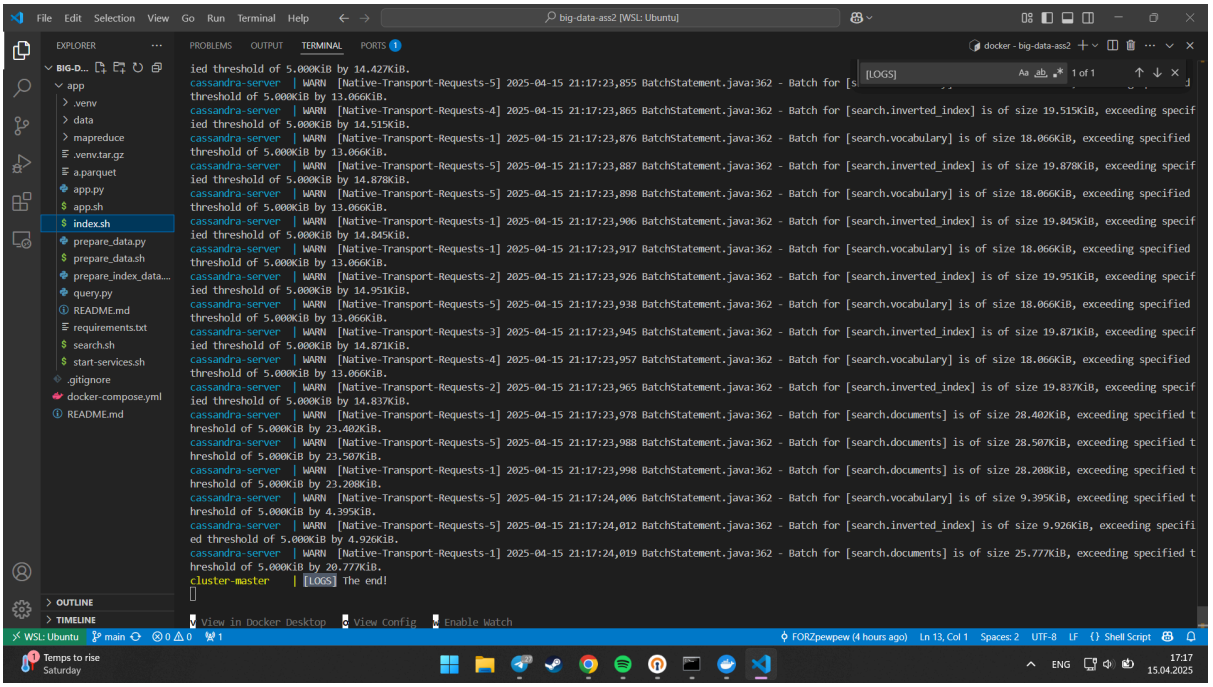
app.py – Loads structured index output into Cassandra tables
Functions:
fetch_hdfs_output(): Reads reducer2 output from HDFS.
connect_cassandra(): Connects to Cassandra and ensures keyspace exists.
ensure_tables(): Creates tables for vocabulary, inverted_index, and documents.
parse_and_insert(): Parses output and performs batch inserts into Cassandra.
Stored Tables:
vocabulary(term, df)
inverted_index(term, doc_id, tf)
documents(doc_id, length, title)

Search tasks

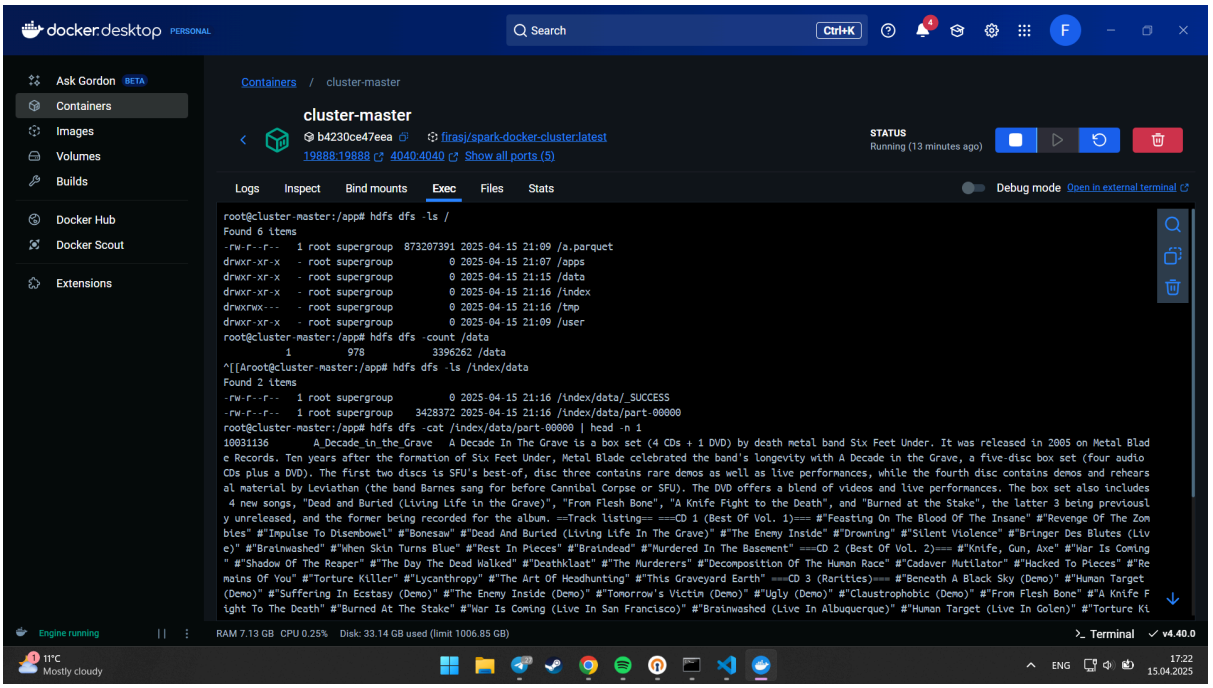
For some reason docker crashes during search query (though it shouldn't), so i commented out search.sh. Basically the whole task is missing :(

Demonstration

Warn messages from cassandra due to batch size is higher than threshold by a couple of KiB(but files are still added to cassandra)



This screenshot shows contents of hdfs / folder, number of documents in /data folder which is 978, contents of /index/data folder and the first line of /index/data/part-00000



Next screenshots show cassandra db tables, their contents and count of rows (global count is expensive without partitions, but there’s no workaround :c)

pewpew@pewpew: /mnt/c/Users/gachi

```
cqlsh:search> describe tables;

documents  inverted_index  vocabulary

cqlsh:search> select count(*) from documents;

 count
-------
   978

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:search> select * from documents limit 5;

 doc_id   | length | title
----------+--------+-----------------------------------------
 10230685 |    115 |                        A_Dead_Sinking_Story
 27568194 |    567 | A_Hero_Ain't_Nothin'_but_a_Sandwich_(film)
 39710446 |    330 |                         A_Little_Bit_of_Luck
 38294693 |    338 | A_Change_Is_Gonna_Come_(Jack_McDuff_album)
 51794980 |    306 |     A_Family_Secret_(Upstairs,_Downstairs)

(5 rows)
cqlsh:search> select count(*) from inverted_index;

 count
--------
 244375

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:search>
```

11°C Mostly cloudy ENG 17:27 15.04.2025

pewpew@pewpew: /mnt/c/Users/gachi

```
(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:search> select * from inverted_index limit 5;

 term   | doc_id   | tf
--------+----------+----
 dobson | 13633480 |  1
 bessus | 12000397 |  3
     ix | 19789501 |  1
     ix | 32497421 |  1
     ix | 67078438 |  2

(5 rows)
cqlsh:search> select count(*) from vocabulary;

 count
-------
 39630

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:search> select * from vocabulary limit 5;

 term     | df
----------+----
   dobson |  1
   bessus |  1
       ix |  4
    await |  2
 libertad |  1

(5 rows)
cqlsh:search>
```

11°C Mostly cloudy ENG 17:30 15.04.2025