Big Data Lab Solution

MapReduce Programming Model (Solution)

Arjun vankani

12-08-2025

Aim

Implement a MapReduce-style program to count word frequencies from a text file and display the **Top 10 most frequent words**.

Step-by-Step Solution

Step 1 – Mapper Function

```
def mapper(file_path):
    mapped = []
    with open(file_path, ’r’) as f:
        for line in f:
            words = line.strip().split()
            for word in words:
                mapped.append((word.lower(), 1))
    return mapped
```

Step 2 – Reducer Function

```
from collections import defaultdict

def reducer(mapped_data):
    reduced = defaultdict(int)
    for word, count in mapped_data:
        reduced[word] += count
    return reduced
```

Step 3 – Main Program

```
if __name__ == "__main__":
    file_path = "data.txt"

    # Map phase
    mapped_data = mapper(file_path)

    # Reduce phase
    reduced_data = reducer(mapped_data)

    # Sort by frequency (descending)
    sorted_words = sorted(reduced_data.items(),
                          key=lambda x: x[1],
                          reverse=True)

    # Print Top 10
    print("Top␣10␣Most␣Frequent␣Words:")
    for word, freq in sorted_words[:10]:
        print(f"{word}:␣{freq}")
```

Example Output

```
Top 10 Most Frequent Words:
the: 150
and: 120
to: 100
of: 95
a: 90
in: 88
is: 80
it: 75
that: 70
for: 65
```

Explanation

1. The **Mapper** generates key-value pairs (`word`, 1).
2. The **Reducer** aggregates counts for each unique word.
3. Sorting is used to get the top 10 most frequent words.

Student Tasks

1. Modify the program to ignore common stopwords such as `the`, `and`, `of`, etc.
2. Display the **Top 5 least frequent words**.
3. Count only words with more than 5 characters.
4. Extend the program to process multiple text files.