[S25] IBD A2 Report

Matevosian Artem, B22-DS-01

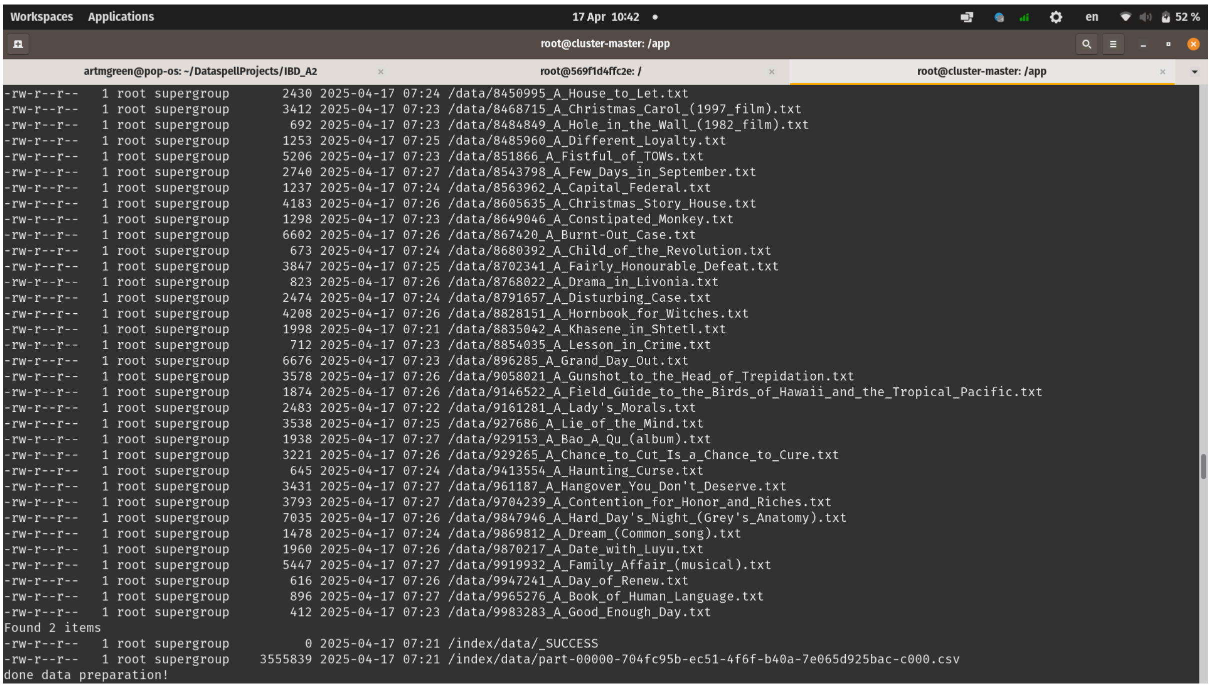
Methodology: Data preparation

The prepare_data.py from the original assignment statement managed to satisfy my needs after uncommenting writing to csv operation, therefore, I directly used the script from the assignment - it reads parquet file, selects 1000 entries, creates docs + writes to csv in the format <doc_id>\t<title>\t<text>

- I spent an abnormal amount of time trying to fix Out-Of-Memory errors that appeared independently of spark vectorized reader configs, later these reduced in frequency when I started deleting all the files from the previous runs right before main script execution.

Demonstration: Data preparation

Given that the script is directly taken from the assignment statement with minimal modifications for successful preparation on my low-resource machine, and the logs during its execution are extremely long, it does not appear feasible (or even needed) to print them out here. However, for the completeness purposes here is the screenshot of how the output end should generally look like whenever the preparation is successful:



Workspaces
Applications
17 Apr 10:42
en
52 %
root@cluster-master: /app
artmgreen@pop-os: ~/DataspellProjects/IBD_A2
root@569f1d4ffc2e: /
root@cluster-master: /app

```
-rw-r--r--   1 root supergroup       2430 2025-04-17 07:24 /data/8450995_A_House_to_Let.txt
-rw-r--r--   1 root supergroup       3412 2025-04-17 07:23 /data/8468715_A_Christmas_Carol_(1997_film).txt
-rw-r--r--   1 root supergroup        692 2025-04-17 07:23 /data/8484849_A_Hole_in_the_Wall_(1982_film).txt
-rw-r--r--   1 root supergroup       1253 2025-04-17 07:25 /data/8485960_A_Different_Loyalty.txt
-rw-r--r--   1 root supergroup       5206 2025-04-17 07:23 /data/851866_A_Fistful_of_TOWs.txt
-rw-r--r--   1 root supergroup       2740 2025-04-17 07:27 /data/8543798_A_Few_Days_in_September.txt
-rw-r--r--   1 root supergroup       1237 2025-04-17 07:24 /data/8563962_A_Capital_Federal.txt
-rw-r--r--   1 root supergroup       4183 2025-04-17 07:26 /data/8605635_A_Christmas_Story_House.txt
-rw-r--r--   1 root supergroup       1298 2025-04-17 07:23 /data/8649046_A_Constipated_Monkey.txt
-rw-r--r--   1 root supergroup       6602 2025-04-17 07:26 /data/867420_A_Burnt-Out_Case.txt
-rw-r--r--   1 root supergroup        673 2025-04-17 07:24 /data/8680392_A_Child_of_the_Revolution.txt
-rw-r--r--   1 root supergroup       3847 2025-04-17 07:25 /data/8702341_A_Fairly_Honourable_Defeat.txt
-rw-r--r--   1 root supergroup        823 2025-04-17 07:26 /data/8768022_A_Drama_in_Livonia.txt
-rw-r--r--   1 root supergroup       2474 2025-04-17 07:24 /data/8791657_A_Disturbing_Case.txt
-rw-r--r--   1 root supergroup       4208 2025-04-17 07:26 /data/8828151_A_Hornbook_for_Witches.txt
-rw-r--r--   1 root supergroup       1998 2025-04-17 07:21 /data/8835042_A_Khasene_in_Shtetl.txt
-rw-r--r--   1 root supergroup        712 2025-04-17 07:23 /data/8854035_A_Lesson_in_Crime.txt
-rw-r--r--   1 root supergroup       6676 2025-04-17 07:23 /data/896285_A_Grand_Day_Out.txt
-rw-r--r--   1 root supergroup       3578 2025-04-17 07:26 /data/9058021_A_Gunshot_to_the_Head_of_Trepidation.txt
-rw-r--r--   1 root supergroup       1874 2025-04-17 07:26 /data/9146522_A_Field_Guide_to_the_Birds_of_Hawaii_and_the_Tropical_Pacific.txt
-rw-r--r--   1 root supergroup       2483 2025-04-17 07:22 /data/9161281_A_Lady's_Morals.txt
-rw-r--r--   1 root supergroup       3538 2025-04-17 07:25 /data/927686_A_Lie_of_the_Mind.txt
-rw-r--r--   1 root supergroup       1938 2025-04-17 07:27 /data/929153_A_Bao_A_Qu_(album).txt
-rw-r--r--   1 root supergroup       3221 2025-04-17 07:26 /data/929265_A_Chance_to_Cut_Is_a_Chance_to_Cure.txt
-rw-r--r--   1 root supergroup        645 2025-04-17 07:24 /data/9413554_A_Haunting_Curse.txt
-rw-r--r--   1 root supergroup       3431 2025-04-17 07:27 /data/961187_A_Hangover_You_Don't_Deserve.txt
-rw-r--r--   1 root supergroup       3793 2025-04-17 07:27 /data/9704239_A_Contention_for_Honor_and_Riches.txt
-rw-r--r--   1 root supergroup       7035 2025-04-17 07:26 /data/9847946_A_Hard_Day's_Night_(Grey's_Anatomy).txt
-rw-r--r--   1 root supergroup       1478 2025-04-17 07:24 /data/9869812_A_Dream_(Common_song).txt
-rw-r--r--   1 root supergroup       1960 2025-04-17 07:26 /data/9870217_A_Date_with_Luyu.txt
-rw-r--r--   1 root supergroup       5447 2025-04-17 07:27 /data/9919932_A_Family_Affair_(musical).txt
-rw-r--r--   1 root supergroup        616 2025-04-17 07:26 /data/9947241_A_Day_of_Renew.txt
-rw-r--r--   1 root supergroup        896 2025-04-17 07:27 /data/9965276_A_Book_of_Human_Language.txt
-rw-r--r--   1 root supergroup        412 2025-04-17 07:23 /data/9983283_A_Good_Enough_Day.txt
Found 2 items
-rw-r--r--   1 root supergroup          0 2025-04-17 07:21 /index/data/_SUCCESS
-rw-r--r--   1 root supergroup    3555839 2025-04-17 07:21 /index/data/part-00000-704fc95b-ec51-4f6f-b40a-7e065d925bac-c000.csv
done data preparation!
```

NOTE: the screenshot is done while app.sh was running (so no commands were typed by hand here now, everything is in script)

Methodology: Indexer

The result after running indexing: cassandra-server gets the tables as listed below
(P = partitioning key, C = clustering key)

term_frequencies		
term	TEXT	P
document_id	INT	C (ASC)
tf	INT	

document_frequencies		
term	TEXT	P
df	INT	

doc_index		
document_id	INT	P
title	TEXT	
length	INT	

Why exactly these tables?

- term_frequencies = **tf(t, d)**
- document_frequencies = **df(t)**
- doc_index serves as mapping of document_id to title AND as **dl(d)**

For each table, there is a specific mapreduce job definition, ordered as below:

1. term_frequencies
2. document_frequencies
3. doc_index

The mapreduce jobs print out to /index/data in HDFS in the format directly transferable to cassandra, given the table definitions as above. There were multiple options to copy these outputs into Cassandra:

- (Selected by me) run consecutive INSERTS for every entry – able to run from wherever we have access to cassandra-server, no need to load the whole tsv locally when reading it lazily from HDFS
- run CQLSH COPY on TSVs (rejected since it supposedly requires to load outputs from HDFS into local filesystem)
- sstableloader after building same tables on the running machine (rejected for this very reason, cassandra-server should store the tables, I did not want to create them “locally” and then push somewhere else)

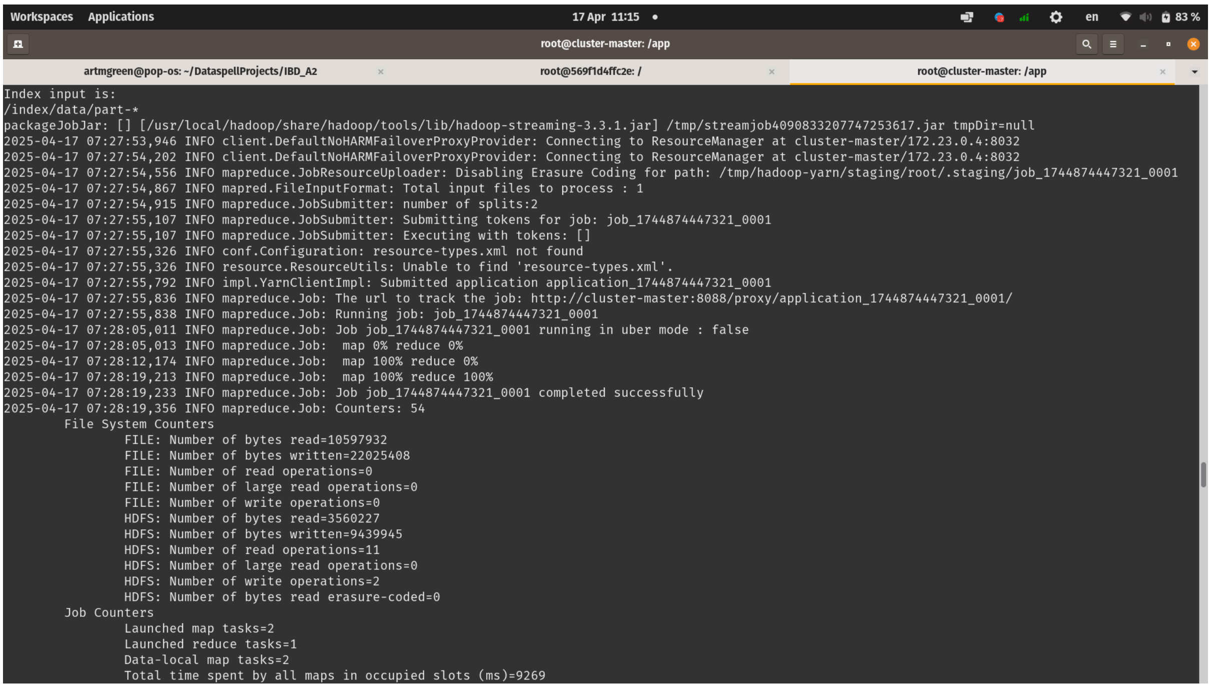
As a result, there are three tables defined as above.

NOTE: document_frequencies could be created much more quickly via
SELECT term, COUNT(document_id) FROM term_frequencies GROUP BY term;
However, according to Apache’s docs, Cassandra’s materialized views prohibit aggregate functions in their definitions. Also, printing the query result into a TSV for a subsequent CQLSH COPY may or may not be faster than running the second MapReduce pipeline, so I

stayed with the second pipeline. Finally, if SQL-based DBs had been available for the assignment (e.g. Postgres), it would have been possible to CREATE table AS SELECT statement as shown above.

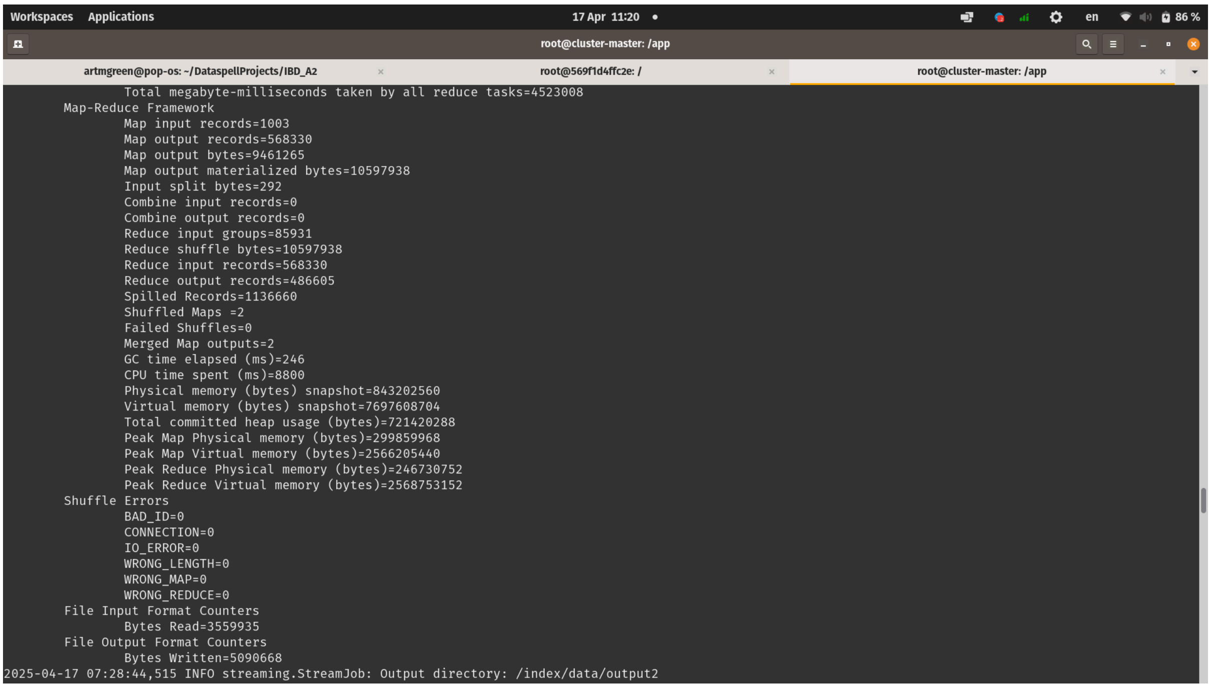
Demonstration: indexer’s result

Right after the indexer’s start the console should generally look like this:


Workspaces
Applications
17 Apr 11:15
en
83 %
root@cluster-master: /app
artmgreen@pop-os: ~/DataspellProjects/IBD_A2
root@569f1d4ffc2e: /
root@cluster-master: /app

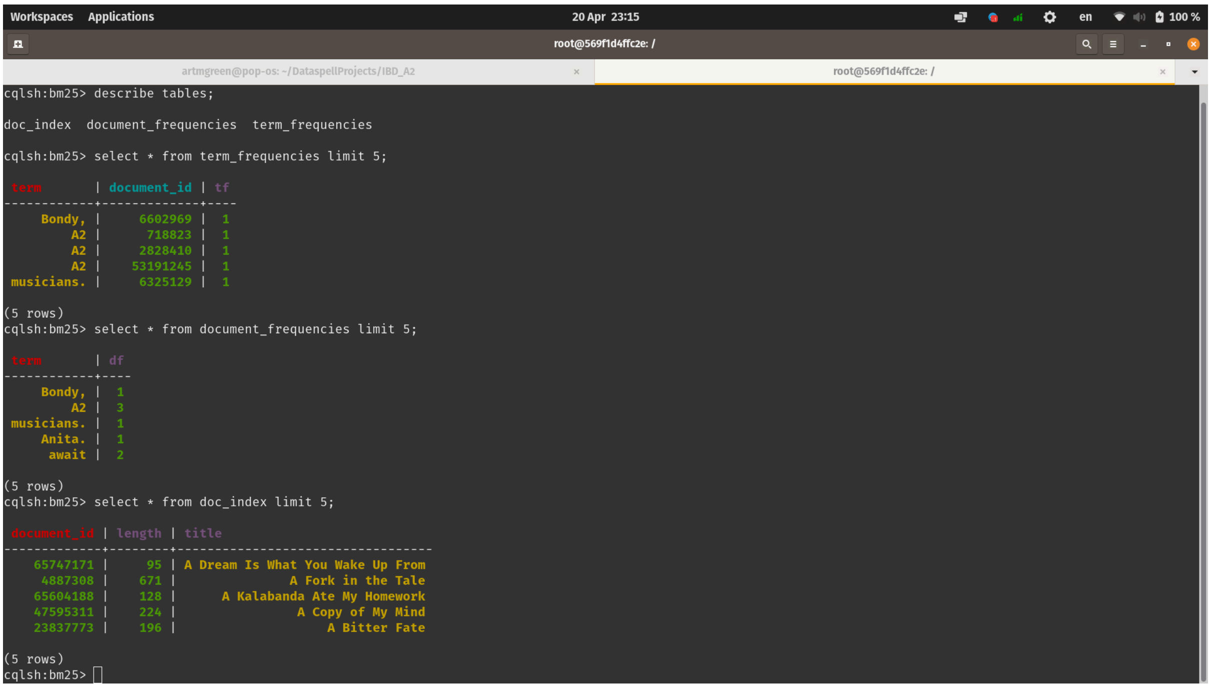
```
Index input is:
/index/data/part-*
packageJobJar: [] [/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar] /tmp/streamjob4090833207747253617.jar tmpDir=null
2025-04-17 07:27:53,946 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.23.0.4:8032
2025-04-17 07:27:54,202 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/172.23.0.4:8032
2025-04-17 07:27:54,556 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1744874447321_0001
2025-04-17 07:27:54,867 INFO mapred.FileInputFormat: Total input files to process : 1
2025-04-17 07:27:54,915 INFO mapreduce.JobSubmitter: number of splits:2
2025-04-17 07:27:55,107 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1744874447321_0001
2025-04-17 07:27:55,107 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-17 07:27:55,326 INFO conf.Configuration: resource-types.xml not found
2025-04-17 07:27:55,326 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-17 07:27:55,792 INFO impl.YarnClientImpl: Submitted application application_1744874447321_0001
2025-04-17 07:27:55,836 INFO mapreduce.Job: The url to track the job: http://cluster-master:8088/proxy/application_1744874447321_0001/
2025-04-17 07:27:55,838 INFO mapreduce.Job: Running job: job_1744874447321_0001
2025-04-17 07:28:05,011 INFO mapreduce.Job: Job job_1744874447321_0001 running in uber mode : false
2025-04-17 07:28:05,013 INFO mapreduce.Job:  map 0% reduce 0%
2025-04-17 07:28:12,174 INFO mapreduce.Job:  map 100% reduce 0%
2025-04-17 07:28:19,213 INFO mapreduce.Job:  map 100% reduce 100%
2025-04-17 07:28:19,233 INFO mapreduce.Job: Job job_1744874447321_0001 completed successfully
2025-04-17 07:28:19,356 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=10597932
                FILE: Number of bytes written=22025408
                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=9439945
                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)=9269
```

For each of the MapReduce jobs the output in the end should look similar to this:


Workspaces
Applications
17 Apr 11:20
en
86 %
root@cluster-master: /app
artmgreen@pop-os: ~/DataspellProjects/IBD_A2
root@569f1d4ffc2e: /
root@cluster-master: /app

```
                Total megabyte-milliseconds taken by all reduce tasks=4523008
        Map-Reduce Framework
                Map input records=1003
                Map output records=568330
                Map output bytes=9461265
                Map output materialized bytes=10597938
                Input split bytes=292
                Combine input records=0
                Combine output records=0
                Reduce input groups=85931
                Reduce shuffle bytes=10597938
                Reduce input records=568330
                Reduce output records=486605
                Spilled Records=1136660
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=246
                CPU time spent (ms)=8800
                Physical memory (bytes) snapshot=843202560
                Virtual memory (bytes) snapshot=7697608704
                Total committed heap usage (bytes)=721420288
                Peak Map Physical memory (bytes)=299859968
                Peak Map Virtual memory (bytes)=2566205440
                Peak Reduce Physical memory (bytes)=246730752
                Peak Reduce Virtual memory (bytes)=2568753152
        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=5090668
2025-04-17 07:28:44,515 INFO streaming.StreamJob: Output directory: /index/data/output2
```

For the purposes of demonstration of indexer’s results, this is the screenshot regarding Cassandra’s tables (I executed bash in cassandra-server container to be able to CQLSH it):



```
cqlsh:bm25> describe tables;

doc_index  document_frequencies  term_frequencies

cqlsh:bm25> select * from term_frequencies limit 5;

 term       | document_id | tf
------------+-------------+----
     Bondy, |     6602969 |  1
         A2 |      718823 |  1
         A2 |     2828410 |  1
         A2 |    53191245 |  1
 musicians. |     6325129 |  1

(5 rows)
cqlsh:bm25> select * from document_frequencies limit 5;

 term       | df
------------+----
     Bondy, |  1
         A2 |  3
 musicians. |  1
     Anita. |  1
      await |  2

(5 rows)
cqlsh:bm25> select * from doc_index limit 5;

 document_id | length | title
-------------+--------+---------------------------------
    65747171 |     95 | A Dream Is What You Wake Up From
     4887308 |    671 |              A Fork in the Tale
    65604188 |    128 |      A Kalabanda Ate My Homework
    47595311 |    224 |               A Copy of My Mind
    23837773 |    196 |                   A Bitter Fate

(5 rows)
cqlsh:bm25> 
```

Methodology: Ranker

Previous approach: wrap everything into a BM25_Calculator class and broadcast it over the workers. Rejected since the class contains Cassandra session pointer, which cannot be packed (“pickled”) to broadcast.

Current approach that one can see in query.py:
init_cassandra(): a function to give session credentials to whoever called it, e.g. workers to receive their own session (since it cannot be serialized and passed from main executor). Additionally it computes N and dl_{avg} (see query.py)

compute_bm25(query_terms, doc_id):

1. receives its own Cassandra session
2. checks if the suggested document exists at all (and returns empty title + 0 score if it does not)
3. iterates over every unique term in query and computes “scary-looking formula” from the assignment statement (+0.5 in logarithm numerator and denominator to prevent numerical errors, given that logarithms do not act nicely with zeros) for every term, sums over the terms and returns the sum with doc info
4. Shutdowns session — I do not want to deal with errors that may or may not appear if it does not.

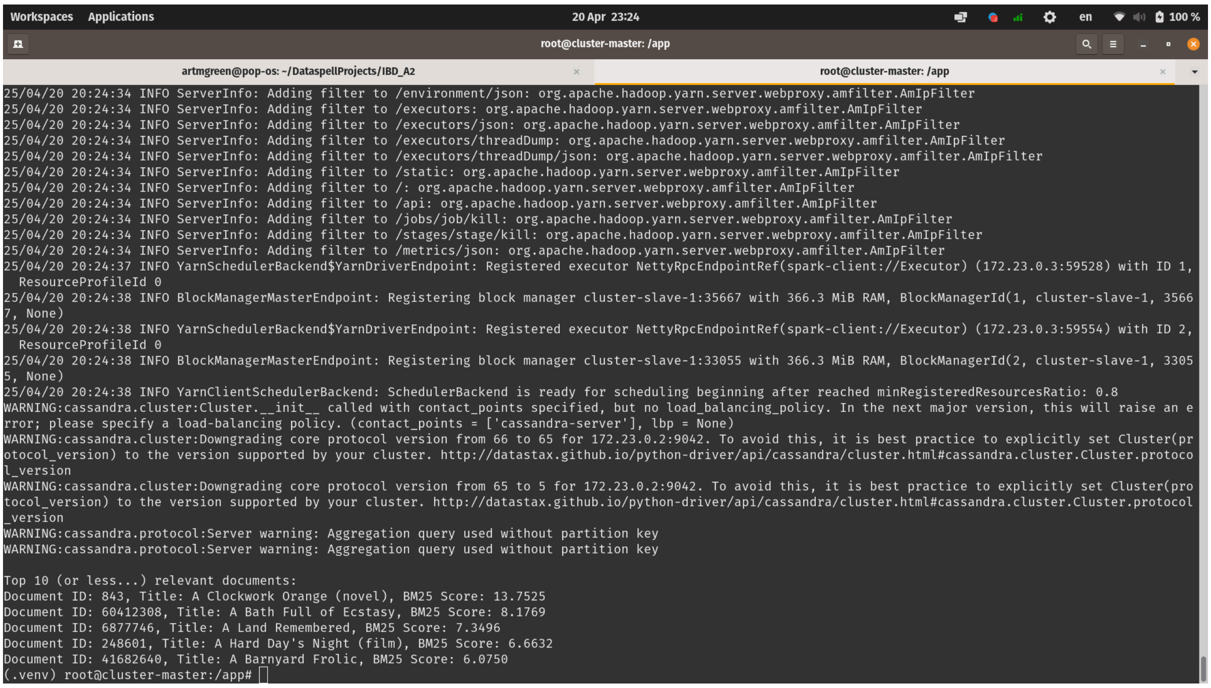
Top-level code:

1. parses query into terms
2. configures Spark

3. receives its own Cassandra session
4. finds all the documents that have non-zero chance to be relevant (tf(doc, term) ≠ 0)
5. computes BM25 in parallel (query is broadcasted, Cassandra session is created by workers on their own. Broadcasting N and dl_avg is a leftover for the testing purposes)

Demonstration: Ranker

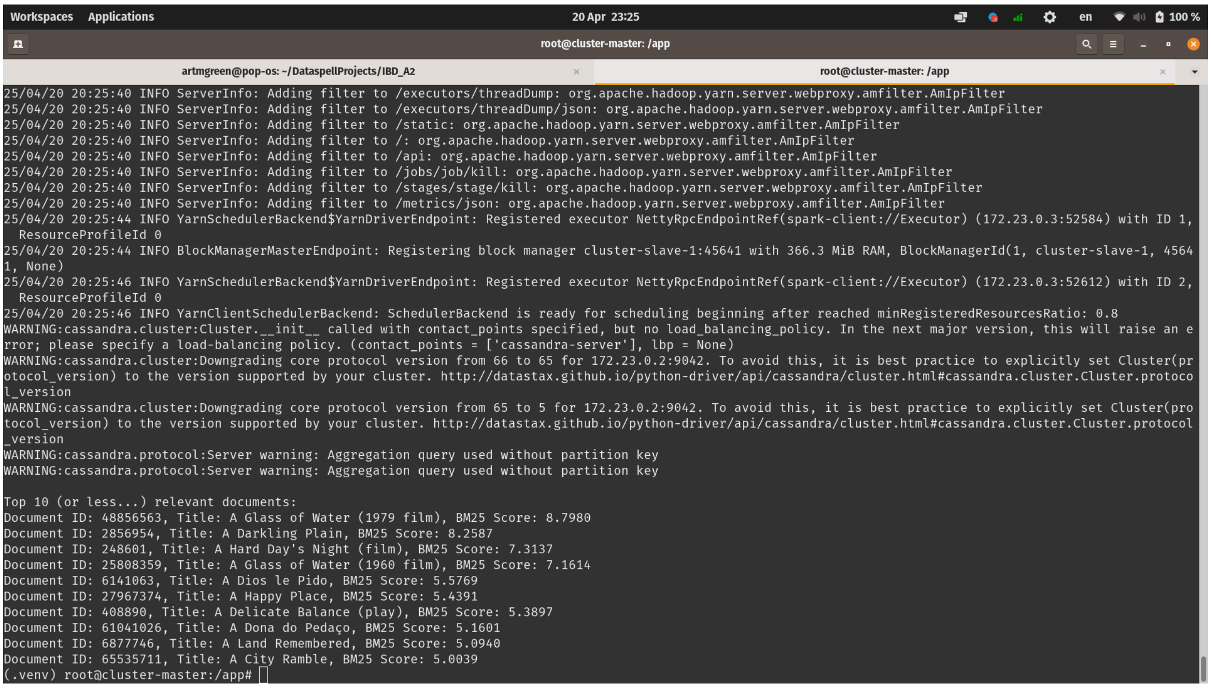
bash search.sh “clockwork orange milk”


Workspaces
Applications
20 Apr 23:24
en
100 %
root@cluster-master: /app
artmgreen@pop-os: ~/DataspellProjects/IBD_A2
root@cluster-master: /app

```
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /environment/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /executors: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /executors/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /executors/threadDump: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /executors/threadDump/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /static: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:34 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:24:37 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor NettyRpcEndpointRef(spark-client://Executor) (172.23.0.3:59528) with ID 1,
  ResourceProfileId 0
25/04/20 20:24:38 INFO BlockManagerMasterEndpoint: Registering block manager cluster-slave-1:35667 with 366.3 MiB RAM, BlockManagerId(1, cluster-slave-1, 3566
7, None)
25/04/20 20:24:38 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor NettyRpcEndpointRef(spark-client://Executor) (172.23.0.3:59554) with ID 2,
  ResourceProfileId 0
25/04/20 20:24:38 INFO BlockManagerMasterEndpoint: Registering block manager cluster-slave-1:33055 with 366.3 MiB RAM, BlockManagerId(2, cluster-slave-1, 3305
5, None)
25/04/20 20:24:38 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.8
WARNING:cassandra.cluster:Cluster.__init__ called with contact_points specified, but no load_balancing_policy. In the next major version, this will raise an e
rror; please specify a load-balancing policy. (contact_points = ['cassandra-server'], lbp = None)
WARNING:cassandra.cluster:Downgrading core protocol version from 66 to 65 for 172.23.0.2:9042. To avoid this, it is best practice to explicitly set Cluster(pr
otocol_version) to the version supported by your cluster. http://datastax.github.io/python-driver/api/cassandra/cluster.html#cassandra.cluster.Cluster.protoco
l_version
WARNING:cassandra.cluster:Downgrading core protocol version from 65 to 5 for 172.23.0.2:9042. To avoid this, it is best practice to explicitly set Cluster(pro
tocol_version) to the version supported by your cluster. http://datastax.github.io/python-driver/api/cassandra/cluster.html#cassandra.cluster.Cluster.protocol
_version
WARNING:cassandra.protocol:Server warning: Aggregation query used without partition key
WARNING:cassandra.protocol:Server warning: Aggregation query used without partition key

Top 10 (or less...) relevant documents:
Document ID: 843, Title: A Clockwork Orange (novel), BM25 Score: 13.7525
Document ID: 60412308, Title: A Bath Full of Ecstasy, BM25 Score: 8.1769
Document ID: 6877746, Title: A Land Remembered, BM25 Score: 7.3496
Document ID: 248601, Title: A Hard Day's Night (film), BM25 Score: 6.6632
Document ID: 41682640, Title: A Barnyard Frolic, BM25 Score: 6.0750
(.venv) root@cluster-master:/app# 
```

bash search.sh “war peace”


Workspaces
Applications
20 Apr 23:25
en
100 %
root@cluster-master: /app
artmgreen@pop-os: ~/DataspellProjects/IBD_A2
root@cluster-master: /app

```
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /executors/threadDump: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /executors/threadDump/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /static: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:40 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
25/04/20 20:25:44 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor NettyRpcEndpointRef(spark-client://Executor) (172.23.0.3:52584) with ID 1,
  ResourceProfileId 0
25/04/20 20:25:44 INFO BlockManagerMasterEndpoint: Registering block manager cluster-slave-1:45641 with 366.3 MiB RAM, BlockManagerId(1, cluster-slave-1, 4564
1, None)
25/04/20 20:25:46 INFO YarnSchedulerBackend$YarnDriverEndpoint: Registered executor NettyRpcEndpointRef(spark-client://Executor) (172.23.0.3:52612) with ID 2,
  ResourceProfileId 0
25/04/20 20:25:46 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.8
WARNING:cassandra.cluster:Cluster.__init__ called with contact_points specified, but no load_balancing_policy. In the next major version, this will raise an e
rror; please specify a load-balancing policy. (contact_points = ['cassandra-server'], lbp = None)
WARNING:cassandra.cluster:Downgrading core protocol version from 66 to 65 for 172.23.0.2:9042. To avoid this, it is best practice to explicitly set Cluster(pr
otocol_version) to the version supported by your cluster. http://datastax.github.io/python-driver/api/cassandra/cluster.html#cassandra.cluster.Cluster.protoco
l_version
WARNING:cassandra.cluster:Downgrading core protocol version from 65 to 5 for 172.23.0.2:9042. To avoid this, it is best practice to explicitly set Cluster(pro
tocol_version) to the version supported by your cluster. http://datastax.github.io/python-driver/api/cassandra/cluster.html#cassandra.cluster.Cluster.protocol
_version
WARNING:cassandra.protocol:Server warning: Aggregation query used without partition key
WARNING:cassandra.protocol:Server warning: Aggregation query used without partition key

Top 10 (or less...) relevant documents:
Document ID: 48856563, Title: A Glass of Water (1979 film), BM25 Score: 8.7980
Document ID: 2856954, Title: A Darkling Plain, BM25 Score: 8.2587
Document ID: 248601, Title: A Hard Day's Night (film), BM25 Score: 7.3137
Document ID: 25808359, Title: A Glass of Water (1960 film), BM25 Score: 7.1614
Document ID: 6141063, Title: A Dios le Pido, BM25 Score: 5.5769
Document ID: 27967374, Title: A Happy Place, BM25 Score: 5.4391
Document ID: 408890, Title: A Delicate Balance (play), BM25 Score: 5.3897
Document ID: 61041026, Title: A Dona do Pedaço, BM25 Score: 5.1601
Document ID: 6877746, Title: A Land Remembered, BM25 Score: 5.0940
Document ID: 65535711, Title: A City Ramble, BM25 Score: 5.0039
(.venv) root@cluster-master:/app# 
```

NOTE: I cut off results with BM25=0, so less than 10 documents appearing is fine. Also, for these screenshots I typed the listed commands on my own into cluster-master’s bash to be able to choose the most interesting ones and screen the outputs properly, but the same results should appear after running app.sh as well.