

Simple Search Engine using Hadoop MapReduce

Assignment 2 Report

This report documents the implementation of a simple search engine using Hadoop MapReduce, Apache Cassandra, and Spark RDD. The search engine indexes a collection of text documents from Wikipedia, builds an inverted index, and retrieves documents relevant to user queries based on the BM25 ranking algorithm. The implementation follows a distributed computing approach where document indexing is handled as batch processing using Hadoop MapReduce, while query processing is performed using Spark RDD for efficient real-time retrieval.

Methodology

System Architecture

The search engine implementation follows a distributed architecture with the following components:

1. **Document Processing:** Using PySpark to extract and prepare documents from parquet files
2. **Indexing Engine:** Using Hadoop MapReduce to build an inverted index of terms
3. **Storage Layer:** Using Apache Cassandra for storing the index data
4. **Query Processing:** Using Spark RDD to process queries and retrieve relevant documents

Indexing Process

The indexing engine uses a two-stage MapReduce pipeline to create an inverted index of terms from the document corpus:

First MapReduce Job:

- **Mapper (mapper1.py):**
 - Reads documents from input files
 - Tokenizes text content (lowercase, remove stopwords, apply stemming)
 - Calculates term frequencies for each document
 - Emits key-value pairs: `(term, (doc_id, term_freq, doc_length))`
- **Reducer (reducer1.py):**
 - Aggregates information for each term across all documents
 - Calculates document frequency for each term
 - Emits term data with document frequencies and posting lists

- Also emits corpus statistics as a special key-value pair

Second MapReduce Job:

- **Mapper (mapper2.py):**
 - Processes output from the first reducer
 - Separates vocabulary entries from posting entries
 - Formats data for insertion into Cassandra tables
- **Reducer (reducer2.py):**
 - Connects to Cassandra cluster
 - Creates required tables if they don't exist
 - Inserts vocabulary, postings, and corpus statistics into respective tables

This two-stage pipeline efficiently distributes the indexing workload across the Hadoop cluster and prepares the data for retrieval operations.

Cassandra Data Model

The search engine uses the following data model in Cassandra:

1. **vocabulary** table:

- **term** (text, PRIMARY KEY): Stemmed word from the corpus
- **doc_frequency** (int): Number of documents containing the term

2. **postings** table:

- **term** (text): Stemmed word
- **doc_id** (text): Document identifier
- **term_frequency** (int): Frequency of the term in the document
- PRIMARY KEY (term, doc_id): Enables efficient lookup of documents by term

3. **corpus_stats** table:

- **id** (text, PRIMARY KEY): Identifier for corpus statistics
- **doc_count** (int): Total number of documents in the corpus
- **avg_doc_length** (float): Average document length in the corpus

4. **document_stats** table:

- **doc_id** (text, PRIMARY KEY): Document identifier
- **doc_length** (int): Length of the document (number of terms)

This schema design allows for efficient retrieval of term statistics and document information needed for BM25 scoring.

Query Processing

Query processing is implemented using Spark RDD to provide efficient retrieval of relevant documents:

1. **Query Tokenization:** The user query is tokenized, stemmed, and processed in the same way as documents during indexing.
2. **Term Information Retrieval:** For each query term, the system retrieves:
 - Document frequency from the `vocabulary` table
 - Posting list from the `postings` table
 - Corpus statistics from the `corpus_stats` table
 - Document lengths from the `document_stats` table
3. **BM25 Scoring:** For each document containing any query term, the system calculates a BM25 score based on term frequency, document frequency, document length, and corpus statistics.
4. **Result Ranking:** Documents are sorted by their BM25 scores, and the top 10 results are returned to the user.

Demonstration

Document Indexing

The indexing process was tested on a collection of 1000 Wikipedia articles. The process included data preparation, two MapReduce jobs, and storage in Cassandra.

```
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedParquetRecordReader.nextBatch(VectorizedParquetRecordReader.java:342)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedParquetRecordReader.nextKeyValue(VectorizedParquetRecordReader.java:233)
Cluster-master at org.apache.spark.sql.execution.datasources.RecordReaderIterator.hasNext(RecordReaderIterator.scala:39)
Cluster-master at org.apache.spark.sql.execution.datasources.FileScanRDD$$anon$1.hasNext(FileScanRDD.scala:131)
Cluster-master at org.apache.spark.sql.execution.datasources.FileScanRDD$$anon$1.nextIterator(FileScanRDD.scala:286)
Cluster-master ... 22 more
Cluster-master Caused by: java.lang.OutOfMemoryError: Java heap space
Cluster-master at org.apache.spark.sql.execution.vectorized.OnHeapColumnVector.reserveInternal(OnHeapColumnVector.java:594)
Cluster-master at org.apache.spark.sql.execution.vectorized.WritableColumnVector.reserve(WritableColumnVector.java:106)
Cluster-master at org.apache.spark.sql.execution.vectorized.WritableColumnVector.appendBytes(WritableColumnVector.java:544)
Cluster-master at org.apache.spark.sql.execution.vectorized.OnHeapColumnVector.putBytesArray(OnHeapColumnVector.java:568)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedPlainValuesReader.readBinary(VectorizedPlainValuesReader.java:366)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.ParquetVectorUpdaterFactory$BinaryUpdater.readValues(ParquetVectorUpdaterFactory.java:736)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedRLEValuesReader.readBatchInternal(VectorizedRLEValuesReader.java:244)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedRLEValuesReader.readBatch(VectorizedRLEValuesReader.java:176)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedColumnReader.readBatch(VectorizedColumnReader.java:252)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedParquetRecordReader.nextBatch(VectorizedParquetRecordReader.java:342)
Cluster-master at org.apache.spark.sql.execution.datasources.parquet.VectorizedParquetRecordReader.nextKeyValue(VectorizedParquetRecordReader.java:233)
Cluster-master at org.apache.spark.sql.execution.datasources.RecordReaderIterator.hasNext(RecordReaderIterator.scala:39)
Cluster-master at org.apache.spark.sql.execution.datasources.FileScanRDD$$anon$1.hasNext(FileScanRDD.scala:131)
Cluster-master at org.apache.spark.sql.execution.datasources.FileScanRDD$$anon$1.nextIterator(FileScanRDD.scala:286)
Cluster-master at org.apache.spark.sql.execution.datasources.FileScanRDD$$anon$1.hasNext(FileScanRDD.scala:131)
Cluster-master at org.apache.spark.sql.execution.datasources.FileScanRDD$$anon$1.nextIterator(FileScanRDD.scala:286)
Cluster-master at org.apache.spark.sql.execution.datasources.FileScanRDD$$anon$1.hasNext(FileScanRDD.scala:131)
Cluster-master at org.apache.spark.sql.catalyst.expressions.GeneratedClass$GeneratedIteratorForCodegenStage1.columnarToRow_nextBatch_0$(Unknown Source)
Cluster-master at org.apache.spark.sql.execution.BufferedRowIterator.hasNext(BufferedRowIterator.java:42)
Cluster-master at org.apache.spark.sql.execution.WholeStageCodegenEvaluatorFactory$WholeStageCodegenPartitionEvaluator$anon$1.hasNext(WholeStageCodegenEvaluatorFactory.scala:43)
Cluster-master at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:268)
Cluster-master at scala.collection.Iterator$$anon$10.hasNext(Iterator.scala:468)
Cluster-master at org.apache.spark.shuffle.sort.BypassMergeSortShuffleWriter.write(BypassMergeSortShuffleWriter.java:140)
Cluster-master at org.apache.spark.shuffle.ShuffleWriteProcessor.write(ShuffleWriteProcessor.scala:59)
Cluster-master at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:104)
Cluster-master at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:54)
Cluster-master at org.apache.spark.TaskContext.runTaskWithListeners(TaskContext.scala:166)
Cluster-master at org.apache.spark.scheduler.Task.run(Task.scala:141)
Cluster-master at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$4(Executor.scala:630)
Cluster-master at org.apache.spark.executor.Executor$TaskRunner$$anon$1.run$$anon$1(Executor.scala:630)
Cluster-master at org.apache.spark.util.SparkErrorUtils.tryWithSafeFinally(SparkErrorUtils.scala:64)
Cluster-master at org.apache.spark.util.SparkErrorUtils.tryWithSafeFinally(SparkErrorUtils.scala:61)
Cluster-master
Cluster-master Driver stacktrace:
Cluster-master 25/04/14 19:52:59 INFO DAGScheduler: Job 3 failed: foreach at /app/prepare_data.py:22, took 28.471510 s
Cluster-master 25/04/14 19:52:59 INFO SparkUI: Stopped Spark web UI at http://cluster-master:4040
Cluster-master 25/04/14 19:53:00 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
Cluster-master 25/04/14 19:53:00 INFO MemoryStore: MemoryStore cleared
Cluster-master 25/04/14 19:53:00 INFO BlockManager: BlockManager stopped
Cluster-master 25/04/14 19:53:00 INFO BlockManagerMaster: BlockManagerMaster stopped
Cluster-master 25/04/14 19:53:00 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
Cluster-master 25/04/14 19:53:01 INFO SparkContext: Successfully stopped SparkContext
Cluster-master 25/04/14 19:53:01 INFO ShutdownHookManager: Shutdown hook called
Cluster-master 25/04/14 19:53:01 INFO ShutdownHookManager: Deleting directory /tmp/spark-3bd8eed-74f9-4087-b081-68f7ab76323/pyspark-c2c468d4-437c-44f3-b9a8-6219a28e18af
Cluster-master 25/04/14 19:53:01 INFO ShutdownHookManager: Deleting directory /tmp/spark-9f1bd1a8-beee-4537-8368-764b054bf6b
Cluster-master 25/04/14 19:53:01 INFO ShutdownHookManager: Deleting directory /tmp/spark-3bd8eed-74f9-4087-b081-68f7ab76323
Cluster-master Running MapReduce jobs to index documents
Cluster-master Waiting...
Cluster-master Starting first MapReduce job...
Cluster-master Waiting...
Cluster-master First MapReduce job completed.
Cluster-master Starting second MapReduce job...
Cluster-master Waiting...
Cluster-master Second MapReduce job completed.
Cluster-master Indexing completed successfully!
Cluster-master Running sample searches...
```

Search Engine Queries

Query: death

```
Activities Terminal anp 15 23:20
root@goodboy-A7S: /home/goodboy/PycharmProjects/BigData_ass2
This script will include commands to search for documents given the query using Spark RDD
25/04/15 20:18:08 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Query: death
-----Top 10 Results:-----
Row(title='A Band Called Death')
1. Document: 40811318 Score: 4.094
Row(title='A Certain Kind of Death')
2. Document: 47387911 Score: 3.889
Row(title='A Beautiful Blue Death')
3. Document: 26389272 Score: 3.883
Row(title='A Bao A Qu (album)')
4. Document: 929153 Score: 3.774
Row(title='A Break In the Weather')
5. Document: 9277570 Score: 3.707
Row(title='A Blaze In the Northern Sky')
6. Document: 2054290 Score: 3.586
Row(title='A Capitol Death')
7. Document: 56905939 Score: 3.509
Row(title='A Blood Pledge')
8. Document: 23233139 Score: 3.470
Row(title='A Bay of Blood')
9. Document: 4136530 Score: 3.353
Row(title='A Child Across the Sky')
10. Document: 40847643 Score: 3.324
25/04/15 20:18:08 INFO ShutdownHookManager: Shutdown hook called
25/04/15 20:18:08 INFO ShutdownHookManager: Deleting directory /tmp/spark-72bcd29-7d55-4910-bbc3-d30efb1ab631
This script will include commands to search for documents given the query using Spark RDD
25/04/15 20:18:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Query: Economist book
-----Top 10 Results:-----
Row(title='A Capitalism for the People')
1. Document: 36350251 Score: 10.079
Row(title='A Brief History of Equality')
2. Document: 70599262 Score: 10.343
Row(title='A Beautiful Mind (book)')
3. Document: 5543816 Score: 8.745
Row(title='A Behavioral Theory of the Firm')
4. Document: 31977736 Score: 8.744
```

Query: Economist book

```
Activities Terminal
root@goodboy-A7S: /home/goodboy/PycharmProjects/BigData_ess2

cluster-master Row(title='A Child Across the Sky')
cluster-master 10. Document: 40847643 Score: 3.324
cluster-master
cluster-master 25/04/15 20:18:08 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 20:18:08 INFO ShutdownHookManager: Deleting directory /tmp/spark-72bcdc29-7d55-4910-bbc3-d30efb1ab631
cluster-master This script will include commands to search for documents given the query using Spark RDD
cluster-master 25/04/15 20:18:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cluster-master Query: Economist book
cluster-master
cluster-master -----Top 10 Results-----
cluster-master Row(title='A Capitalism for the People')
cluster-master 1. Document: 36350251 Score: 10.079
cluster-master
cluster-master Row(title='A Brief History of Equality')
cluster-master 2. Document: 70599262 Score: 10.343
cluster-master
cluster-master Row(title='A Beautiful Mind (book)')
cluster-master 3. Document: 5543016 Score: 8.745
cluster-master
cluster-master Row(title='A Behavioral Theory of the Firm')
cluster-master 4. Document: 33277726 Score: 5.634
cluster-master
cluster-master Row(title='A Better Way')
cluster-master 5. Document: 52245959 Score: 4.465
cluster-master
cluster-master Row(title='A Book of Rhymes')
cluster-master 6. Document: 71478678 Score: 2.987
cluster-master
cluster-master Row(title='A Ball for Datsy')
cluster-master 7. Document: 34488106 Score: 2.971
cluster-master
cluster-master Row(title='A Big Mooncake for Little Star')
cluster-master 8. Document: 60578215 Score: 2.965
cluster-master
cluster-master Row(title='A Birthday Cake For George Washington')
cluster-master 9. Document: 49242941 Score: 2.955
cluster-master
cluster-master Row(title='A Bilingual Field Guide to the Frogs of Zululand')
cluster-master 10. Document: 69610272 Score: 2.955
cluster-master
cluster-master 25/04/15 20:18:10 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 20:18:10 INFO ShutdownHookManager: Deleting directory /tmp/spark-ca2c07fd-9f61-4233-a0c6-7398134bf91e
cluster-master This script will include commands to search for documents given the query using Spark RDD
cluster-master 25/04/15 20:18:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cluster-master Query: Hero of our time
cluster-master
cluster-master -----Top 10 Results-----
cluster-master Row(title='A Bewildered Lovebird')
cluster-master 1. Document: 30230015 Score: 10.232
cluster-master
cluster-master Row(title='A Cartoonist's Nightmare')
cluster-master 2. Document: 32729135 Score: 7.320
cluster-master
cluster-master Row(title='A Bit Off the Map')
cluster-master 3. Document: 23397726 Score: 6.703
cluster-master
cluster-master Row(title='A Bite of China')
cluster-master 4. Document: 30253472 Score: 6.420
cluster-master
cluster-master Row(title='A Case of Spring Fever')
cluster-master 5. Document: 7214270 Score: 5.754
cluster-master
cluster-master Row(title='A Bride from the Bush')
cluster-master 6. Document: 43098388 Score: 5.702
cluster-master
cluster-master Row(title='A Brief History of Everyone Who Ever Lived')
cluster-master 7. Document: 60121915 Score: 5.663
cluster-master
cluster-master Row(title='A Brand New Hero')
cluster-master 8. Document: 13720602 Score: 5.647
cluster-master
cluster-master Row(title='A Blackmailer's Trick')
cluster-master 9. Document: 48857668 Score: 5.447
cluster-master
cluster-master Row(title='A Better Understanding')
cluster-master 10. Document: 59929735 Score: 5.432
cluster-master
cluster-master 25/04/15 20:18:13 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 20:18:13 INFO ShutdownHookManager: Deleting directory /tmp/spark-9b485fef-0aa1-4f90-ae24-fe6775380b89
cluster-master This script will include commands to search for documents given the query using Spark RDD
cluster-master 25/04/15 20:18:14 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

Query: Hero of our time

```
Activities Terminal
root@goodboy-A7S: /home/goodboy/PycharmProjects/BigData_ess2

cluster-master Row(title='A Child Across the Sky')
cluster-master 10. Document: 40847643 Score: 3.324
cluster-master
cluster-master 25/04/15 20:18:08 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 20:18:08 INFO ShutdownHookManager: Deleting directory /tmp/spark-72bcdc29-7d55-4910-bbc3-d30efb1ab631
cluster-master This script will include commands to search for documents given the query using Spark RDD
cluster-master 25/04/15 20:18:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cluster-master Query: Hero of our time
cluster-master
cluster-master -----Top 10 Results-----
cluster-master Row(title='A Bewildered Lovebird')
cluster-master 1. Document: 30230015 Score: 10.232
cluster-master
cluster-master Row(title='A Cartoonist's Nightmare')
cluster-master 2. Document: 32729135 Score: 7.320
cluster-master
cluster-master Row(title='A Bit Off the Map')
cluster-master 3. Document: 23397726 Score: 6.703
cluster-master
cluster-master Row(title='A Bite of China')
cluster-master 4. Document: 30253472 Score: 6.420
cluster-master
cluster-master Row(title='A Case of Spring Fever')
cluster-master 5. Document: 7214270 Score: 5.754
cluster-master
cluster-master Row(title='A Bride from the Bush')
cluster-master 6. Document: 43098388 Score: 5.702
cluster-master
cluster-master Row(title='A Brief History of Everyone Who Ever Lived')
cluster-master 7. Document: 60121915 Score: 5.663
cluster-master
cluster-master Row(title='A Brand New Hero')
cluster-master 8. Document: 13720602 Score: 5.647
cluster-master
cluster-master Row(title='A Blackmailer's Trick')
cluster-master 9. Document: 48857668 Score: 5.447
cluster-master
cluster-master Row(title='A Better Understanding')
cluster-master 10. Document: 59929735 Score: 5.432
cluster-master
cluster-master 25/04/15 20:18:13 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 20:18:13 INFO ShutdownHookManager: Deleting directory /tmp/spark-9b485fef-0aa1-4f90-ae24-fe6775380b89
cluster-master This script will include commands to search for documents given the query using Spark RDD
cluster-master 25/04/15 20:18:14 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

Query: crime fiction

```
Activities Terminal
root@goodboy-A7S: /home/goodboy/PycharmProjects/BigData_ess2

cluster-master Row(title='A Bride from the Bush')
cluster-master 6. Document: 43098388 Score: 5.702
cluster-master Row(title='A Brief History of Everyone Who Ever Lived')
cluster-master 7. Document: 60121915 Score: 5.663
cluster-master Row(title='A Brand New Hero')
cluster-master 8. Document: 13720602 Score: 5.647
cluster-master Row(title='A Blackmailer's Trick')
cluster-master 9. Document: 48857668 Score: 5.447
cluster-master Row(title='A Better Understanding')
cluster-master 10. Document: 59929735 Score: 5.432
cluster-master 25/04/15 20:18:13 INFO ShutdownHookManager: Shutdown hook called
cluster-master 25/04/15 20:18:13 INFO ShutdownHookManager: Deleting directory /tmp/spark-9b485fef-0a21-4f90-ae24-fe6775380b89
cluster-master This script will include commands to search for documents given the query using Spark RDD
cluster-master 25/04/15 20:18:14 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cluster-master Query: crime fiction
cluster-master -----Top 10 Results:-----
cluster-master Row(title='A Catalogue of Crime')
cluster-master 1. Document: 11904610 Score: 8.703
cluster-master Row(title='A Brief History of Crime')
cluster-master 2. Document: 25804321 Score: 7.720
cluster-master Row(title='A A Dhand')
cluster-master 3. Document: 54187039 Score: 7.376
cluster-master Row(title='A Beautiful Place to Die')
cluster-master 4. Document: 20980418 Score: 7.203
cluster-master Row(title='A Case of Deadly Force')
cluster-master 5. Document: 22161712 Score: 7.105
cluster-master Row(title='A Beautiful Crime')
cluster-master 6. Document: 68846919 Score: 6.590
cluster-master Row(title='A Brief History of Blasphemy')
cluster-master 7. Document: 34353837 Score: 6.081
cluster-master Row(title='A Brief History of Seven Killings')
cluster-master 8. Document: 45611510 Score: 5.385
cluster-master Row(title='A Bullet in the Gun Barrel')
cluster-master 9. Document: 49458160 Score: 5.331
cluster-master Row(title='A Brooklyn State of Mind')
cluster-master 10. Document: 42089894 Score: 5.318
```

Conclusion

This implementation demonstrates a functional search engine using Hadoop MapReduce for indexing, Cassandra for storage, and Spark RDD for query processing. The BM25 ranking algorithm provides relevant search results by considering term frequency, document frequency, and document length.

The distributed architecture allows the system to handle large document collections efficiently, with batch processing for indexing and low-latency retrieval for query processing. While this implementation focuses on basic text search functionality, it provides a foundation that could be extended with additional features such as phrase matching, relevance feedback, or vector space models.

The project successfully demonstrates the application of Big Data technologies (Hadoop, Cassandra, Spark) to the problem of information retrieval, highlighting both the benefits and challenges of distributed computing approaches for search applications.