Report: Implementation of MapReduce for Text Analysis

Srinibas Masanta (202318054)

February 14, 2024

1 Introduction

This report documents the process of implementing the MapReduce algorithm for text analysis using PySpark. The objective was to analyze the frequency of words in a given text dataset.

2 Setup and Environment Configuration

- Python environment with PySpark installed was set up.
- Access to a cloud-based storage service (Google Drive, DBFS, etc.) for storing the text dataset was ensured.
- The PySpark environment was configured to connect to the cloud storage service.

3 Data Loading

- The text dataset was loaded from the cloud storage service into an RDD (Resilient Distributed Dataset) using PySpark's `textFile()` function.
- Text files were loaded from both Google Drive and Databricks File System (DBFS) to demonstrate versatility.

4 Word Splitting

- Regular expressions were utilized to split the text into individual words.
- A regular expression pattern for word splitting was defined, considering various punctuation marks, whitespace characters, and special symbols.
- The text dataset was split into words using the defined pattern.

5 Word Counting (Map Phase)

- Each word was transformed into a key-value pair, where the word was the key and the count was initialized to 1.
- Each word was mapped to its corresponding count, effectively creating a list of tuples $(\text{word}, 1)$.

6 Aggregation and Reduction (Reduce Phase)

- The key-value pairs were aggregated by key (word) and the counts were summed up.
- The key-value pairs were reduced by combining counts for each word.

7 Sorting (Optional)

- The word counts were sorted in descending order based on their frequencies.
- This step was optional but provided valuable insights into the distribution of word frequencies.

8 Results and Analysis

- The analysis of the text dataset yielded the following results:
  - Total number of words analyzed: 328,091
  - Most frequent words (top 20):
    - **sed:** 7,575
    - **in:** 6,438
    - **amet:** 6,174
    - **sit:** 6,103
    - **ut:** 5,200
    - **id:** 5,198
    - **eget:** 5,024
    - **et:** 4,667
    - **nunc:** 4,613
    - **vitae:** 4,528
    - **at:** 4,377
    - **enim:** 4,045
    - **eu:** 3,812
    - **egestas:** 3,739
    - **pellentesque:** 3,675
    - **diam:** 3,582
    - **viverra:** 3,519
    - **quis:** 3,497
    - **ac:** 3,478
    - **arcu:** 3,368

9 Conclusion

- Key findings and insights obtained from the MapReduce analysis were summarized.
- Reflections were made on the effectiveness and efficiency of the implemented algorithm in analyzing text data.
- Potential areas for further optimization or expansion of the analysis were discussed.