University Of Science And Technology Of Hanoi

Distributed Systems

Practical Work 5: The Longest Path
Custom MapReduce Framework in C

Luong Quynh Nhi, 23BI14356, Cyber Security

Lecturer: Ms. Le Nhu Chu Hiep

1. Introduction

The Longest Path problem involves reading multiple text files, where each file represents data collected from a different laptop. Each line in these files contains the full path of a file stored on that laptop. The objective of this task is to determine the longest file path or paths among all input files. For example, given the following input paths:
/home/user/documents/report.pdf
/home/user/music/songs/2025/new/song.mp3
/etc/hosts

The output of the program should be: /home/user/music/songs/2025/new/song.mp3
This path is selected because it has the greatest number of characters.

This assignment requires building a MapReduce-style solution written in the C programming language.The processing pipeline consists of two main stages:

- Mapper: Reads file paths line by line and outputs pairs of (length, path)
- Reducer: Finds the maximum length and outputs all paths that match it

2. Why MapReduce?

MapReduce is a powerful programming model designed to process large amounts of data efficiently. It is especially useful when:

- The input consists of large text files
- Each line can be processed independently

A final aggregation step is required

The Longest Path problem is well suited for the MapReduce model because each file path can be processed separately, and the final result depends on comparing all computed lengths.

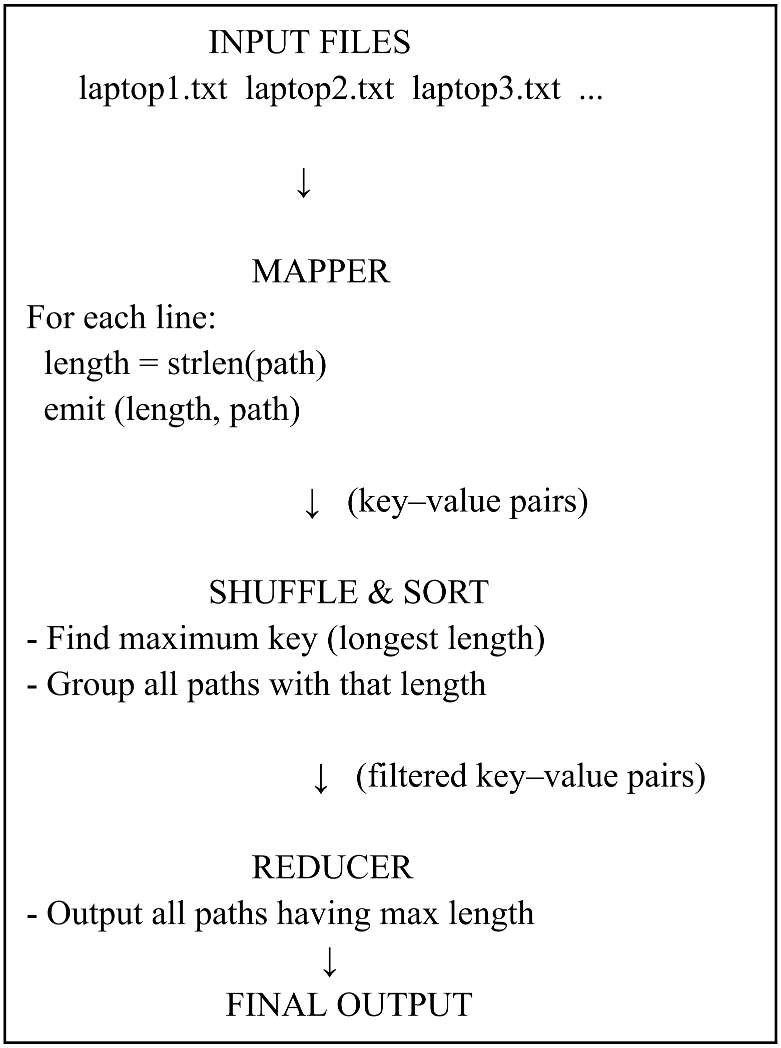
The roles of each MapReduce stage are summarized below:

- Mapper: Computes the length of each file path
- Shuffle: Groups all paths by their lengths
- Reducer: Finds the global maximum length and prints the longest path(s)

Even though this solution is implemented in C on a single machine, the logical workflow mirrors that of a real distributed MapReduce system used in big data environments.

3. System Architecture

The system architecture follows a simplified MapReduce pipeline. Multiple input files are provided, with each file representing data from a different laptop. These files are processed sequentially, simulating distributed data sources.



INPUT FILES

laptop1.txt laptop2.txt laptop3.txt ...

↓

MAPPER

```
For each line:
  length = strlen(path)
  emit (length, path)
```

↓ (key–value pairs)

SHUFFLE & SORT

- Find maximum key (longest length)
- Group all paths with that length

↓ (filtered key–value pairs)

REDUCER

- Output all paths having max length

↓

FINAL OUTPUT

3.1 Mapper Operation

The Mapper is responsible for processing the input files. It reads each file line by line, where each line represents a full file path.

For every path, the Mapper:

- Removes the newline character
- Calculates the length of the path using a string length function

- Emits a key–value pair in the form (path length, file path)

Each line is processed independently, making this stage suitable for parallel execution in real MapReduce systems.

3.2 Shuffle and Sort Operation

The Shuffle and Sort stage collects all key–value pairs produced by the Mapper. During this phase:

- All path lengths are examined
- The maximum length is identified
- Paths that share the same maximum length are grouped together

This stage prepares the data so that the Reducer can easily determine the longest path or paths.

3.3 Reducer Operation

The Reducer receives the grouped results from the Shuffle phase. Its main task is to produce the final output.
The Reducer:

- Identifies the maximum path length
- Selects all file paths with this length
- Outputs all longest paths

If multiple file paths have the same longest length, all of them are printed as valid results.

4. Execution

Test Input Files

```
laptop1.txt
/home/user/docs/report.pdf
/etc/hosts
```

```
laptop2.txt
/home/user/music/2025/new/song.mp3
```

The program is compiled using GCC: *gcc longest_path.c -o longest_path.exe*
Program Execution: *./longest_path.exe*

User input:
Enter number of input files: 2
Enter file 1 name: laptop1.txt
Enter file 2 name: laptop2.txt

Result

```
/home/user/music/2025/new/song.mp3
Longest Length = 36 characters
```

The program correctly identifies and outputs the longest file path among all input files.

5. Conclusion

This practical work demonstrates how the MapReduce programming model can be applied to solve the Longest Path problem. Although implemented locally in C, the program follows the same logical stages as a distributed MapReduce system: mapping, shuffling, and reducing.

The solution efficiently processes multiple input files and correctly identifies the longest file path or paths. This project provides a strong foundation for understanding large-scale data processing and distributed computing concepts.