

Assignment #2 Report

Timofey Brayko
t.brayko@innopolis.university

April 19, 2025

Methodology

Architecture Design

The search engine indexing system implements:

- **2-Stage MapReduce Pipeline:**
 - Pipeline 1: Term Frequency (TF) calculation with Hadoop
 - Pipeline 2: Document Frequency (DF) aggregation
- **Cassandra Data Model:**
 - `inv_index`: Term-document mappings
 - `doc_stats`: Document lengths
 - `vocab_stats`: Global term frequencies
- **Batch Processing:**
 - Batched Cassandra writes (100 operations/batch)
 - Hadoop-Cassandra integration via Python driver

Implementation Details

Listing 1: Mapper 1 (TF Calculation)

```
def count_tf():
    for content in sys.stdin:
        input_file = os.getenv('mapreduce_map_input_file')
        id = extract_id(input_file) # Custom logic
        tokens = re.findall(r'\w+', content.lower())
        print(f"DOC_{id}\tLEN\t{len(tokens)}")
        for term, count in Counter(tokens).items():
            print(f"{term}\t{id}\t{count}")
```

Implementation Details (Reducer 2)

Listing 2: Cassandra Batch Reducer (reducer2.py)

```
# Batch flushing logic
def flush_batches(force=False):
    #Executing collected commands and clear batches...

def process_key_data(key, data):
    global batch_counts
    if key.startswith("DOC_"):
        # Handle document metadata
        doc_id = int(key.split("_")[-1])
        for value_type, value in data:
            if value_type == "LEN":
                batch_doc_stats.add(doc_stats_stmt,
                                   (doc_id, int(value)))
                batch_counts['doc'] += 1
    else:
        # Handle term statistics
        term = key
        doc_frequency = 0
        term_doc_pairs = []
        # Process DF and TF values
        for value_type, value in data:
            if value_type == "DF":
                doc_frequency += int(value)
            else:
                term_doc_pairs.append(
                    (int(value_type), int(value)))
        # Update vocabulary and inverted index
        if doc_frequency > 0:
            batch_vocab.add(vocab_stats_stmt,
                           (term, doc_frequency))
            batch_counts['vocab'] += 1
        for doc_id, tf in term_doc_pairs:
            batch_inv_idx.add(inverted_idx_stmt,
                              (term, doc_id, tf))
            batch_counts['inv'] += 1
    flush_batches()

# Main execution flow
if __name__ == "__main__":
    # Cassandra connection setup
    #Here connecting to Cassandra and creating batch processors....

    # Data processing loop
    current_key = None
    key_data = defaultdict(list)
    for line in sys.stdin:
        key, value_type, value = line.strip().split('\t')
        # Key change detection
        if current_key and key != current_key:
            process_key_data(current_key, key_data[current_key])
            del key_data[current_key]
            current_key = key
            key_data[current_key].append((value_type, value))

    # Final flush
    if current_key:
        process_key_data(current_key, key_data[current_key])
    flush_batches(force=True)
```

Demonstration

Execution Guide

1. Start infrastructure:

```
docker-compose up -d
```

2. Enter docker entry:

```
sudo docker exec -it cluster-master bash
```

3. Run pipeline:

```
bash app.sh
```

[illegible]

```

11/04/19 14:01:00 INFO BroadcastManager: Added broadcast_8 placed in memory on cluster-master-20043 (size: 3.8 KiB, free: 366.3 MiB)
11/04/19 14:01:00 INFO SparkContext: Created broadcast_9 from collect at /app/qq.py:183
11/04/19 14:01:00 INFO BroadcastManager: Added broadcast_9 placed in memory on cluster-master-20043 (size: 3.8 KiB, free: 366.3 MiB)
11/04/19 14:01:00 INFO SparkContext: Code generated in 12.737276 ms
11/04/19 14:01:00 INFO BroadcastManager: Added broadcast_10 placed in memory on cluster-master-20043 (size: 3.8 KiB, free: 366.3 MiB)
11/04/19 14:01:00 INFO SparkContext: Code generated in 63.815216 ms
11/04/19 14:01:00 INFO SparkContext: Starting job: collect at /app/qq.py:183
11/04/19 14:01:00 INFO DAGScheduler: Got job_7 (collect at /app/qq.py:183) with 1 output partitions
11/04/19 14:01:00 INFO DAGScheduler: Initial DAG
11/04/19 14:01:00 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 18)
11/04/19 14:01:00 INFO DAGScheduler: Missing parents: List()
11/04/19 14:01:00 INFO DAGScheduler: Submitting ShuffleMapStage 17 at collect at /app/qq.py:183, which has no missing parents
11/04/19 14:01:00 INFO MemoryStore: Block broadcast_8 stored as value in memory (estimated size 79.2 KiB, free 366.1 MiB)
11/04/19 14:01:00 INFO MemoryStore: Block broadcast_9 stored as value in memory (estimated size 39.6 KiB, free 366.1 MiB)
11/04/19 14:01:00 INFO MemoryStore: Added broadcast_8 placed in memory on cluster-master-20043 (size: 39.6 KiB, free: 366.3 MiB)
11/04/19 14:01:00 INFO SparkContext: Creating task 0 from broadcast at DAGScheduler.scala:1585
11/04/19 14:01:00 INFO DAGScheduler: Submitting 1 missing tasks from ShuffleMapStage 17 (ShuffleMapStage 17) at collect at /app/qq.py:183 (first 15 tasks are for partitions Vector())
11/04/19 14:01:00 INFO TaskScheduler: Adding task set 11.0 with 1 tasks: resource profile 0
11/04/19 14:01:00 INFO TaskScheduler: Starting task 0.0 in stage 11.0 (TID 1) (collector-l1-executor-2-partition-0, NODE_LOCAL, 8018 bytes)
11/04/19 14:01:00 INFO TaskScheduler: Added broadcast_8 placed in memory on cluster-lane-1-02493 (size: 39.6 KiB, free: 366.3 MiB)
11/04/19 14:01:00 INFO TaskScheduler: Added task set 11.0 with 1 tasks: resource profile 0
11/04/19 14:01:00 INFO TaskScheduler: Finished task 0.0 in stage 11.0 (TID 1) in 348 ms on cluster-lane-1-executor-2 (r/r)
11/04/19 14:01:00 INFO TaskScheduler: Renewed taskset 11.0, whose tasks have all completed, from pool
11/04/19 14:01:00 INFO DAGScheduler: ShuffleMapStage 17 (collect at /app/qq.py:183) finished in 348 s
11/04/19 14:01:00 INFO DAGScheduler: Job 7 is finished. Cleaning potential executors at node tasks for this job
11/04/19 14:01:00 INFO YarnScheduler: Killing all running tasks in stage 11: Stage finished
11/04/19 14:01:00 INFO DAGScheduler: Job 7 finished: collect at /app/qq.py:183, took 0.300461 s

top 10 relevant documents:
1 Document ID: 11154518, Score: 3.1964
2 Document ID: 53484821, Score: 2.7557
3 Document ID: 13628411, Score: 2.4753
4 Document ID: 33629700, Score: 2.4055
5 Document ID: 18299310, Score: 2.3484
6 Document ID: 32155051, Score: 2.2174
7 Document ID: 48588410, Score: 2.2121
8 Document ID: 33155051, Score: 2.1685
9 Document ID: 48877881, Score: 2.4051
10 Document ID: 36434220, Score: 1.9553

11/04/19 14:01:07 INFO SparkContext: SparkContext is stopping with exitCode 0
11/04/19 14:01:07 INFO SparkContext: Stopped task with 01 at http://cluster-master-20043:4040
11/04/19 14:01:07 INFO YarnClientSchedulerBackend: Interrupting monitor thread
11/04/19 14:01:07 INFO YarnClientSchedulerBackend: Shutting down all executors
11/04/19 14:01:07 INFO YarnClientSchedulerBackend: Interrupting executors
11/04/19 14:01:07 INFO YarnClientSchedulerBackend: Asking each executor to shut down
11/04/19 14:01:07 INFO YarnClientSchedulerBackend: Yarn client scheduler backend stopped
11/04/19 14:01:07 INFO YarnClientSchedulerBackend: YarnClientSchedulerBackend stopped
11/04/19 14:01:07 INFO MemoryStore: MemoryStore cleared
11/04/19 14:01:07 INFO BlockManager: BlockManager stopped
11/04/19 14:01:07 INFO BlockManager: BlockManager stopped
11/04/19 14:01:07 INFO OutputCommitCoordinator: OutputCommitCoordinator stopped
11/04/19 14:01:07 INFO SparkContext: Successfully stopped SparkContext
11/04/19 14:01:07 INFO ShutdownHookManager: Shutdown hook called

```