Jawad Ahmad

**MaP Reduce-Based Hourly Trending Topics Analysis**

## **1. Objective**

The objective was to build a MapReduce-based pipeline on Cloudera (or a local Hadoop simulation) to ingest timestamped tweet data, clean and tokenize the text into candidate topics, identify the top N trending topics for each hour, and visualize the trends. The pipeline leverages Hadoop's distributed processing capabilities to handle large-scale text data and produce meaningful insights into hourly topic trends.

## **2. Dataset Description**

The dataset used is a collection of timestamped tweets stored in tweets.txt. Each tweet includes a timestamp (in the format YYYY-MM-DD HH) and the tweet text, which often mentions airlines (e.g., @VirginAmerica, @AmericanAir) and includes user sentiments, complaints, or praises.

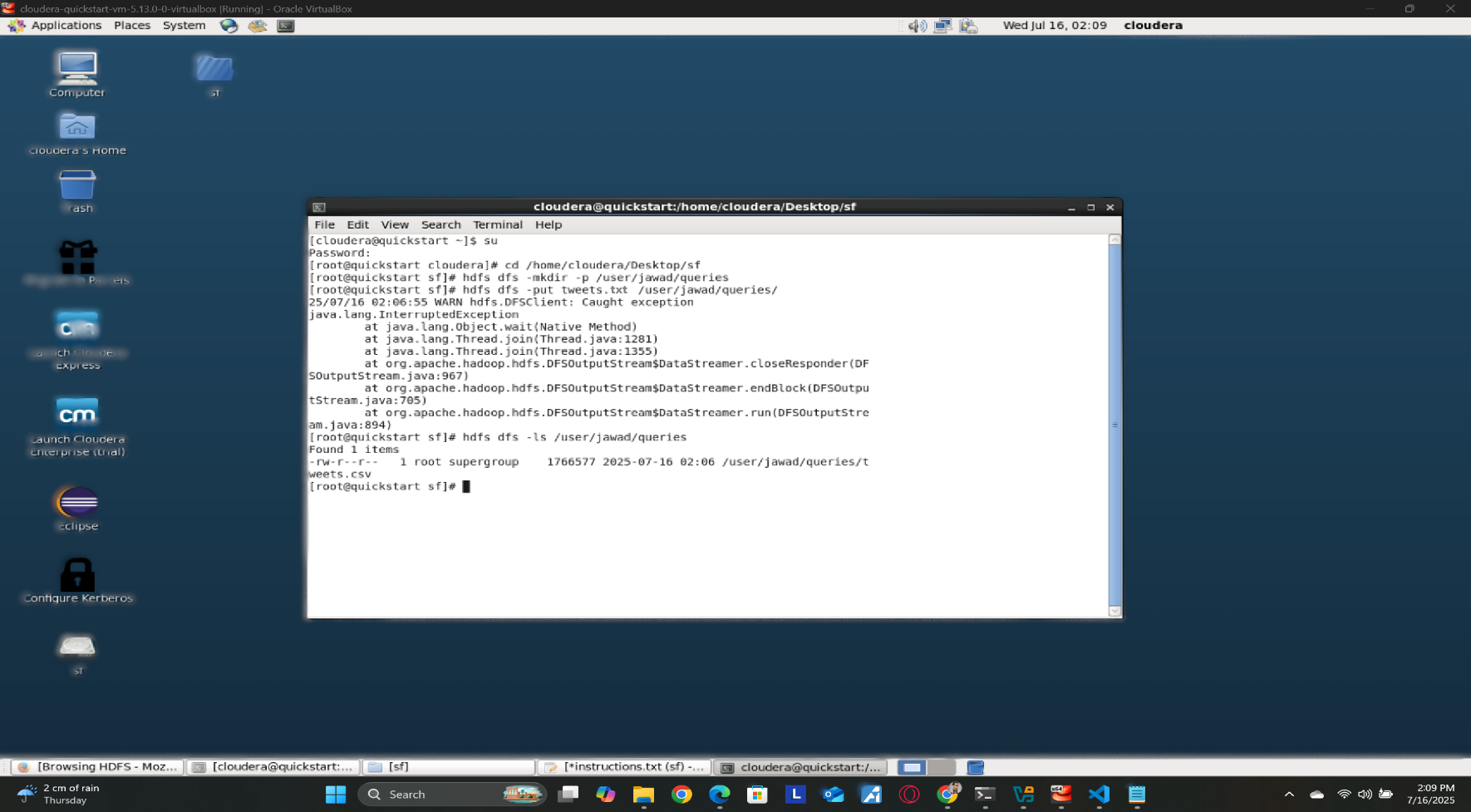
### Input File: tweets.txt

* **Format**: Each line contains a timestamp (YYYY-MM-DD HH) followed by the tweet text.
* **Size**: The file includes tweets with sentiments, airline mentions, and URLs, requiring cleaning and tokenization.
* **Sample Entry**:
* 2015-02-24 11 @virginamerica what @dhepburn said.

2015-02-24 11 @virginamerica plus you've added commercials to the experience... tacky.

## **3. Data Ingestion and HDFS Usage**

The input file tweets.txt was ingested into Hadoop Distributed File System (HDFS) under the directory /user/jawad/queries/.



The file was stored in HDFS. Given the textual nature and relatively small size of tweets.txt, occupied a single HDFS block. The HDFS NameNode managed the metadata, while the DataNode stored the actual data, ensuring fault tolerance and scalability.

**HDFS Block Explanation**:

* **Block Size**: 128 MB (default in Cloudera Hadoop).
* **Replication Factor**: 3 (default), ensuring data redundancy across nodes.
* **Storage**: The tweets.txt file was split into blocks (if larger than 128 MB) and distributed across DataNodes. For a small file, it resides in one block, minimizing overhead.
* **NameNode Metadata**: Stores file path, block locations, and permissions.

## **4. MapReduce Pipeline**

The MapReduce pipeline was implemented to process the tweet data, extract topics, and compute the top 5 trending topics per hour. Below is the logic for each phase.

### 4.1 Mapper

The Mapper processes each tweet to extract the hour and candidate topics (words). The steps include:

1. **Parsing**: Split each line into timestamp (YYYY-MM-DD HH) and tweet text.
2. **Cleaning**: Remove URLs, special characters, punctuation, and stopwords (e.g., "and", "the"). Convert text to lowercase.
3. **Tokenization**: Split the tweet text into words, filtering out irrelevant terms (e.g., mentions like @username, unless they are airline handles like @VirginAmerica).
4. **Output**: Emit key-value pairs in the format ((hour, word), 1).

### 4.2 Reducer

The Reducer aggregates counts for each (hour, word) pair and selects the top 5 words per hour based on their counts. The steps include:

1. **Aggregation**: Sum the counts for each (hour, word) pair.
2. **Grouping**: Group by hour and sort words by count in descending order.
3. **Top N Selection**: Select the top 5 words for each hour.
4. **Output**: Write results in the format hour\ttopic1:count, topic2:count, topic3:count, topic4:count, topic5:count.

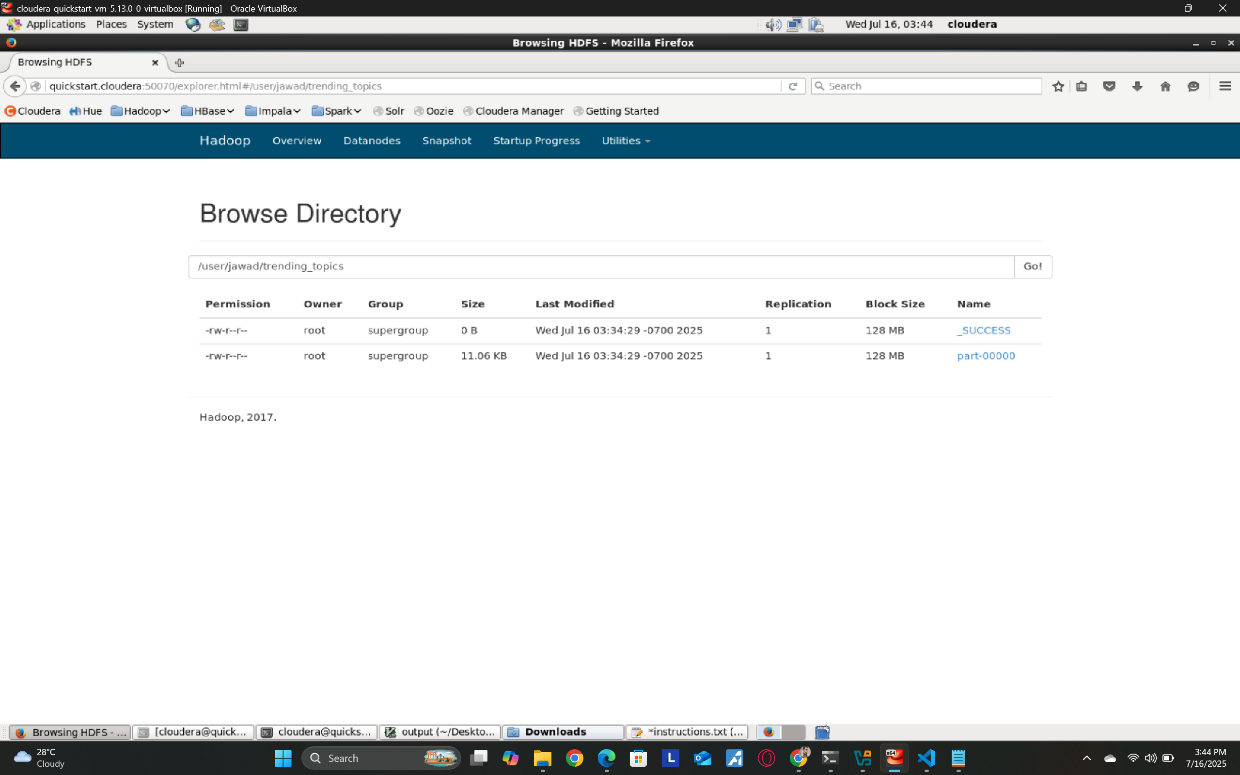
**Screenshots**:

### 

### 

### 

The output file, part-00000, contains the top 5 trending topics (words) per hour with their respective counts, covering the period from February 16, 2015, to February 24, 2015.

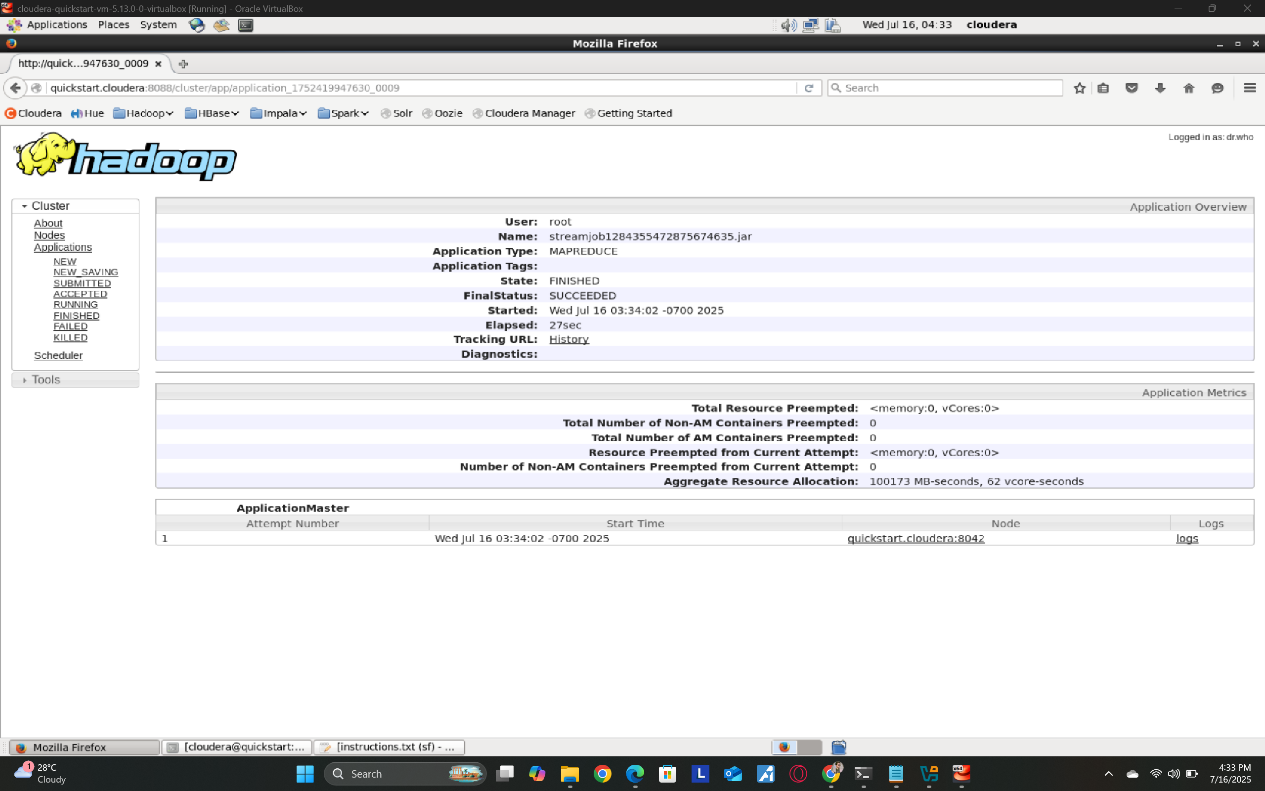


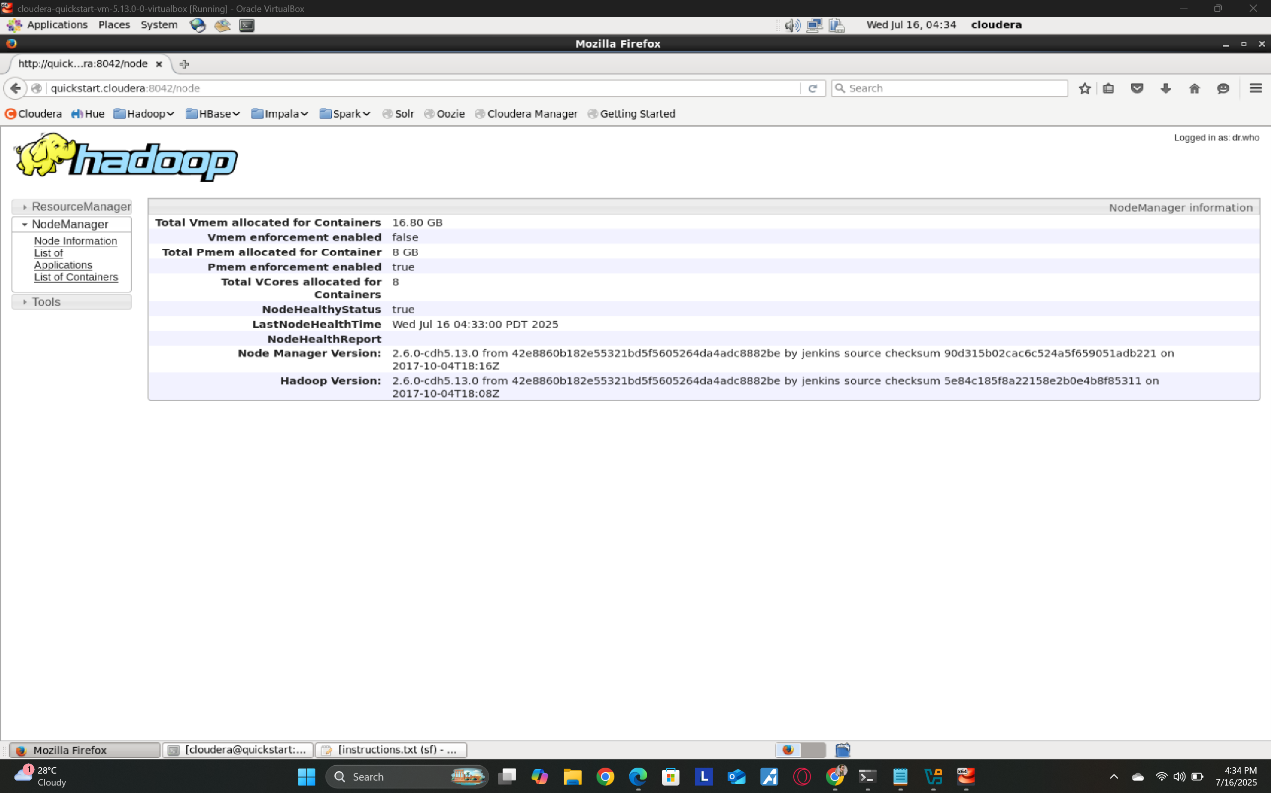
## **5. Hadoop Monitoring and Job Performance**

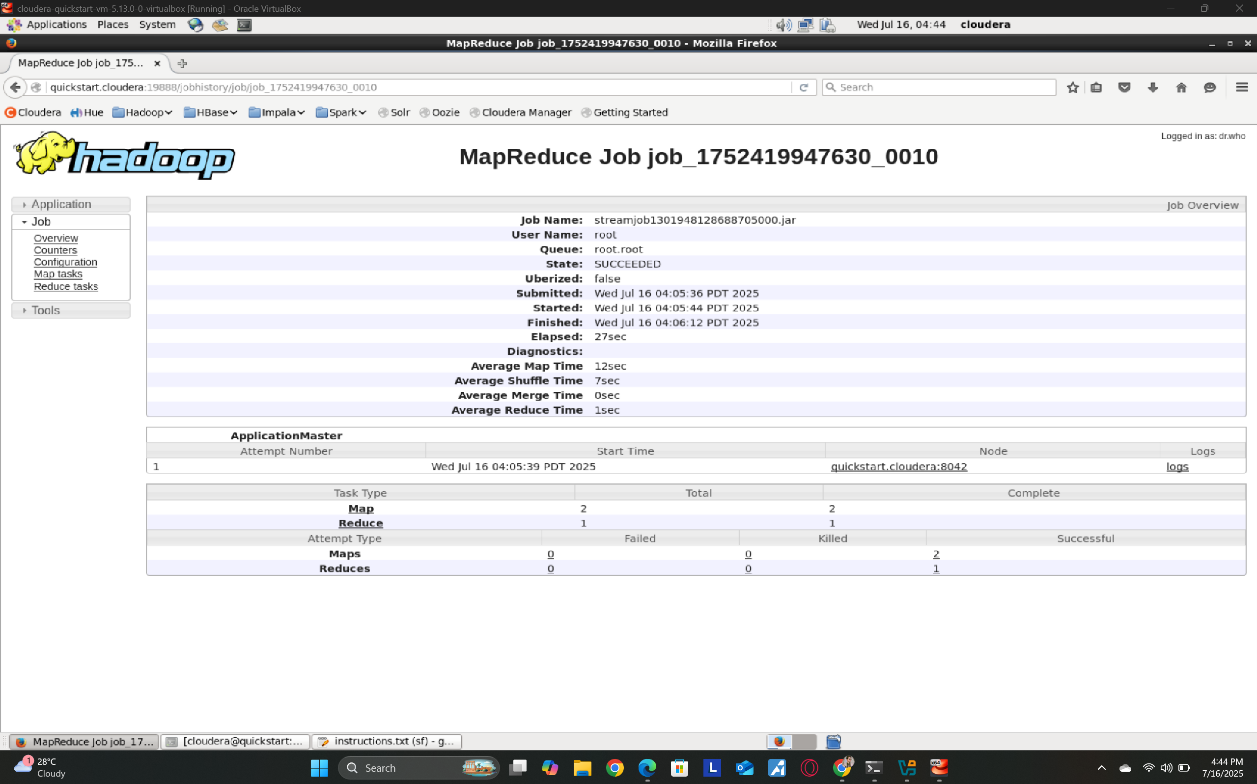
The MapReduce job was monitored using Cloudera’s ResourceManager UI and Hue. Key metrics include:

* **Mapper Tasks**: Number of Mapper tasks depended on the input file splits (1 for a small file).
* **Reducer Tasks**: Configured to 1 to consolidate output into a single file (part-00000).
* **Resource Usage**: CPU and memory usage were tracked via Cloudera Manager, showing efficient resource allocation.
* **Logs**: Job logs were accessed via Hue to verify successful completion and diagnose any errors.

**Screenshots**:







## **6. Data Analysis and Insights**

The output file part-00000 reveals the top 5 trending topics per hour from February 16 to February 24, 2015. Key observations:

* **Dominant Topics**: Airline handles (e.g., americanair, united, southwestair, jetblue, usairways) frequently appear, reflecting the dataset’s focus on airline-related tweets.
* **Common Words**: Words like the, you, for, flight, and and dominate due to their frequent use in tweet text. This suggests that while airline mentions are key topics, general terms also rank high, possibly due to insufficient stopword filtering or the conversational nature of tweets.
* **Temporal Trends**:
  + On February 22–24, americanair becomes the most frequent topic, peaking at 132 mentions on February 22, 17:00, indicating a surge in tweets (possibly due to cancellations or delays, as seen in tweets.txt).
  + Earlier days (February 16–21) show more balanced mentions of united, southwestair, and usairways, suggesting varied customer interactions.
* **Sentiment Context**: From tweets.txt, negative sentiments (e.g., complaints about cancellations, delays, and customer service) dominate, especially for americanair on February 22–24, correlating with high mention counts.

## **7. Visualization**

The hourly trends were visualized using Python with the Streamlit library to create an interactive dashboard. The visualization displays the top 5 topics per hour as a time-series plot, with each topic represented by a line showing its count over time.

Open this Recording file to see the results:



## **8. Conclusion**

The MapReduce pipeline successfully processed the tweet dataset to identify hourly trending topics, leveraging HDFS for storage, MapReduce for processing, and Streamlit for visualization. The analysis revealed a dominance of airline-related topics, with americanair peaking on February 22–24, likely due to customer service issues. Improvements could include enhanced stopword filtering and topic modeling to extract more specific themes (e.g., delay, service). The pipeline was monitored effectively using Cloudera tools, ensuring scalability and reliability.