UNIVERSITY OF SCIENCE & TECHNOLOGY OF HANOI
TRƯỜNG ĐẠI HỌC KHOA HỌC & CÔNG NGHỆ HÀ NỘI

Practical Work 5: Longest Path

Vũ Đức Hiếu - BI12-162

I) System architecture

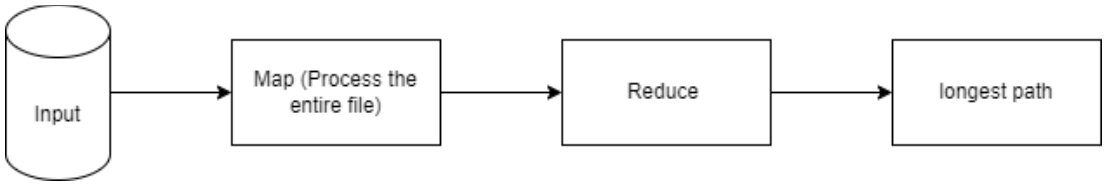

Figure 1. Workflow of MapReduce

II) Implementation

- Map function():
  +) The map function reads each line (path) from the input file, removes the newline character (if present), and compares the length of the current path with the length of the longest path found so far. If the current path is longer, it updates the longest path variable with the content of the current path.
- Reduce function():
  +) The reduce function opens the file specified by chunk_path, calls the map function to process the file chunk, and then closes the file. The purpose of the reduce function here is to perform the mapping step on a specific chunk of data, which is provided as a file path. In this implementation, the reduce function acts as a wrapper around the map function, allowing for modularization and separation of concerns.
- Detail explanation:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_PATH_LENGTH 256

char longest_path[MAX_PATH_LENGTH] = ""; // Store the longest path found

// Map function
void map(FILE *file) {
    char path[MAX_PATH_LENGTH];
    // Read each line from the file
    while (fgets(path, MAX_PATH_LENGTH, file) != NULL) {
        int length = strlen(path);
        // Remove newline character if present
        if (path[length - 1] == '\n')
            path[length - 1] = '\0';

        // Check if path length is greater than current longest path length
        if (length > strlen(longest_path)) {
            // Update the longest path if the current path is longer
```

```
            strcpy(longest_path, path);
        }
    }
}

// Reduce function
void reduce(char *chunk_path) {
    // Open the file specified by chunk_path for reading
    FILE *file = fopen(chunk_path, "r");
    if (file == NULL) {
        // Print an error message if file opening fails
        perror("Error opening file");
        // Exit the program with failure status
        exit(EXIT_FAILURE);
    }

    // Perform mapping on the chunk
    map(file);

    // Close the file
    fclose(file);
}

int main() {
    // Open the file "paths.txt" for reading
    FILE *file = fopen("paths.txt", "r");
    if (file == NULL) {
        // Print an error message if file opening fails
        perror("Error opening file");
        // Exit the program with failure status
        return 1;
    }

    // Perform mapping on the entire file
    map(file);

    // Close the file
    fclose(file);

    // Reduce the results
    printf("The longest path found: %s\n", longest_path);

    return 0;
}
```

- Result:

```
PS D:\ds2024\longest_path> gcc .\longest_path.c -o lp
PS D:\ds2024\longest_path> ./lp
THe longest path found: /Users/user/Documents/folder1/file2.txt
PS D:\ds2024\longest_path>
```